

Article

On the Application LBP Texture Descriptors and Its Variants for No-Reference Image Quality Assessment

Pedro Garcia Freitas ^{1,*} , Luísa Peixoto da Eira ², Samuel Soares Santos ²
and Mylene Christine Queiroz de Farias ² 

¹ Department of Computer Science, University of Brasília, Brasília 73345-010, Brazil

² Department of Electrical Engineering, University of Brasília, Brasília 73345-010, Brazil;
luisa.eira@gmail.com (L.P.d.E.); samuelss0906@gmail.com (S.S.S.); mylene@ieee.org (M.C.Q.d.F.)

* Correspondence: pedrogarcia@ieee.org

Received: 16 July 2018; Accepted: 26 September 2018; Published: 3 October 2018



Abstract: Automatic assessing the quality of an image is a critical problem for a wide range of applications in the fields of computer vision and image processing. For example, many computer vision applications, such as biometric identification, content retrieval, and object recognition, rely on input images with a specific range of quality. Therefore, an effort has been made to develop image quality assessment (IQA) methods that are able to automatically estimate quality. Among the possible IQA approaches, No-Reference IQA (NR-IQA) methods are of fundamental interest, since they can be used in most real-time multimedia applications. NR-IQA are capable of assessing the quality of an image without using the reference (or pristine) image. In this paper, we investigate the use of texture descriptors in the design of NR-IQA methods. The premise is that visible impairments alter the statistics of texture descriptors, making it possible to estimate quality. To investigate if this premise is valid, we analyze the use of a set of state-of-the-art Local Binary Patterns (LBP) texture descriptors in IQA methods. Particularly, we present a comprehensive review with a detailed description of the considered methods. Additionally, we propose a framework for using texture descriptors in NR-IQA methods. Our experimental results indicate that, although not all texture descriptors are suitable for NR-IQA, many can be used with this purpose achieving a good accuracy performance with the advantage of a low computational complexity.

Keywords: texture descriptors; random forest regression; no-reference image quality assessment; machine learning

1. Introduction

With the fast growth of imaging systems, a large number of digital images are being generated every day. These images are often altered in the acquisition, transmission or compression stages. These alterations can introduce distortions that may affect how human and machines understand the image content. Therefore, multimedia and computer vision applications can really benefit from automatic tools that are capable of assessing image quality. More specifically, image quality assessment (IQA) methods can be used, for example, to determine optimal codec parameters [1], find best perceptual coding schemes [2–6], and design efficient image watermarking algorithms [7,8]. Moreover, a recent report by Conviva[®] shows that viewers are demanding a higher quality of delivered multimedia content [9]. As users' demands increase, the importance of designing automatic tools to predict the quality of the visual stimuli also increases.

In the context of computer vision (CV), the quality of input images can affect the performance of the algorithms. For instance, Kupyn et al. [10] have shown that object detection methods based on deep learning approaches are greatly affected by the quality of the input images, as can be seen

in the images depicted in Figure 1. Moreover, Dodge and Karam [11] demonstrated that deep neural networks are susceptible to image quality distortions, particularly to blur and noise. Another examples of known CV algorithms that are affected by the quality of the input images include finger vein detection [12], biometric sensor spoofing [13], face recognition [14], video stream recognition systems [15], deep learning reconstruction of magnetic resonance imaging (MRI) [16], and multi-view activity recognition [17].



Figure 1. Object detection using YOLO [18] on the distorted (**left**) and pristine (**right**) images, taken from GoPro [19] dataset. The detection effectiveness of YOLO is remarkably impaired by the quality of the input image.

There are mainly two ways of measuring image quality. The first consists of performing psychophysical experiments in which humans rate the quality of a set of images. These experiments use standardized experimental methodologies to obtain quality scores for a broad range of images processed with a diverse number of algorithms and procedures. Since these experiments use human subjects, this approach is known as subjective quality assessment and it is considered the most accurate method to estimate quality [20]. Unfortunately, subjective methods are expensive and time-consuming and, therefore, are unsuitable for the most real-time applications. The second approach consists of using computer algorithms to obtain a quality estimate. Since this does not require human subjects, this approach is often entitled ‘objective quality assessment’. If a given objective method produces results that are well correlated with the quality scores provided by human viewers, it can be used to replace subjective methods.

Objective IQA methods are classified according to the amount of available reference information they require. If the full reference image (pristine content) is required to estimate quality, the method is classified as full-reference (FR). If the method only requires a limited amount of information regarding the reference image, the method is a reduced-reference (RR) method. Since requiring full or limited reference information can be a severe impediment for applications, one solution is to use no-reference (NR) methods, which evaluate the quality of images without requiring any information about the reference image. Objective methods can also be classified according to their target applications. Methods designed for specific applications are known as distortion-specific (DS) methods. DS methods can be designed to estimate the amount of sharpness [21–23], JPEG/JPEG2000 degradations [24,25], blockiness artifacts [26], contrast distortions [27], and enhancement [28] in an image. Although DS methods can be useful for specific scenarios, they have limited applicability in the real world. An alternative to DS methods is the distortion-generic (DG) methods, which do not require a prior knowledge of the type of distortion and, therefore, are more adequate for diverse scenarios. As expected, the design of DG methods is more challenging [29,30].

According to Hemami and Reibman [31], the design of IQA methods requires three major steps: measuring, pooling, and mapping. Measuring refers to the extraction of a set of specific physical attributes of the image. In other words, the method must compute a set of image features that describes visual quality. Pooling refers to the combination of these measurements to create a link between the image features and its quality. Mapping refers to the model of correspondence between the

result of the pooling and the subjective scores. Most existing works focus on the measuring stage, where quality-aware features are designed to measure the level of image distortion. These features are usually based on the natural scene statistics (NSS) [32–35], assuming that pristine natural images have particular statistical properties that are disturbed by the distortions. NSS-based methods can extract features in different domains, such as discrete cosine transform (DCT) domain [36–38], discrete wavelet transform (DWT) domain [39–41], spatial domain [42], etc. More recently, convolutional neural networks (CNN) have also been used in the design of NR-IQA methods [43–45]. CNN-based methods use the direct correspondence between the hierarchy of the human visual system and the layers of a CNN [46–50].

Another trend has been the use of saliency models (SM) [51–54]. SMs provide a measure of the perceptual importance of each image region, which allows quality assessment methods to weight the distortion according to region importance. In other words, quality and saliency are inherently associated because both of them depend on how the human visual system perceives the content and, consequently, on how (suprathreshold) distortions are detected [53]. Some investigators have studied how to include saliency information into existing visual quality metrics in order to boost their performance [52,55–58]. Nevertheless, most of these investigations are targeted at either FR or DS image quality metrics.

In this paper, we investigate the suitability of texture descriptors to assess image quality. This paper has inspiration on the studies of Ciocca et al. [59] and Larson and Chandler [60]. The premise is that visible impairments alter the statistics of texture descriptors, making it possible to estimate image quality. To investigate this premise, we analyze the use of a set of state-of-the-art texture descriptors in quality assessment methods. Additionally, we propose a framework to use these texture descriptors for NR-IQA. The framework is based on supervised machine learning (ML) approach that takes into account how impairments affect the statistics of the texture descriptors. These statistics are used as feature vectors of a random forest regression algorithm that learns the predictive quality model via regression [61].

The rest of this paper is organized as follows. Section 2 presents a brief review of the texture descriptors investigated in this paper. Section 3 describes the proposed framework, the experimental setup, all simulation results, and a discussion of these results. Finally, Section 4 presents the conclusions.

2. Texture Descriptors

Texture is a fundamental attribute of images, but there is no consensus on its definition. Petrou and Garcia-Sevilla, for instance, define texture as a variation of the visual stimuli at scales smaller than the scale of interest [62]. Davies associates texture to patterns with both randomness and regularity [63]. In this paper, texture refers to area characteristics that are perceived as combinations of basic image patterns. These basic patterns present a certain regularity that is captured by statistical measures.

To characterize a texture, texture analysis methods identify and select a set of relevant texture features. Over the years, several texture analysis methods have been proposed, using a variety of approaches [62,63], including gray level run-length (GLRLM) [64], gray level co-occurrence matrices (GLCM) [65], texture spectrum [66], and textons [67]. Among the popular texture operators is the local binary patterns (LBP) [68], which describes the local textures of an image by performing simple operations. More specifically, the textures are labeled according to the relationships between each pixel and its neighbors. One of the advantages of the LBP descriptor is that it unifies traditional texture analysis models.

There are several modifications of the LBP operator [69,70]. Most of them try to improve the performance of the LBP in specific applications (e.g., texture classification, face recognition, object detection, etc.). However, few works have investigated the performance of the LBP (and its variants) in specific applications. This paper is inspired by the work of Hadid et al. [69], who compared the performance of 13 different LBP-based methods in gender recognition applications. Our focus is to test

the performance of LBP-based descriptors in IQA applications. This section describes the basic LBP descriptor and the state-of-the-art LBP variants considered in this work.

2.1. Basic Local Binary Patterns (LBP)

The Local Binary Pattern (LBP) is arguably one of the most powerful texture descriptors. It was first proposed by Ojala et al. [68] and it has since been proven to be an effective feature extractor for texture-based problems. The traditional LBP descriptor takes the following form:

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

where

$$S(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In Equation (1), $I_c = I(x, y)$ is an arbitrary central pixel at the position (x, y) and $I_p = I(x_p, y_p)$ is a neighboring pixel surrounding I_c , where

$$x_p = x + R \cos \left(2\pi \frac{p}{P} \right) \tag{3}$$

and

$$y_p = y - R \sin \left(2\pi \frac{p}{P} \right). \tag{4}$$

P is the total number of neighboring pixels I_p , sampled with a distance R from I_c . Figure 2 illustrates examples of symmetric samplings with different numbers of neighboring points (P) and radius (R) values.

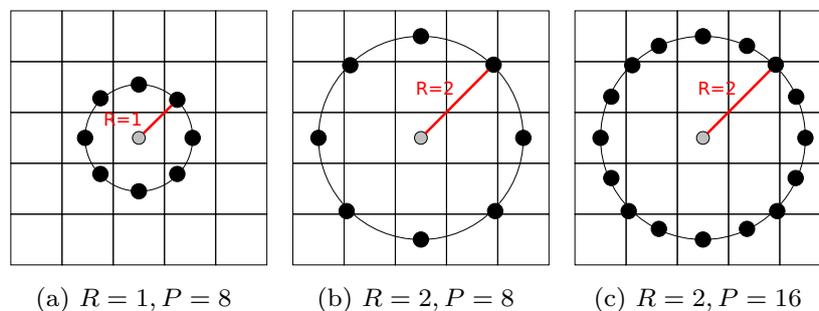


Figure 2. Circularly symmetric P neighbors extracted from a distance R .

Figure 3 illustrates the steps for applying the LBP descriptor on a single pixel ($I_c = 8$) located in the center of a 3×3 image block, as shown in the bottom-left of this figure. The numbers in the yellow squares of the block represent the order in which the descriptor is computed (counter-clockwise direction starting from 0). In this figure, we use a unitary neighborhood radius ($R = 1$) and eight neighboring pixels ($P = 8$). After calculating $S(t)$ (Equation (2)) for each neighboring pixel I_p ($0 \leq p \leq 7$), we obtain a binary output for each I_p , as illustrated in upper-left position of Figure 3. The black circles correspond to '0' and white circles to '1'. These binary outputs are stored in a binary format, according to their position (yellow squares). The LBP output for I_c is the decimal number, obtained by converting this binary number. After the LBP is applied to all pixels in an image, we get a set of labels that compose the LBP channel. Figure 4 shows examples of LBP channels for the image 'Baboon', obtained using different radius values and different numbers of neighbors.

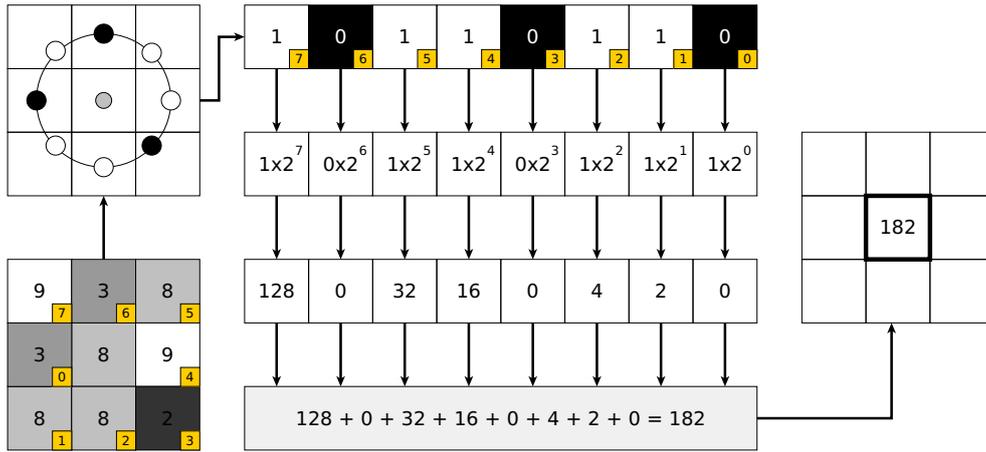


Figure 3. Calculation of LBP labels.

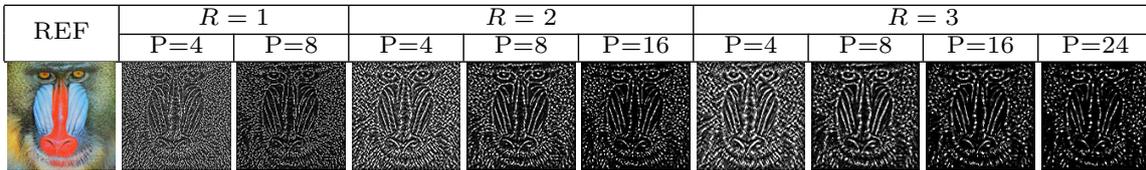


Figure 4. Reference image and its correspondent Local Binary Pattern (LBP) channels computed using three different radius (R) values.

When an image is rotated, the I_p sampled values move along the perimeter of the circumference around I_c , generating a circular shift in the binary number generated. As a consequence, a different decimal $LBP_{R,P}(I_c)$ value is obtained. To remove this effect, we can use the following rotation invariant (ri) descriptor, defined as:

$$LBP_{R,P}^{ri}(I_c) = \min\{ROTR(LBP_{R,P}(I_c), k)\}, \tag{5}$$

where $k = \{0, 1, 2, \dots, P - 1\}$ and $ROTR(x, k)$ is the circular bit-wise right shift descriptor that shifts the tuple x by k positions.

Due to the crude quantization of the angular space and to the occurrence of specific frequencies in individual patterns, $LBP_{R,P}$ and $LBP_{R,P}^{ri}$ descriptors do not always provide a good discrimination [71]. To improve the discriminability, Ojala et al. [68] proposed a ‘uniform’ descriptor that captures fundamental pattern properties:

$$LBP_{R,P}^u(I_c) = \begin{cases} \sum_{p=0}^{P-1} S(I_p - I_c), & \text{if } U(LBP_{R,P}^{ri}) \leq 2, \\ P + 1, & \text{otherwise,} \end{cases} \tag{6}$$

where

$$U(LBP_{R,P}^u) = \Delta(I_{P-1}, I_0) + \sum_{p=1}^{P-1} \Delta(I_p, I_{p-1}), \tag{7}$$

and

$$\Delta(I_x, I_y) = |S(I_x - I_c) - S(I_y - I_c)|. \tag{8}$$

In addition to a better discriminability, the uniform LBP descriptor has the advantage of generating fewer distinct LBP labels. While the ‘nonuniform’ descriptor (Equation (1)) produces 2^P different output values, the ‘uniform’ descriptor produces only $P + 2$ distinct output values. Finally, once the

LBP mask is calculated using any the LBP approaches described above, we compute its histogram. Next, we present some of the LBP variants that have been proposed to improve the robustness and discriminability of the original descriptor.

2.2. Local Ternary Patterns (LTP)

The LTP operator is an extension of the LBP descriptor that assumes up to 3 coded values ($\{-1, 0, 1\}$). This is achieved by changing the step function S in the following manner:

$$\hat{S}(t) = \begin{cases} 1, & t \geq \tau, \\ 0, & -\tau < t < \tau \\ -1, & t < -\tau, \end{cases} \quad (9)$$

where τ is a threshold that determines how sharp an intensity change should to be considered as an edge. After computing the ternary codes, each ternary pattern is split into two codes: a positive (upper pattern) and a negative (lower pattern) codes, which are treated as two separate channels.

Figure 5 illustrates the basic feature extraction procedure for a single pixel using a LTP descriptor. The numbers in yellow squares represent the order in which the step function is computed (Equations (2) and (9)). In this example, we consider an unitary neighborhood radius ($R = 1$), eight neighboring pixels ($P = 8$), and a threshold τ equal to five. While in the LBP the binary code takes only two values (0 or 1, represented by the colors black and white), the LTP descriptor generates three possible values (see Equation (9)) that are represented by the colors black ($\hat{S}(t) = 1$), white ($\hat{S}(t) = 0$), and red ($\hat{S}(t) = -1$).

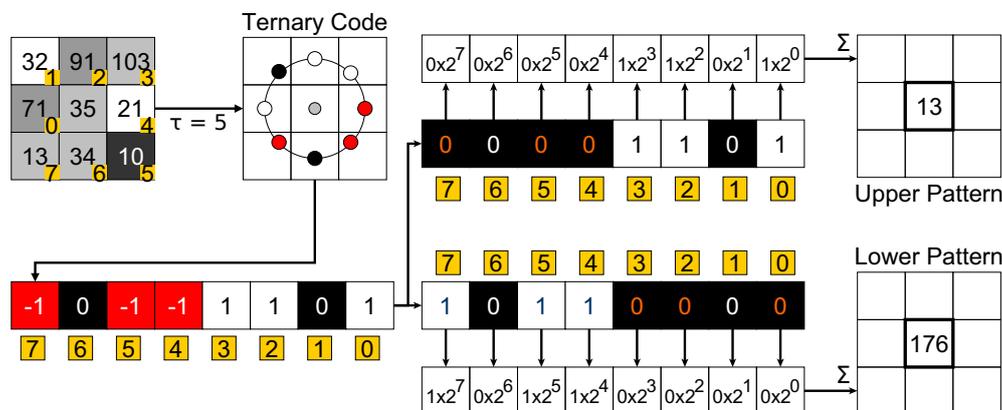


Figure 5. Illustration of the basic Local Ternary Pattern descriptor.

We split the LTP code into two LBP codes (with only positive values). First, we create the upper pattern by converting the negative codes to zero. Next, we create the lower pattern by setting the positives values to zero, by converting the negative values to positive. Comparing Figure 5 and Figure 3, we notice that the LTP descriptor generates two texture information maps that are two separate LBP channels. Finally, we compute independent histograms and similarity measures for each of these maps and combine these histograms to generate the feature vector.

2.3. Local Phase Quantization (LPQ)

A limitation of the LBP is its sensitivity to blur. To tackle this problem, the local phase quantization (LPQ) descriptor was proposed [72]. The LPQ descriptor performs a quantization of the Fourier transform phase in local neighborhoods. Assuming that $G(u)$ and $F(u)$ are the discrete

Fourier transforms (DFT) of the blurred $g(z)$ and original $f(z)$ images, which are related by the following equation:

$$G(u) = F(u) \cdot H(u). \tag{10}$$

Assuming that $h(x) = h(-x)$, its DFT is always real and the phase assumes only two values, namely:

$$\angle H(u) = \begin{cases} 0, & H(u) \geq 0 \\ \pi, & \text{otherwise.} \end{cases} \tag{11}$$

For the LPQ descriptor, the phase is computed in the local neighborhood N_z , for each pixel position of $f(z)$. The local spectrum is computed with the following equation:

$$F(u, x) = \sum_{y \in N_z} f(y) \cdot w_R(y - x) \cdot e^{-j2\pi uy}, \tag{12}$$

where u is the frequency and w_R is a window given by:

$$w_R(x) = \begin{cases} 1, & |x| < \frac{N_R}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The local Fourier coefficients are computed at four frequencies for each pixel position, i.e.,

$$F(x) = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)], \tag{14}$$

where $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$. In these cases, a is sufficiently small to satisfy $H(u_i) > 0$.

The phase of the Fourier coefficients is given by the signs of the real and imaginary parts of each component $F(x)$, computed by scalar quantization:

$$q_j = \begin{cases} 1, & g_j \geq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where g_j is the j -th component of $G(x) = [Re\{F(x)\}, Im\{F(x)\}]$. After generating the binary coefficients q_j , the feature vector is generated using the same technique used in the LBP.

2.4. Binarized Statistical Image Features (BSIF)

The binarized statistical image features (BSIF) is a descriptor proposed by Kannala and Rahtu [73], which does not use a manually predefined set of filters. Instead, it learns the filters using the statistics of natural images. BSIF is among the best texture descriptors for face recognition and texture classification applications [69,73]. Differently from previous descriptors, which operate on pixels, BSIF works on patches of pixels. Given an image patch X of size $l \times l$ pixels and a linear symmetric filter W_i of the same size, the filter response s_i is obtained computing the following expression:

$$s_i = \sum_{u,v} W_i(u, v) X(u, v) = w_i^T x, \tag{16}$$

where vectors w and x contain the pixels of W_i and X , respectively. The binarized feature is acquired using the following function:

$$b_i = \begin{cases} 1, & s_i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The filters W_i are learned via independent component analysis (ICA). The binarized features b_i are aggregated following the same procedure described for generating the LBP labels. The descriptive features are obtained by computing the histogram of the aggregated data.

Similarly to the LBP, which generates LBP channels, the BSIF generates coded images. These coded images are the set of labels generated after the binarized features are computed using Equation (17) and aggregated using Equation (1). The aggregation of BSIF results is based on a selected number of bits, instead of the number of neighbors of the labeled pixel. The labeling depends on the relationship between the patch size l and the number of binarized features b_i . Figure 6 shows the BSIF coded images corresponding to the same reference using different BSIF parameters. As can be seen in this figure, the textured information depends on the patch size l and on the number of bits. The number of bits is less or equal $l^2 - 1$. This is the reason why the second column does not contain BSIF coded images for 9, 10, 11, or 12 bits. Figure 6 shows that the choice of the number of bits and patch sizes is important for texture analysis algorithms. Therefore, multiscale approaches that incorporate several combinations of these parameters are interesting [74–77].

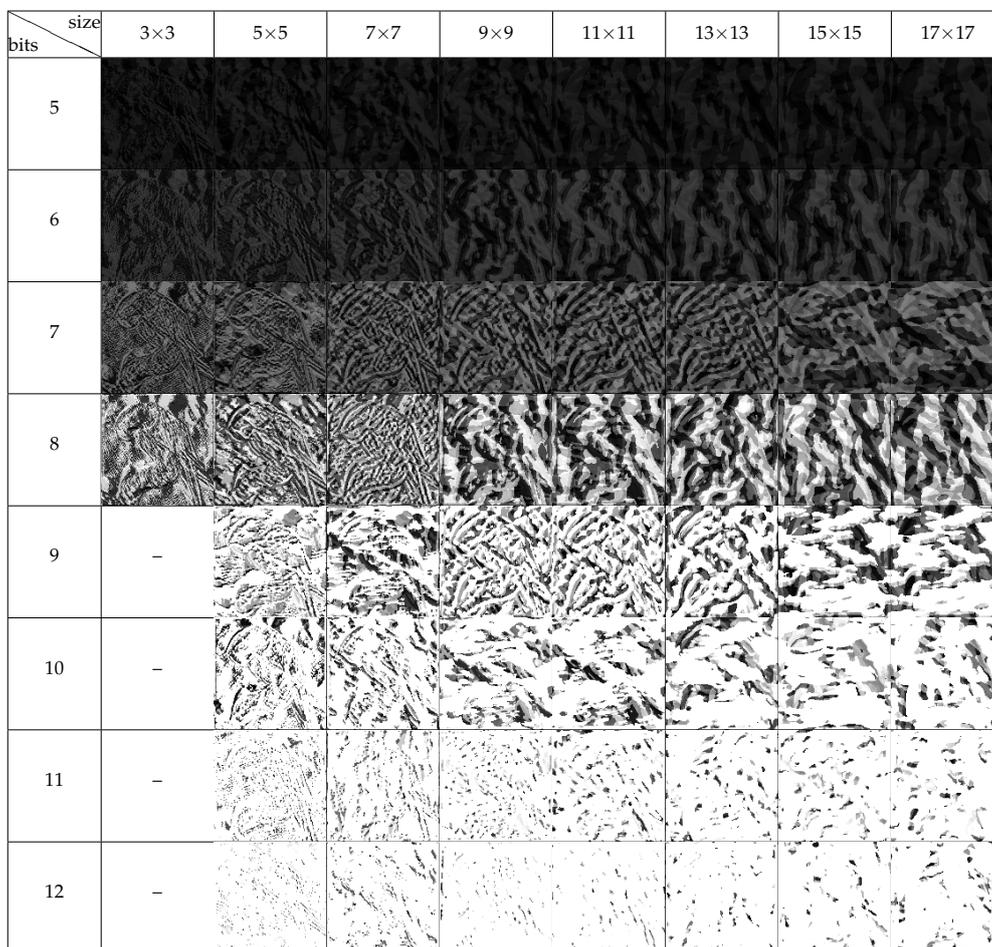


Figure 6. BSIF code images at different scales.

2.5. Rotated Local Binary Patterns (RLBP)

For some applications, image rotation affect the LBP results because of the fixed order of its weights. Since weights are distributed in a circular way, the effect of rotation can be eliminated by rotating the weights by the same angle. When the rotation angle is not known, an adaptive arrangement of weights, based on the locally computed reference direction, can be used. Mehta and Egiazarian [78]

proposed the rotated local binary (RLBP) descriptor, which considers that, if an image is rotated, the descriptor should be rotated by the same angle.

The RLBP makes the LBP invariant to rotation by circularly shifting the weights according to the dominant direction (D). In a neighborhood of a pixel I_c , D is the index of the neighbor whose difference to I_c is maximum, i.e.,

$$D = \operatorname{argmax}_{p \in \{0,1,\dots,P-1\}} |I_p - I_c|. \tag{18}$$

Since D is taken as a reference, the weights are assigned with respect to it. The RLBP descriptor is computed as follows:

$$RLBP_{R,P} = \sum_{p=0}^{P-1} S(I_p - I_c) 2^{(p-D \bmod P)}, \tag{19}$$

where $i \pmod j$ is the remainder of the division of i by j .

Figure 7 depicts the effect of a rotation on LBP and RLBP descriptors. Notice that the LBP changes for a rotation. The red color indicates pixels with values above the threshold, while the yellow color indicates pixels with the maximum difference to I_c (D). The position D takes the smallest weight, while the other positions get weights that correspond to circular shifts with relation to D. From Figure 7g, we notice that the weight corresponding to D is the same both for the original and rotated images, even when these pixels are at different angles. Therefore, the RLBP values for two rotated neighborhoods are the same.

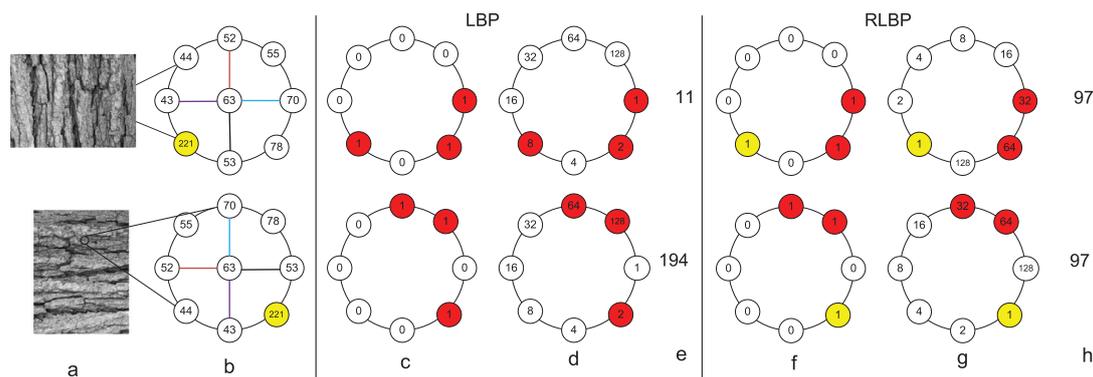


Figure 7. Rotation effect on LBP and RLBP descriptors: (a) original image and its rotated version, (b) Illustration of the neighbors rotation for the same pixel ‘63’, (c) Thresholded neighbors, values above threshold are shown in red color, (d) The weights corresponding to the thresholded neighbors, (e) LBP values, (f) Thresholded neighbors for RLBP with reference denoted in yellow color, (g) The weights of the thresholded neighbors, (h) The RLBP values for the original and rotated image is same [79].

Figure 8 shows the effect of rotation after generating the LBP and RLBP channels. The first row shows the LBP and RLBP maps of the original images and their corresponding histograms. The second row shows the same information for a 90 degrees rotated version of the original image. To compare the differences between the LBP and RLBP histograms, before and after the rotation, we use three statistical divergences measures: Kullback–Leibler divergence (KLD) [80], Jensen–Shannon divergence (JSD) [81], and chi-square distance (CSD) [82]. The KLD, JSD, and CSD of the LBP histogram are 2.92×10^{-2} , 6.96×10^{-3} , and 2.11×10^2 , respectively. These divergences for the RLBP histograms are 2.06×10^{-4} , 5.12×10^{-5} , and 1.57, respectively. Therefore, the order of magnitude of the LBP statistical divergences is two times higher than for the RLBP statistical divergences.

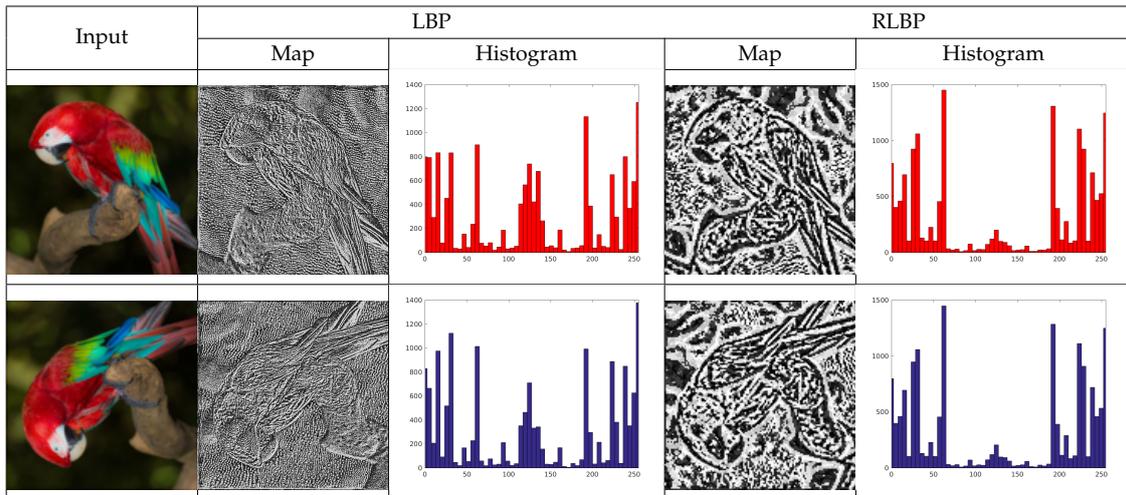


Figure 8. Effect of rotation on LBP and RLBP information.

2.6. Complete Local Binary Patterns (CLBP)

The LBP descriptor considers only the local differences of each pixel and its neighbors. The complete local binary patterns consider both signs (S) and magnitude (M) of the local differences, as well as the original intensity value of the center pixel [83]. Therefore, the CLBP feature is a combination of three descriptors, namely $CLBP_S$, $CLBP_M$, and $CLBP_C$. Figure 9 illustrates the computation of the CLBP feature.

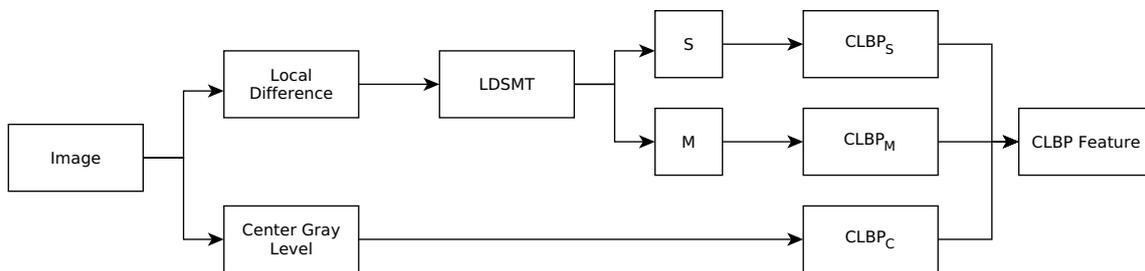


Figure 9. Framework of CLBP descriptor [83].

The $CLBP_S$ and $CLBP_M$ components are computed using the local difference sign-magnitude transform (LDSMT), which is defined as:

$$LDSMT_p = s_p \cdot m_p, \tag{20}$$

where $s_p = S(I_p - I_c)$ and $m_p = |I_p - I_c|$. The s_p is the sign descriptor used to compute $CLBP_S$, i.e., $CLBP_S$ is the same as the original LBP and it is used to code the sign information of the local differences. $CLBP_M$ is used to code the magnitude information of local differences:

$$CLBP_M = \sum_{p=0}^{P-1} thresh(m_p, c) \cdot 2^p, \tag{21}$$

where

$$thresh(x, c) = \begin{cases} 1 & x \geq c, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

In the above equation, c is a threshold set as the mean value of the input image I . Finally, the $CLBP_C$ is used to code the information of original center gray level value:

$$CLBP_C = thresh(I_c, c). \tag{23}$$

The three descriptors, $CLBP_S$, $CLBP_M$, and $CLBP_C$, are combined. Individual histograms are computed and concatenated. This joint histogram is used as a CLBP feature.

2.7. Local Configuration Patterns (LCP)

Local configuration patterns (LCP) is a rotation invariant image descriptor proposed by Guo et al., which is more discriminative [84]. LCP decomposes the image information into two levels: local structural information and microscopic configuration information. The local structural information is composed by LBP features, while the microscopic configuration (MiC) information is determined by the image configuration and by the pixel-wise interaction relationships.

To model the image configuration, we estimate the optimal weights, which are associated with the neighboring pixels, to linearly reconstruct the central pixel intensity for each pattern type. This can be expressed by the following equation:

$$E(a_0, a_1, \dots, a_{p-1}) = |I_c - \sum_{p=0}^{p-1} a_p I_p|, \tag{24}$$

where I_c and I_p denote the intensity values of the center pixel and neighboring pixels, a_p are weighting parameters associated with I_p , and $E(a_0, a_1, \dots, a_{p-1})$ are the reconstruction errors with respect to the model parameters. To minimize the reconstruction errors, the optimal parameters for each pattern are determined by a least squares estimation.

Suppose the occurrence of a particular pattern type j is f_j . There are f_j pixels in the image with the pattern j . We denote intensities of those f_j pixels as $c_{j,i}$, where $i = 0, 1, \dots, f_j - 1$. These intensities $c_{j,p}$ are organized into a vector:

$$\mathbf{c}_j = \begin{bmatrix} c_{j,0} \\ c_{j,1} \\ \vdots \\ c_{j,f_j-1} \end{bmatrix}. \tag{25}$$

We denote the intensities of neighboring pixels with respect to each $c_{j,i}$ as $v_{i,0}, \dots, v_{i,p-1}$, which are organized into a matrix with the following form:

$$\mathbf{V}_j = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,p-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{f_j-1,0} & v_{f_j-1,1} & \cdots & v_{f_j-1,p-1} \end{bmatrix}. \tag{26}$$

To minimize the reconstruction error (Equation (24)), the unknown parameters a_p are organized as a vector:

$$\mathbf{A}_j = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{p-1} \end{bmatrix}, \tag{27}$$

and the optimal parameters are determined by solving the following equation:

$$\mathbf{A}_j = (\mathbf{V}_j^T \mathbf{V}_j)^{-1} \mathbf{V}_j^T \mathbf{c}_j. \tag{28}$$

After determining \mathbf{A}_j , we apply the Fourier transform to the estimated parameter, which can be expressed by:

$$\mathbf{H}_j(k) = \sum_{p=0}^{P-1} \mathbf{A}_j(p) e^{-\frac{i2\pi kp}{P}}, \tag{29}$$

where $\mathbf{H}_j(k)$ is the k -th element of \mathbf{H}_j and $\mathbf{A}_j(p)$ is the p -th element of \mathbf{A}_j . The magnitude part of each element of the vector \mathbf{H}_j is taken as the resulting MiC, which is defined by:

$$|\mathbf{H}_j| = [|\mathbf{H}_j(0)|, |\mathbf{H}_j(1)|, \dots, |\mathbf{H}_j(P-1)|]. \tag{30}$$

The LCP feature is formed by both pixelwise interaction relationships and local shape information, which is expressed as:

$$LCP = [[|\mathbf{H}_0|; O_0], [|\mathbf{H}_1|; O_1], \dots, [|\mathbf{H}_{P-1}|; O_{P-1}]], \tag{31}$$

where $|\mathbf{H}_j|$ is computed using Equation (30) with respect to the j -th pattern and O_j is the number of occurrences of the j -th LBP label.

2.8. Opposite Color Local Binary Patterns (OCLBP)

To combine both texture and color information into a joint descriptor, Maenpaa [85] proposed to use the Opponent Color Local Binary Pattern (OCLBP) descriptor. This descriptor improves the descriptor proposed by Jain and Healey [86] by substituting the Gabor filter with a variant of the LBP descriptor, decreasing its computational cost. The OCLBP descriptor uses two approaches. In the first, the LBP descriptor is applied, individually, on each color channel, instead of being applied only on a single luminance channel. This approach is called ‘intra-channel’ because the central pixel and the corresponding sampled neighboring points belong to the same color channels. In the second approach, called ‘inter-channel’, the central pixel belongs to a color channel and its corresponding sampled neighboring points belong to another color channel. More specifically, for an $OCLBP_{MN}$ descriptor, the central pixel is positioned in the channel M , while the neighborhood is sampled in the channel N . For a three-channel color space, such as RGB, there are six possible combinations of channels: $OCLBP_{RG}$, $OCLBP_{RG}$, $OCLBP_{RB}$, $OCLBP_{RB}$, $OCLBP_{GB}$, and $OCLBP_{GB}$.

Figure 10 depicts the sampling approach of OCLBP when the central pixel is sampled in R channel. From this figure, we can notice that two combinations are possible: $OCLBP_{RG}$ (left) and $OCLBP_{RB}$ (right). In $OCLBP_{RG}$, the gray circle in the red channel is the central point, while the green circles in the green channel correspond to ‘0’ sampling points and the white circles correspond to ‘1’ sampling points, respectively. Similarly, in the $OCLBP_{RB}$ the blue circles correspond to ‘0’ sampling points and the white circles correspond to ‘1’ sampling points, respectively.

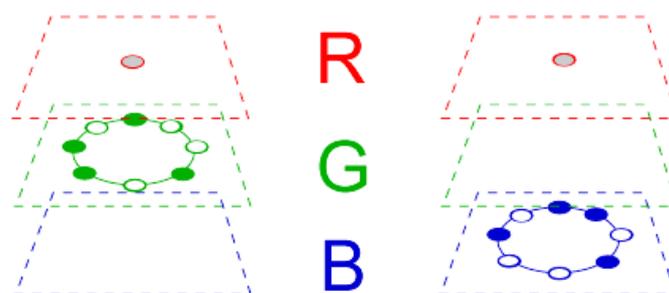


Figure 10. Sampling scheme for the $OCLBP_{RG}$ and $OCLBP_{RB}$ descriptors.

After computing the OCLBP descriptor for all pixels, a total of six texture channels are generated. As depicted in Figure 11, three LBP intra-channels (LBP_R , LBP_G , and LBP_B) and three LBP inter-channels ($OCLBP_{RG}$, $OCLBP_{RB}$, and $OCLBP_{GB}$) are generated. Although all possible combinations of the opposite color channels allow six distinct channels, we observed that the symmetric opposing pairs are very redundant (e.g., $OCLBP_{RG}$ is equivalent to $OCLBP_{GR}$). Due to this redundancy, only the three more descriptive inter-channels are used.

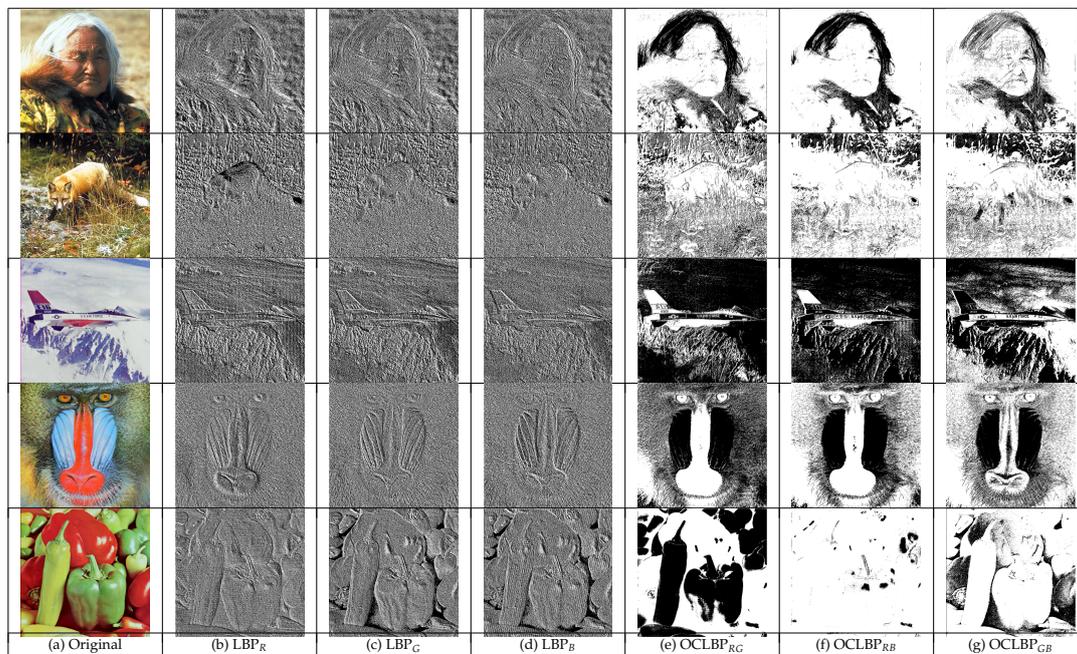


Figure 11. Original images and their output channels, computed using the OCLBP descriptor.

2.9. Three-Patch Local Binary Patterns (TPLBP)

Wolf et al. [87] proposed a family of LBP-related descriptors designed to encode additional types of local texture information. While variants of LBP descriptor use short binary strings to encode information about local micro-texture pixel-by-pixel, the authors considered capturing information which is complementary to that computed pixel-by-pixel. These patch-based descriptors are named Three-Patch LBP (TPLBP) and Four-Patch-LBP (FPLBP).

TPLBP considers a $w \times w$ patch centered on a pixel and S additional patches distributed uniformly on a ring of radius r around it, as illustrated in Figure 12. For an angle α , we get a set of neighboring patches along a circle and compare their values with those of the central patch. More specifically, the TPLBP is given by:

$$TPLBP_{r,S,w,\alpha}(p) = \sum_{i=0}^S f(\vartheta(C_i, C_p) - \vartheta(C_{i+\alpha \bmod S}, C_p)) \cdot 2^i, \quad (32)$$

where

$$f(t) = \begin{cases} 1, & \text{if } t \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (33)$$

The function $\vartheta(x, y)$ is any distance function between two patches under a vector representation. Examples of $\vartheta(x, y)$ are Manhattan [88], Mahalanobis [89], Minkowski [90], etc. The parameter τ is slightly larger than zero to provide some stability in uniform regions.

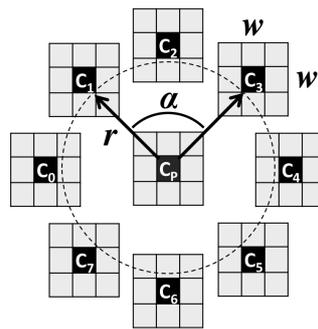


Figure 12. The Three-Patch LBP code with $\alpha = 2$, $w = 3$ and $\mathcal{S} = 8$ [87].

2.10. Four-Patch Local Binary Patterns (FPLBP)

In FPLBP, two rings centered on the pixel are used, instead of only one ring as used in TPLBP. As depicted in Figure 13, two rings of radii r_1 and r_2 (centered in the central pixel) are considered, with \mathcal{S} patches of size $w \times w$ equally distributed on each ring, positioned α patches away along the circle. We compare the two center symmetric patches in the inner ring with the two center symmetric patches in the outer ring. The bit in each coded pixel is set according to which of the two pairs is being compared. Therefore, the FPLBP code is computed as follows:

$$FPLBP_{r,S,w,\alpha}(p) = \sum_{i=0}^{\mathcal{S}/2} f(\vartheta(C_{1,i}, C_{2,i+\alpha \bmod S}) - \vartheta(C_{1,i+S/2}, C_{2,i+S/2+\alpha \bmod S})) \cdot 2^i. \quad (34)$$

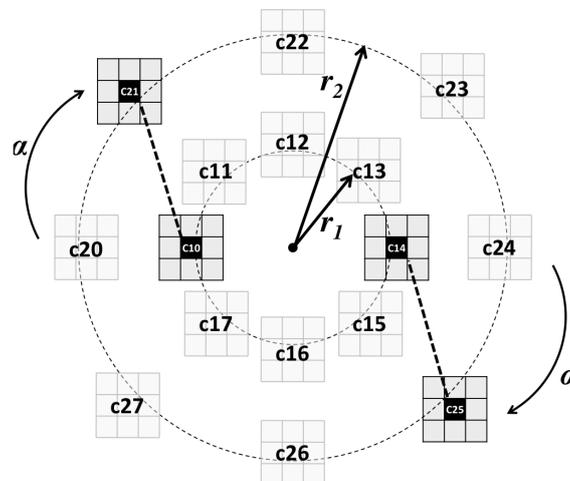


Figure 13. The Four-Patch LBP code with $\alpha = 1$, $w = 3$ and $\mathcal{S} = 8$ [87].

2.11. Multiscale Local Binary Patterns (MLBP)

The Multiscale local binary pattern (MLBP) is an extension of the LBP, designed with the goal of extracting image quality information [91]. A block diagram of the MLBP descriptor is depicted in Figure 14 and it is computed as follows. First, we generate several LBP channels, by varying the parameters R and P and performing a symmetrical sampling. For the smallest possible radius, $R = 1$, there are two possible P values that produce rotational symmetrical sampling ($P = 4$ and $P = 8$). When $R = 2$, there are three possible P values ($P = 4$, $P = 8$, and $P = 16$). In general, for a given radius R, there is a total of $R + 1$ distinct LBP channels.

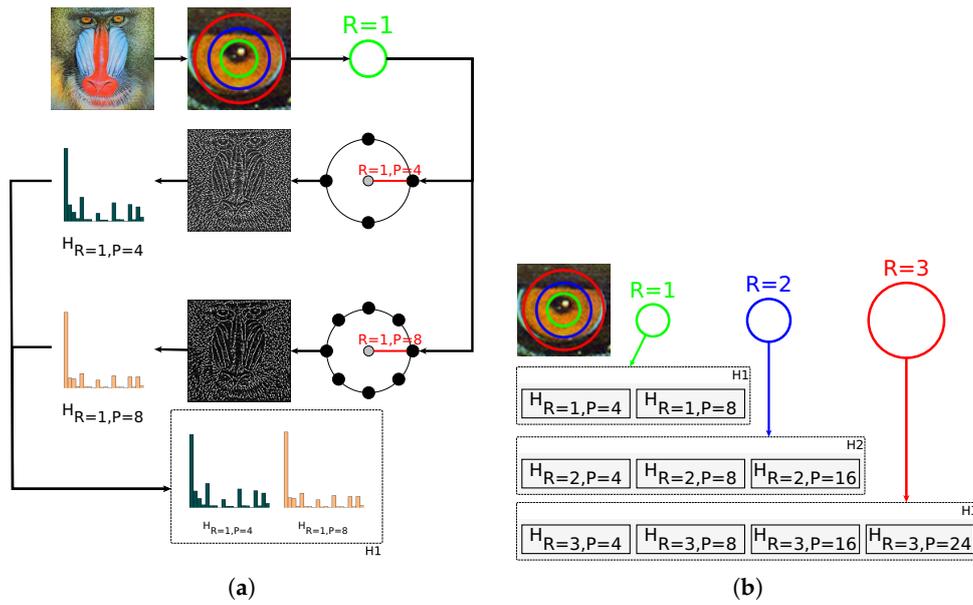


Figure 14. Feature extraction steps. (a) Multipoint LBP sampling. (b) Multiple histogram generation from LBP.

Figure 14a depicts the feature extraction for $R = 1$. The unitary radius generates only two distinct symmetrical patterns ($P = 4$ and $P = 8$). Each pattern generates a distinct LBP channel (see Figure 4). For a radius R , LBP maps are generated and combined:

$$L_R = \{LBP_{R,4}^u, LBP_{R,8}^u, LBP_{R,16}^u, \dots, LBP_{R,8R}^u\}, \quad (35)$$

where $LBP_{R,P}^u$ is computed according to Equation (6) and L_R contains $R + 1$ elements. From these LBP channels, the texture features are obtained by computing the histogram of each member of L_R :

$$H_{R,P} = [h_{R,P}(l_1), h_{R,P}(l_2), \dots, h_{R,P}(l_{P+2})], \quad (36)$$

where

$$h_{R,P}(l_i) = \sum_{x,y} \delta(LBP_{R,P}^u(x,y), i), \quad (37)$$

and

$$\delta(s, t) = \begin{cases} 1 & s = t, \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

In the above equations, (x, y) indicates the position of a given point of $LBP_{R,P}^u$ and l_i is the i -th LBP label. Notice that we are using ‘uniform’ LBP descriptors (Equation (6)) since their histograms provide a better discrimination of the texture properties.

To obtain the feature vector, we vary the radius, compute all possible symmetric LBP patterns and their histograms, as illustrated in Figure 14b. For a radius R , we generate a vector of histograms by concatenating all individual LBP histograms:

$$H_R = H_{R,4} \oplus H_{R,8} \oplus H_{R,16} \oplus \dots \oplus H_{R,8R}, \quad (39)$$

where \oplus denotes the concatenation descriptor.

The steps for computing the multiscale LBP histogram are summarized in Figure 15. For $R = N$, the final feature vector is generated by concatenating the histograms of the LBP channels with radius values smaller than N :

$$x = x_N = H_1 \oplus H_2 \oplus H_3 \oplus \dots \oplus H_N, \tag{40}$$

where $R = N$ is the maximum radius value and x_N is the feature vector used to compute the histogram.

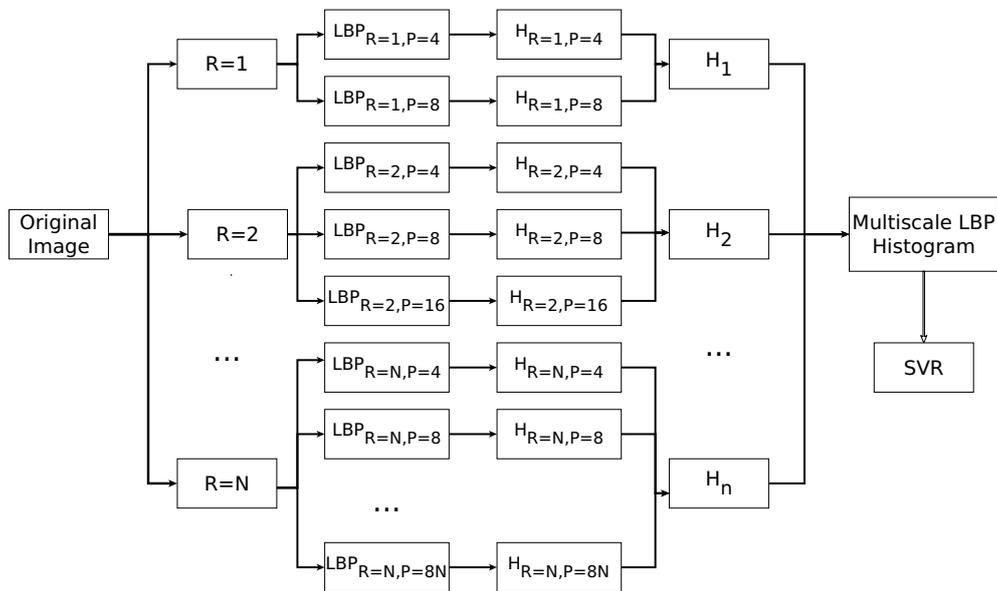


Figure 15. Feature extraction using MLBP histograms.

2.12. Multiscale Local Ternary Patterns (MLTP)

In general, the LTP threshold τ is adjusted for the target application. Anthimopoulos et al. [92] demonstrated that the τ values correspond to the gradient of the image. The choice of τ may affect the discrimination of edge and non-edge pixels, which is an important step in the texture analysis. We propose [93] an optimal set of thresholds to be used in the multilevel edge description operation, which make it possible to cluster gradient PDFs. The procedure is described as follows. First, the image gradients are fit using an exponential distribution:

$$PDF_e(z) = \lambda e^{-\lambda z}, \tag{41}$$

where λ is the rate parameter of the distribution. Then, the average value of the image gradient λ^{-1} is computed. The inverse cumulative distribution function of PDF_e is, then, obtained using the following equation:

$$F_e(\Delta_i) = \lambda^{-1} \ln(1 - \Delta_i), \tag{42}$$

where:

$$\Delta_i = \frac{i}{L+1}, \Delta_i \in [0, 1) \tag{43}$$

and $i \in \{1, 2, \dots, L\}$ and L is the number of levels. To select a threshold, we take

$$\tau_i = F_e(\Delta_i) \tag{44}$$

for equally spaced values of Δ_j .

The feature extraction process is illustrated in Figure 16. We decompose the image into LTP channels. These channels are generated by varying the τ values according to Equations (42)–(44). Since for a single image the LTP descriptor produces two channels, for L numbers of τ_i , $2L$ LTP channels

are produced. For example, in Figure 17, we use $L = 4$, generating eight distinct LTP channels. In the proposed LTP approach, instead of computing the differences between t_c and its neighbors on the grayscale image, we take the maximum difference on the R, G, or B channels.

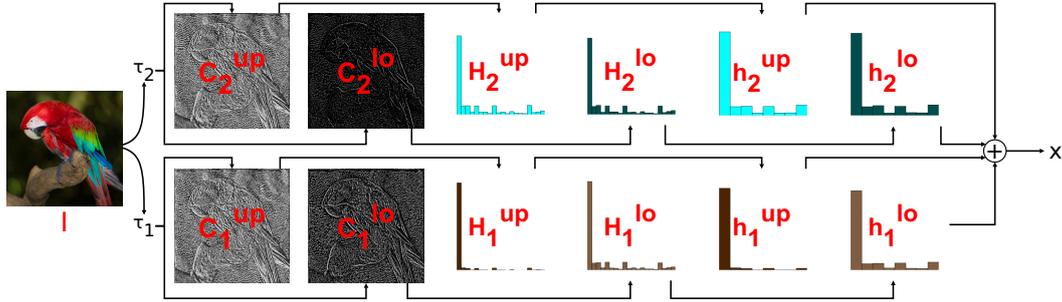


Figure 16. Illustration of process of extracting the feature vector x with $L = 2$.

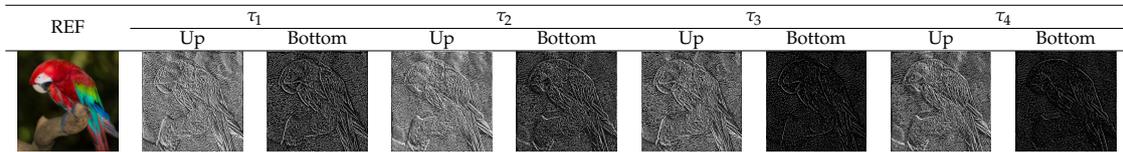


Figure 17. The reference image and its upper and lower patterns generated using the Local Ternary Pattern (LTP) descriptor with four different threshold values.

After the aforementioned steps are completed, we obtain a set of LTP channels with $2 \times L$ elements: $\{C_1^{up}, C_1^{lo}, C_2^{up}, C_2^{lo}, \dots, C_L^{up}, C_L^{lo}\}$. In this set, the subscript index corresponds to the i -th τ value, while the superscript index indicates whether the element is an upper (up) or lower (lo) pattern. For each LTP channel C_i^j , where $j \in \{up, lo\}$, we compute the corresponding LTP histogram H_i^j . These histograms are used to build the feature vector. If we simply concatenate these histograms, we generate a feature vector with a $2^P \times 2 \times L$ dimension. Depending on the L and P parameters, the number of features can be very high, what has a direct impact on the performance of the proposed algorithm.

In order to limit the number of dimensions, the number of bins of the LTP histograms is reduced according to the following formula:

$$k_i^j = \left\lfloor \frac{\max H_i^j - \min H_i^j}{n} \right\rfloor, \tag{45}$$

where $\lfloor \cdot \rfloor$ is the operation of rounding to the nearest integer, n defines the number of equal-width bins in the given range, and k_i^j is the reduced number of bins of histogram H_i^j . After this quantization, we acquire a set of quantized histograms $\{h_1^{up}, h_1^{lo}, h_2^{up}, h_2^{lo}, \dots, h_L^{up}, h_L^{lo}\}$. This new set is used to generate the feature vector associated with the image I . More specifically, the feature vector x is generated by concatenating the quantized histograms h_i^j , i.e.,

$$\check{x} = h_1^{up} \oplus h_1^{lo} \oplus h_2^{up} \oplus h_2^{lo} \oplus \dots \oplus h_L^{up} \oplus h_L^{lo}, \tag{46}$$

where \oplus is the concatenation descriptor and x is the feature vector.

2.13. Local Variance Patterns (LVP)

The Local Variance Pattern (LVP) is an extension of the LBP descriptor proposed in this work. This descriptor was developed specifically for quality assessment tasks. The LVP descriptor computes the texture local energy using the following formula:

$$LVP_{R,P}^u(I_c) = \left\lceil \frac{P \cdot V_{R,P}(I_c) - [LBP_{R,P}(I_c)]^2}{P^2} \right\rceil, \tag{47}$$

where:

$$V_{R,P}(I_c) = \sum_{p=0}^{P-1} [S(I_p - I_c) \cdot 2^p]^2. \tag{48}$$

LVP descriptor estimates the spread of the texture local energy. By measuring the texture energy, the LVP descriptor is able to estimate the effect that specific impairments have on the texture. For example, a Gaussian blurring impairment decreases the local texture energy, while a noise impairment increases it. Figure 18 shows a comparison of the steps used to extract texture information using the LBP and LVP descriptors, assuming that $R = 1$ and $P = 8$. The numbers in the yellow squares represent the order in which the steps are computed. The LBP descriptor generates two possible values (see Equation (2)), which are represented by the colors white ($S(t) = 1$) and black ($S(t) = 0$). Next, we use Equation (6) to compute the LBP label and Equation (47) to compute the LVP label.

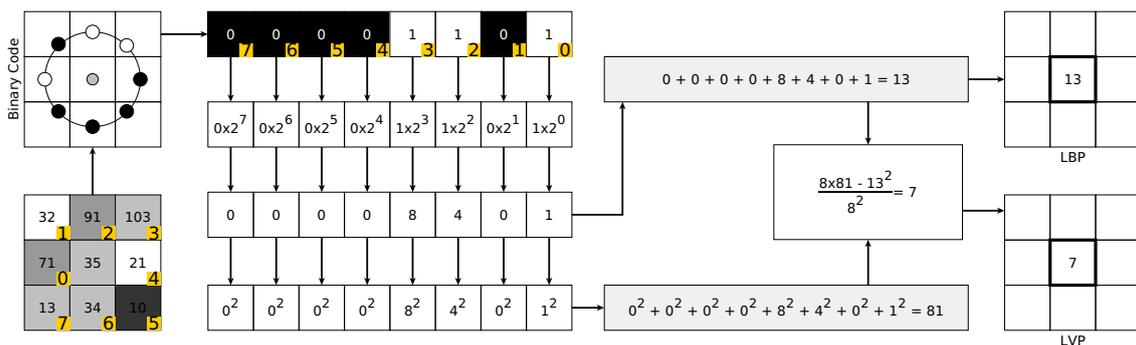


Figure 18. Pattern extraction process for a given pixel using LBP and LVP descriptors with $R = 1$, $P = 8$, $t_c = 35$, and $t_p = \{71, 32, 91, 103, 21, 10, 34, 13\}$.

After computing the LBP and LVP labels for all pixels of a given image, we obtain two channels for each image. These channels, C_{LBP} and C_{LVP} , correspond to the LBP and LVP patterns, respectively. Examples of these channels are shown in Figure 19. The first row of this figure shows the unimpaired reference image and three impaired images, degraded with different types of distortions. The second and third rows show the C_{LBP} and C_{LVP} channels for each image, respectively. Observing the C_{LBP} and C_{LVP} patterns in Figure 19, we notice that textures are affected differently by the different impairments. Comparing the C_{LBP} channels corresponding to the noisy, blurry, and jpeg2k compressed images (2nd line of Figure 19), we can notice that they are very different among themselves. The C_{LBP} channels corresponding to the blurry and jpeg2k images are also very different from the C_{LBP} channel corresponding to the reference (unimpaired) image. Nevertheless, the C_{LBP} channel corresponding to the noisy and reference images are visually similar. This similarity makes it difficult to discriminate between unimpaired and impaired images, what affects the quality prediction. Nevertheless, the C_{LVP} channels clearly show the differences between impaired and reference images, as can be seen in the 3rd line of Figure 19.

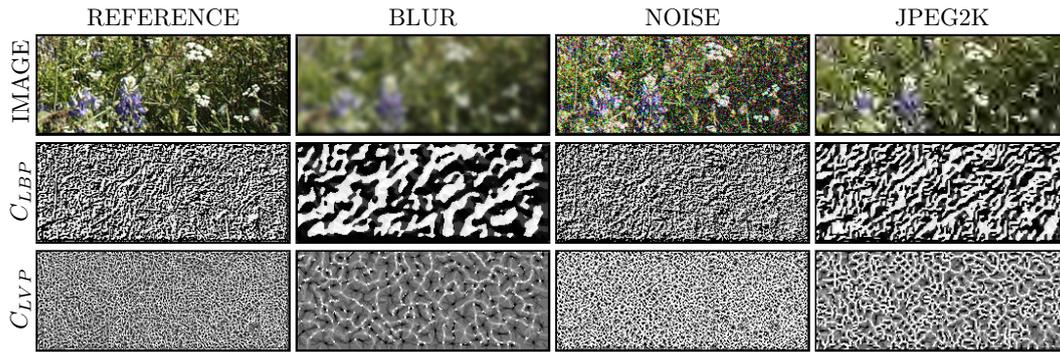


Figure 19. Reference image, its impaired versions, and their respective LBP and LVP maps (C_{LBP} and C_{LVP}).

2.14. Orthogonal Color Planes Patterns (OCPP)

The Orthogonal Color Planes Pattern (OCPP) descriptor extends the LBP to make it more sensitive to color and contrast distortions. Consider a pixel $\tau_c = \mathcal{J}(x, y, z)$ of a tri-dimensional (XYZ) color image \mathcal{J} . This image 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. In this work, we concatenate the LBP descriptors corresponding to the XY, XZ, and YZ planes to build an orthogonal color planes pattern (OCPP) texture descriptor.

As can be noticed from the aforementioned formulation, the LBP descriptor corresponding to the XY, XZ, and YZ planes can be computed independently to generate the three LBP maps: LBP_{XY} , LBP_{XZ} , and LBP_{YZ} . But, 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. Figure 20a illustrates how the points along the tri-dimensional HSV color space are sampled, while Figure 20b–d illustrate how each of the XY, XZ, and YZ planes are sampled.

Considering $R_Z = 1$ and $R_X = R_Y = R$, the coordinates of the neighboring points in the XY, XZ, and YZ orthogonal planes are given by:

$$x_{XY} = x + R \cos\left(2\pi \frac{p_{XY}}{P_{XY}}\right) \qquad y_{XY} = y - R \sin\left(2\pi \frac{p_{XY}}{P_{XY}}\right),$$

$$x_{XZ} = x + R \cos\left(2\pi \frac{p_{XZ}}{P_{XZ}}\right) \qquad z_{XZ} = z - \sin\left(2\pi \frac{p_{XZ}}{P_{XZ}}\right),$$

and

$$y_{YZ} = y + R \cos\left(2\pi \frac{p_{YZ}}{P_{YZ}}\right) \qquad z_{YZ} = z - \sin\left(2\pi \frac{p_{YZ}}{P_{YZ}}\right).$$

We compute the LBP for each plane using the following equations:

$$\mathfrak{L}_{XY} = LBP_R^{P_{XY}}(\tau_c) = \sum_{p_{XY}=0}^{P_{XY}-1} S(\tau_c - \tau_{XY})2^{p_{XY}},$$

$$\mathfrak{L}_{XZ} = LBP_R^{P_{XZ}}(\tau_c) = \sum_{p_{XZ}=0}^{P_{XZ}-1} S(\tau_c - \tau_{XZ})2^{p_{XZ}},$$

and

$$\mathcal{L}_{YZ} = LBP_R^{P_{YZ}}(\tau_c) = \sum_{p_{YZ}=0}^{P_{YZ}-1} S(\tau_c - \tau_{YZ})2^{p_{YZ}}.$$

The OCPP descriptor is built by concatenating these individual LBP descriptors:

$$OCPP_R^P(\tau_c) = [\mathcal{L}_{XY}, \mathcal{L}_{XZ}, \mathcal{L}_{YZ}]^T. \tag{49}$$

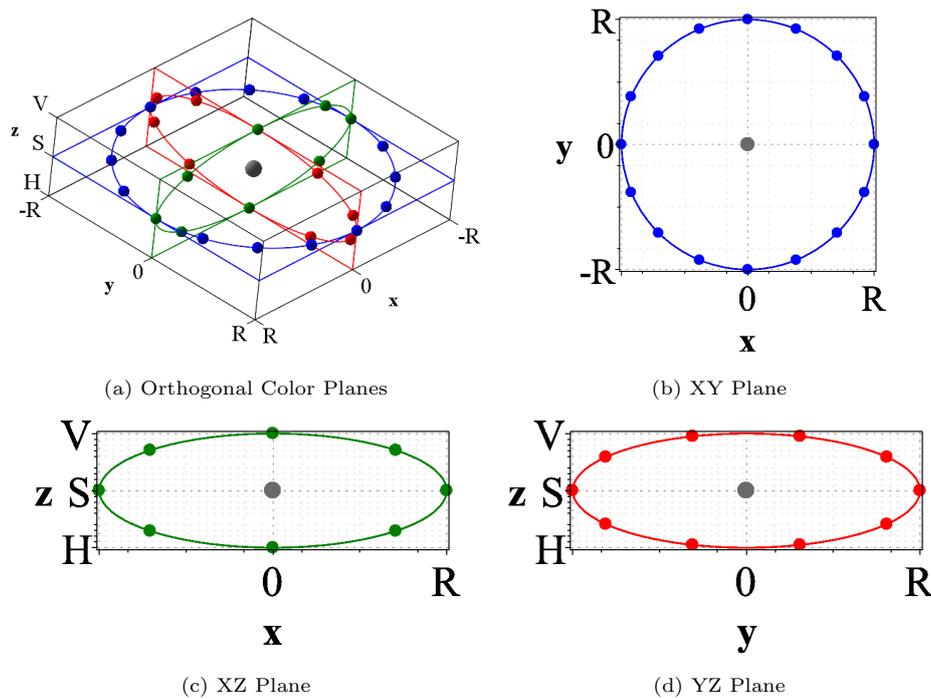


Figure 20. (a) General view of OCPP, (b) XY ($P_{XY} = 16$) plane, (c) XZ ($P_{XZ} = 8$) plane, and (d) YZ ($P_{YZ} = 10$) plane.

2.15. Salient Local Binary Patterns (SLBP)

The salient local binary pattern (SLBP) is an extension of the LBP which is designed to be used in image quality assessment methods. The descriptor incorporates visual salient information, given that recent results show that visual attention models improve the performance of visual quality assessment methods [52,58].

To estimate the saliency of the different areas of an image I , we use a computational visual attention model. More specifically, to keep the computational complexity low, we chose the Boolean map-based saliency (BMS) model [94]. When compared with other state-of-the-art visual attention models, BMS is noticeably faster, while still providing a good performance.

After computing the LBP descriptor of all pixels of image I , we obtain a LBP map \mathcal{L} , where each $\mathcal{L}[x, y]$ gives the local texture associated to the pixel $I[x, y]$. Similarly, the output of BMS is a saliency map \mathcal{W} , where each element $\mathcal{W}[x, y]$ corresponds to the probability that the pixel $I[x, y]$ attracts the attention of a human observer. The first, second, and third columns of Figure 21 depict a set of original images I , their corresponding LBP maps \mathcal{L} , and their corresponding saliency maps \mathcal{W} , respectively.

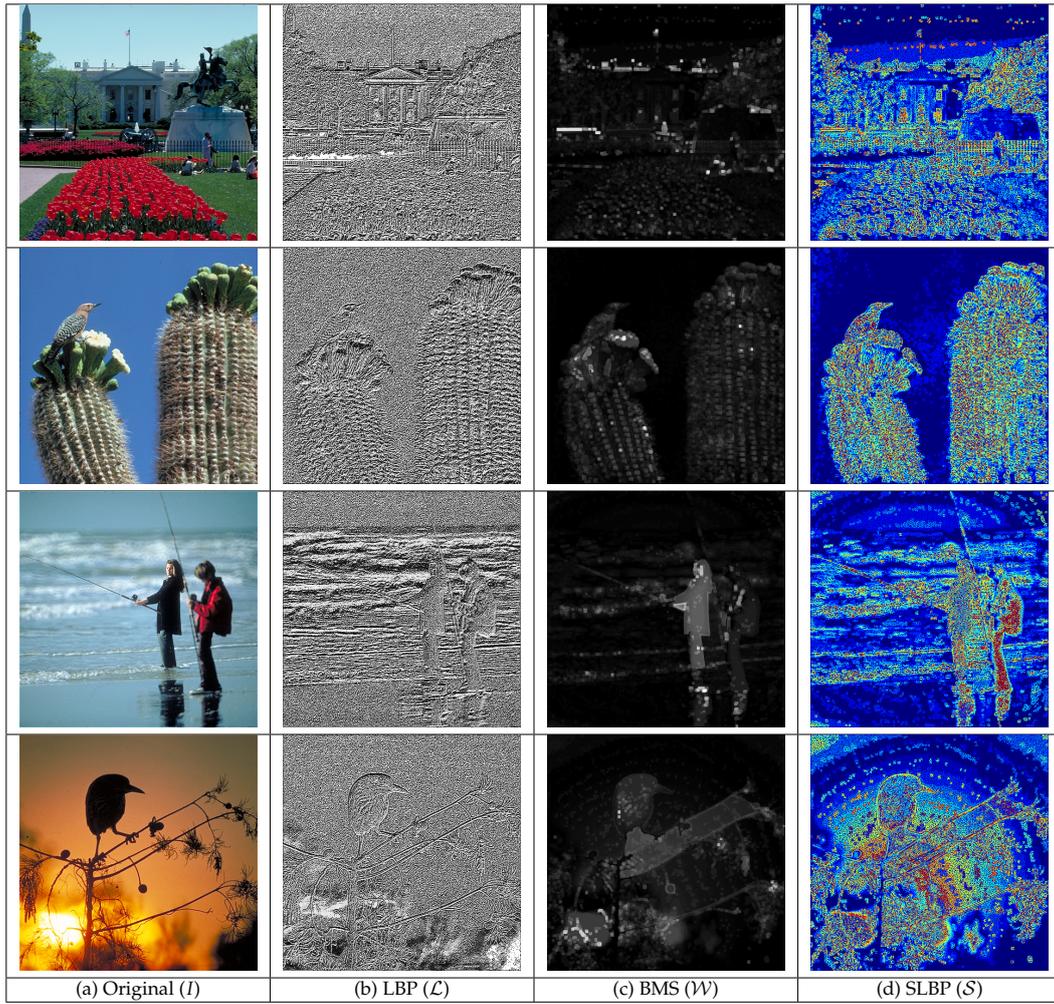


Figure 21. Example of original images (a), their LBP (b), BMS (c), and SLBP (d) maps.

We generate the feature vector by computing the histogram of \mathcal{L} weighted by \mathcal{W} . The histogram $\mathbf{H} = \{h[0], h[1], \dots, h[P + 1]\}$ is given by the following expression:

$$h[\phi] = \sum_i \sum_j \mathcal{W}[i, j] \Delta(\mathcal{L}[i, j], \phi), \tag{50}$$

where

$$\Delta(v, u) = \begin{cases} 1 & v = u, \\ 0 & \text{otherwise.} \end{cases} \tag{51}$$

The number of bins of this histogram is similar to the number of distinct LBP patterns of \mathcal{L} . So, we can remap each $\mathcal{L}[i, j]$ to its weighted form, generating the map \mathcal{S} displayed in Figure 21d. This figure depicts a heatmap representing the importance of each local texture. We name this weighted LBP map as the “Salient Local Binary Patterns” (SLBP).

2.16. Multiscale Salient Local Binary Patterns (MSLBP)

The multiscale salient local binary patterns (MSLBP) is an extension of SLBP in combination with MLBP. The idea behind MSLBP is to achieve fine information about frame texture by varying the parameters of LBP and combining the multiple generated LBP maps with saliency maps. In other words, we variate the SLBP to obtain multiple maps, as illustrated in Figure 22. For each combination of radius (R) and sampled points (P), we have an associated histogram $H_{R,P}$.

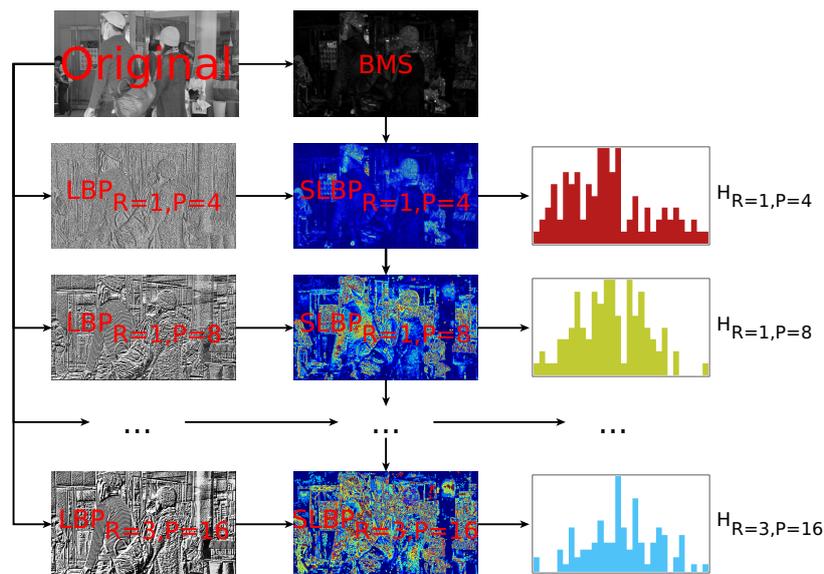


Figure 22. Multiple histogram generation from SLBP.

3. No-Reference Image Quality Assessment Using Texture Descriptors

In the previous section, we presented a series of texture descriptors. Most of them were designed for pattern recognition and computer vision applications. We also presented a set of proposed descriptors (MLBP, MLTP, LVP, OCPP, and SLBP), which were specially designed for visual quality assessment. Our goal is to investigate which descriptors are more suitable for no-reference (blind) image quality assessment (NR-IQA) methods. Moreover, we are interested in the relation between the type of descriptor and the performance accuracy of the IQA method.

3.1. Training and Testing Stages

Figure 23 depicts the training stage of the set of IQA methods proposed in this work. First, we collect subjective scores corresponding to each image of a training set. This procedure generates a set of labeled images, where each training set entry is composed by a pair of an image marker and its associated MOS (mean observer score). In other words, for the k -th unlabeled image I_k the algorithm associates a real value v_k , which corresponds to the overall quality of I_k .

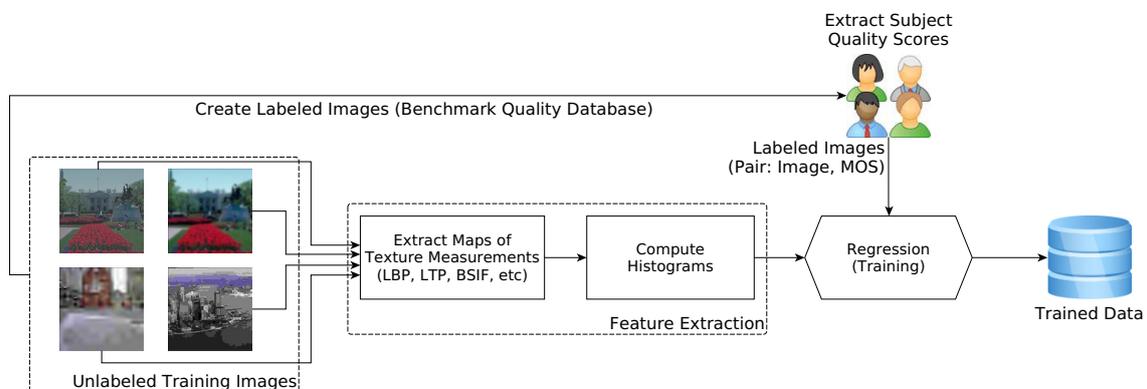


Figure 23. Training the quality metric.

After generating the labeled database formed by the set of pairs (I_k, v_k) , the features are extracted in order to generate the IQA model. For each image I_k , we compute the histogram of the given LBP

variant H_k and concatenate all histograms to produce the feature vector. Therefore, the training data is composed by the set (H_k, v_k) . The model is created using (H_k, v_k) , which is formed by a matrix $H \in \mathbb{R}^{K \times Q}$ and a vector $v \in \mathbb{R}^{1 \times K}$. In this case, K is the number of training entries (rows of H) and Q is the number of features (columns and the numbers of bins of H_k).

The prediction model is built using a regression model. This model maps each H_k into a real value v_k that predicts a corresponding quality score. The chosen regression model is the random forest (RF) regressor [61]. RF was chosen based on the results of Fernandez-Delgado et al. [95], which conducted an exhaustive evaluation of several machine learning methods and concluded that the best results are achieved by a family of RF methods.

The quality assessment task is depicted in Figure 24. After generating the prediction model, the image quality can be estimated using the model trained in the previous stage. The procedure is the same used for the images in the training set. In other words, the same feature (LBP histogram) is computed using the test image as input and, using this feature, the trained model predicts the quality score.

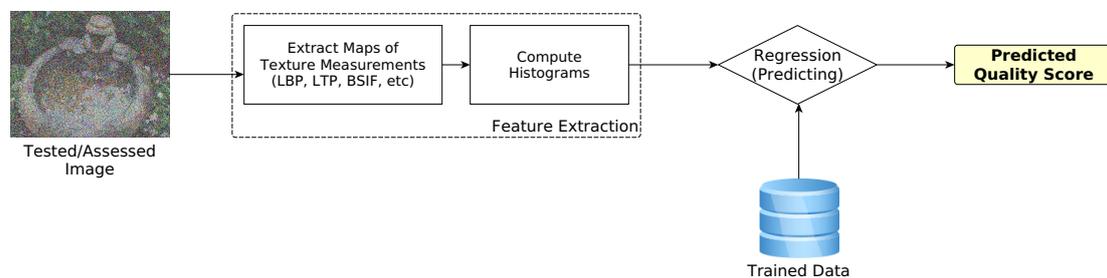


Figure 24. Predicting quality scores.

3.2. Test Setup

Results were generated using an Intel i7-4790 processor at 3.60 GHz. To assess the performance of the proposed NR-IQA method, we compute the Spearman's Rank Ordered Correlation (SROCC) between the mean opinion scores (MOS) and the predicted scores. Although other correlation coefficients (such as KRCC and PCC) can be added in the analysis, we decided to report the results using only the SROCC to prevent this article from becoming too lengthy. The proposed method is compared with the fastest state-of-the-art NR-IQA methods, including BRISQUE [42], CORNIA [96], CQA [97], SSEQ [98], and LTP [93]. These methods were chosen because they are all based on machine learning techniques, making the comparison with the proposed method straightforward. Moreover, the codes of these methods are publicly available for download.

For all machine learning NR-IQA methods, we use the same procedure for training and testing. In order to avoid overlapping between content detection and quality prediction, we divide the benchmark databases into content-independent training and testing subsets. Specifically, image content in the training subset was not used in the testing subset, and vice-versa. This division is made in a way that 80% of images are used for training and 20% are used for testing. This split is a common procedure used by several ML-based NR-IQA methods [42,96,97]. For the machine learning NR-IQA methods that are based on SVR, we use a LibSVR implementation accessed via Python interface and provided by the Sklearn library [99]. The optimal SVR meta parameters (C , γ , ν , etc.) are found using exhaustive grid search methods provided by Sklearn's API. No optimized search methods are used for the RF version of the proposed method.

The tests were performed using three image quality databases, which include subjective scores collected from psychophysical experiments. These databases are:

- LIVE2 [100] database has 982 test images, including 29 originals. This database includes 5 categories of distortions: JPEG, JPEG 2000 (JPEG2k), white noise (WN), Gaussian blur (GB), fast fading (FF).
- CSIQ [101] database has a total of 866 test images, consisting of 30 originals and 6 different categories of distortions. The distortions include JPEG, JPEG 2000 (JPEG2k), JPEG, white noise (WN), Gaussian blur (GB), fast fading (FF), global contrast decrements (CD), and additive Gaussian pink noise (PN).
- TID2013 [102] database contains 25 reference images with the following distortions: Additive Gaussian noise (AGN), Additive noise in color components (AGC), Spatially correlated noise (SCN), Masked noise (MN), High frequency noise (HFN), Impulse noise (IN), Quantization noise (QN), Gaussian blur (GB), Image denoising (ID), JPEG, JPEG2k, JPEG transmission errors (JPEGTE), JPEG2k transmission errors (JPEG2kTE), Non eccentricity pattern noise (NEPN), Local block-wise distortions (LBD), Intensity shift (IS), Contrast change (CC), Change of color saturation (CCS), Multiplicative Gaussian noise (MGN), Comfort noise (CN), Lossy compression (LC), Image color quantization with dither (ICQ), Chromatic aberration (CA), and Sparse sampling and reconstruction (SSR).

3.3. Results for Basic Descriptor with Varying Parameters

In order to test the LBP and its variants, we vary some parameters of each algorithm. Specifically, we vary the parameters of LBP, BSIF, CLBP, and LPQ. For the other tested variants, we choose the parameters $R = 1$ and $P = 8$. Table 1 depicts the parameters used by the tested algorithms.

To investigate the suitability of the basic LBP descriptor, we vary the parameters R and P using the Rotation Invariant LBP (LBP^r), the Uniform LBP (LBP^u), and the Uniform LBP with Rotation Invariance (LBP^{riu2}), which are described in Section 2.1. Figure 25 depicts the distribution of SROCC over simulations on the general case (i.e., when all distortions are considered). Table 2 shows the average SROCC correlation values for 100 simulations following the aforementioned protocol. In this table, STD represents the standard deviation and Δ is the subtraction between the maximum and minimum value in a given row or column.

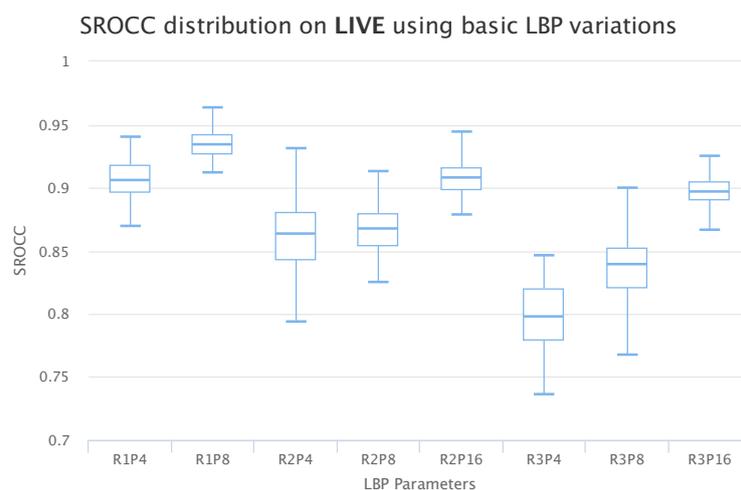


Figure 25. SROCC distribution on LIVE2 using basic LBP.

Table 1. Tested LBP variants.

Abbreviation	Name	Parameters
LBP ^{ri}	Basic Local Binary Patterns with rotation invariance	Radius (R) and number of neighbors (P)
LBP ^u	Uniform Local Binary Patterns	Radius (R) and number of neighbors (P)
LBP ^{riu2}	Uniform Local Binary Patterns with rotation invariance	Radius (R) and number of neighbors (P)
BSIF	Binarized Statistical Image Features	Window size and number of bits
LPQ	Local Phase Quantization)	Local frequency estimation
CLBP	Complete Local Binary Patterns	CLBP-S, CLBP-C, and CLBP-M
LCP	Local Configuration Patterns	Radius (R) and number of neighbors (P)
LTP	Local Ternary Patterns	Threshold (τ), Radius (R) and number of neighbors (P)
RLBP	Rotated Local Binary Patterns	Radius (R) and number of neighbors (P)
TPLBP	Three-Patch Local Binary Patterns	Patch size (w), Radius (R), and angle between neighboring patches
FPLBP	Four-Patch Local Binary Patterns	Patch size (w), Radius of first ring (R1), Radius of second ring (R2), and angle between neighboring patches
LVP	Local Variance Patterns	Radius (R) and number of neighbors (P)
OCLBP	Opposite Color Local Binary Patterns	Radius (R) and number of neighbors (P)
OCPP	Orthogonal Color Planes Patterns	Radius (R) and number of neighbors (P)
SLBP	Salient Local Binary Patterns	Radius (R) and number of neighbors (P)
MLBP	Multiscale Local Binary Patterns	Multiple values of Radius (R) and number of neighbors (P)
MLTP	Multiscale Local Ternary Patterns	Multiple values of Radius (R) and number of neighbors (P)
MSLBP	Multiscale Salient Local Binary Patterns	Multiple values of Radius (R) and number of neighbors (P)

Table 2. Average SROCC of simulations on tested image databases using basic LBP variations.

DB	DIST	LBP ^{ri}									LBP ^{rii}						LBP ^{riu2}											
		R = 1			R = 2			R = 3			R = 1			R = 2			R = 3			R = 1			R = 2			R = 3		
		P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16	P = 4	P = 8	P = 16
LIVE 2	JPEG	0.8959	0.9306	0.8238	0.9058	0.9124	0.7759	0.8683	0.9065	0.8921	0.9275	0.8376	0.9063	0.9176	0.8176	0.8301	0.9069	0.8955	0.9204	0.8343	0.8481	0.8906	0.7716	0.7971	0.8813			
	JPEG2k	0.9062	0.9423	0.8772	0.9161	0.9324	0.7812	0.8999	0.9238	0.9056	0.9353	0.8691	0.9149	0.9277	0.8181	0.8464	0.9023	0.9088	0.9245	0.8742	0.8724	0.8895	0.7857	0.8241	0.8816			
	WN	0.9753	0.9794	0.9521	0.9671	0.9694	0.9309	0.9553	0.9676	0.9743	0.9782	0.9356	0.9661	0.9703	0.9285	0.9465	0.9687	0.9753	0.9771	0.9538	0.9607	0.9661	0.9294	0.9407	0.9642			
	GB	0.9123	0.9621	0.9169	0.9474	0.9551	0.8873	0.9331	0.9479	0.9253	0.9611	0.9168	0.9494	0.9632	0.8771	0.9349	0.9666	0.9137	0.9481	0.9197	0.9144	0.9317	0.8808	0.8946	0.9134			
	FF	0.8341	0.8871	0.7878	0.8687	0.9054	0.6459	0.8027	0.8539	0.8521	0.8755	0.7692	0.8493	0.9026	0.6588	0.6756	0.8714	0.8325	0.8974	0.7821	0.7959	0.8755	0.6488	0.7585	0.8672			
	ALL	0.9015	0.9532	0.8713	0.9288	0.9422	0.8038	0.8988	0.9274	0.9101	0.9417	0.8631	0.9235	0.9427	0.8208	0.8501	0.9282	0.9048	0.9366	0.8704	0.8826	0.9174	0.8034	0.8493	0.9079			
CSIQ	JPEG	0.8245	0.8861	0.8135	0.8908	0.8705	0.8142	0.8682	0.8806	0.8241	0.8912	0.8513	0.8631	0.8725	0.8446	0.8506	0.8701	0.8176	0.8521	0.8073	0.8518	0.8642	0.8083	0.8323	0.8688			
	JPEG2k	0.7695	0.8532	0.7867	0.8379	0.8414	0.6964	0.8272	0.8339	0.7654	0.8266	0.7658	0.8065	0.8123	0.7025	0.7452	0.7977	0.7699	0.7851	0.7738	0.7571	0.7625	0.6844	0.7063	0.7524			
	WN	0.7079	0.8452	0.6404	0.7926	0.8229	0.5241	0.7984	0.8905	0.6328	0.9133	0.7658	0.7185	0.7499	0.6801	0.7176	0.6588	0.7149	0.8173	0.6403	0.6615	0.7428	0.6793	0.7031	0.6704			
	GB	0.8592	0.9078	0.8378	0.8891	0.9125	0.7889	0.8808	0.9141	0.8669	0.8856	0.8273	0.8738	0.8972	0.7946	0.8455	0.8873	0.8547	0.8923	0.8335	0.8718	0.8778	0.7969	0.8457	0.8828			
	PN	0.5786	0.8827	0.5289	0.8333	0.8768	0.6654	0.7331	0.8541	0.7821	0.8511	0.6184	0.7446	0.7648	0.5857	0.6698	0.6801	0.5735	0.8258	0.5323	0.7571	0.7191	0.5238	0.6301	0.6729			
	CD	0.3066	0.5901	0.3159	0.4791	0.4968	0.2615	0.3857	0.4577	0.3884	0.4714	0.4561	0.2929	0.3536	0.4051	0.3607	0.3052	0.2661	0.3788	0.2967	0.3245	0.3145	0.2731	0.3976	0.2093			
ALL	0.6735	0.8278	0.6471	0.7946	0.8019	0.6274	0.7561	0.7961	0.6854	0.8028	0.6635	0.7341	0.7457	0.6365	0.7059	0.7086	0.6638	0.7718	0.6421	0.7091	0.7181	0.6211	0.6796	0.6861				
TID 2013	AGC	0.4781	0.6135	0.2353	0.1084	0.4713	0.3703	0.2131	0.3554	0.1954	0.3496	0.1742	0.1309	0.2519	0.2509	0.2154	0.2912	0.4607	0.3273	0.1975	0.1665	0.1469	0.3681	0.2061	0.2746			
	AGN	0.7861	0.7757	0.4346	0.5881	0.6799	0.5642	0.4426	0.6969	0.6201	0.6138	0.3626	0.2873	0.6726	0.5726	0.2207	0.4581	0.7619	0.5353	0.4434	0.4146	0.5673	0.5342	0.4753	0.5957			
	CA	0.2186	0.2052	0.3674	0.2211	0.2453	0.2967	0.2693	0.2061	0.2035	0.2186	0.2797	0.2216	0.2475	0.3032	0.2962	0.2771	0.2065	0.2407	0.4155	0.3651	0.2828	0.2505	0.3781	0.2939			
	CC	0.1287	0.1007	0.1178	0.1181	0.0869	0.0971	0.1476	0.1696	0.1284	0.1742	0.1131	0.0623	0.0957	0.1238	0.0607	0.0749	0.1551	0.1438	0.1161	0.0996	0.0773	0.0938	0.1098	0.1073			
	CCS	0.1891	0.1241	0.1666	0.1255	0.2309	0.1898	0.2131	0.1473	0.1751	0.1319	0.1938	0.2195	0.2903	0.1754	0.1881	0.2311	0.1699	0.1786	0.1684	0.1587	0.1599	0.1671	0.2374	0.1852			
	CN	0.3052	0.1979	0.1655	0.1425	0.3253	0.1959	0.1181	0.1384	0.1834	0.1851	0.1491	0.1384	0.1742	0.1465	0.1257	0.1842	0.3645	0.1473	0.1748	0.1467	0.1365	0.1701	0.2301	0.2325			
	GB	0.8216	0.8384	0.8041	0.8006	0.8122	0.7781	0.8208	0.8261	0.8139	0.8341	0.8027	0.8253	0.8038	0.7391	0.8152	0.8075	0.8073	0.8199	0.8023	0.8095	0.8253	0.7766	0.7969	0.8276			
	HFN	0.7934	0.8126	0.6968	0.7648	0.8365	0.7793	0.6121	0.8473	0.7901	0.7541	0.6431	0.6719	0.8648	0.7248	0.5231	0.7821	0.7891	0.6511	0.7048	0.6717	0.6701	0.7604	0.6415	0.7375			
	ICQ	0.7741	0.7715	0.7638	0.8246	0.7973	0.6748	0.8088	0.8196	0.7642	0.7633	0.7498	0.7904	0.8183	0.7099	0.7703	0.8173	0.7634	0.7908	0.7554	0.7911	0.8011	0.6383	0.7542	0.7818			
	ID	0.3503	0.8107	0.6211	0.7631	0.7238	0.6084	0.6892	0.6938	0.2738	0.5346	0.5523	0.7415	0.7919	0.5349	0.5742	0.7081	0.3534	0.4192	0.6384	0.6038	0.5019	0.5901	0.4749	0.4761			
	IN	0.1384	0.3423	0.1394	0.5396	0.5431	0.1327	0.3873	0.5954	0.1169	0.0932	0.1551	0.4252	0.4021	0.1269	0.2188	0.3059	0.1665	0.1384	0.1323	0.5894	0.4722	0.1202	0.2169	0.4401			
	IS	0.1378	0.0631	0.1201	0.0977	0.0692	0.1183	0.0795	0.0894	0.1068	0.0598	0.0995	0.0743	0.0659	0.1075	0.0742	0.1054	0.1652	0.0936	0.1322	0.1328	0.0982	0.1025	0.0866	0.1271			
	JPEG	0.7241	0.8392	0.6678	0.8016	0.7973	0.6265	0.7814	0.7861	0.6912	0.8035	0.6523	0.7615	0.7657	0.6311	0.6751	0.7448	0.6888	0.7519	0.6762	0.6631	0.6831	0.6431	0.6367	0.6941			
	JPEG2k	0.1273	0.2942	0.1361	0.3361	0.2784	0.1353	0.3007	0.2869	0.1434	0.1988	0.1261	0.3026	0.3599	0.1452	0.1092	0.2523	0.1707	0.1534	0.1351	0.2103	0.2803	0.1591	0.1453	0.1888			
	JPEG2k	0.7949	0.8669	0.6876	0.8057	0.8384	0.7751	0.8153	0.8373	0.8103	0.8057	0.8151	0.8511	0.8323	0.7634	0.8126	0.7996	0.7888	0.8411	0.8311	0.8218	0.8107	0.7673	0.7515	0.7673			
	JPEG2kTE	0.3888	0.5015	0.8326	0.6049	0.5934	0.5526	0.7203	0.7073	0.4142	0.4981	0.6149	0.7099	0.7131	0.5823	0.5888	0.7007	0.3765	0.4057	0.6853	0.6238	0.5121	0.5581	0.6584	0.5642			
	LBD	0.1634	0.1739	0.1462	0.1657	0.1175	0.1184	0.1442	0.1894	0.1502	0.1605	0.1569	0.1331	0.1332	0.1566	0.1411	0.1323	0.1753	0.1343	0.1263	0.1562	0.1392	0.1335	0.1288	0.1556			
	LC	0.4419	0.5581	0.2869	0.2731	0.4507	0.2769	0.2996	0.4807	0.1542	0.1515	0.1865	0.2107	0.1553	0.1901	0.1476	0.1769	0.3596	0.2734	0.3473	0.1519	0.1284	0.3092	0.2984	0.1146			
	MGN	0.6977	0.6947	0.5239	0.4977	0.7766	0.5871	0.3519	0.7002	0.5971	0.4191	0.2848	0.2084	0.4796	0.4731	0.1605	0.4014	0.7214	0.4916	0.5139	0.4658	0.4893	0.5911	0.4483	0.4081			
	MN	0.2677	0.4295	0.1952	0.3469	0.1832	0.1448	0.1501	0.1615	0.1667	0.3236	0.1531	0.1286	0.1398	0.1449	0.3087	0.1288	0.2438	0.2652	0.1631	0.1573	0.1319	0.1611	0.2252	0.1771			
	NEPN	0.1413	0.2054	0.2107	0.2358	0.3383	0.1611	0.2721	0.3708	0.1329	0.1273	0.1795	0.2862	0.2706	0.1391	0.2917	0.2094	0.1254	0.1416	0.1667	0.2787	0.3373	0.1533	0.2252	0.1996			
	QN	0.7733	0.8584	0.7871	0.8073	0.8353	0.7306	0.7965	0.8115	0.8254	0.8631	0.8069	0.8226	0.8757	0.7957	0.8019	0.8384	0.7769	0.8042	0.7772	0.7764	0.8053	0.7431	0.7828	0.8242			
	SCN	0.6399	0.6603	0.7103	0.6426	0.5357	0.5411	0.6003	0.6807	0.6111	0.6303	0.5681	0.6257	0.7496	0.5811	0.4169	0.6084	0.6673	0.6803	0.6965	0.5442	0.4238	0.5538	0.5457	0.6853			
	SSR	0.8246	0.8846	0.8151	0.8507	0.9142	0.7042	0.7911																				

From Table 2, we can notice that the basic LBP descriptor is suitable for predicting quality. This suitability is indicated by the high correlation indices obtained on LIVE2 database. On this database, the average SROCC vary from 0.8034 to 0.9532 in the general case, from 0.6459 to 0.9054 for the FF distortion, from 0.8771 to 0.9666 for the GB distortion, from 0.9285 to 0.9794 for the WN distortion, from 0.7812 to 0.9423 for the JPEG2k distortion, and from 0.7716 to 0.9306 for the JPEG distortion. These values suggest that basic LBP variations are well appropriate to model quality of images under WN and GB distortions. Regarding the LBP parameters, the prediction performance of WN and GB are less affected by these parameters when compared with other distortions (see the variance and Δ values).

Although the basic LBP works well for WN and GB distortions independently of its parameters, the performance for other distortions varies according to the parameters. This variation is also observed in CSIQ and TID2013 databases. For example, on the CSIQ database, the SROCC values varies from 0.8073 to 0.8912 in the best case (JPEG) and from 0.2093 to 0.5901 in the worst case (CD). These values indicate that the prediction performance is affected by the basic LBP parameters. Actually, this is the premise used by Freitas et al. [91], who assume that different parameters of LBP can be used to achieve a better performance. In their work, an aggregation of features obtained with different LBP parameters results in a more robust quality assessment model.

3.4. Results for Variants of Basic Descriptors

Once it has been demonstrated that basic LBP variants present a suitable descriptor to describe image quality, we check the performance of the other LBP variants described in Section 2. To perform the tests, we vary the parameters of BSIF, LPQ, and CLBP descriptors. For the remaining extensions (i.e., LCP, LTP, RLBP, TPLBP, FPLBP, LVP, OCLBP, OCPP, SLBP, MLBP, MLTP, and MSLBP), we do not vary the parameters. Figure 26 depicts the distribution of SROCC for the general case using the tested LBP variants (100 simulations).

To investigate the suitability of the basic BSIF descriptor, we performed the simulations by changing the patch size and the number of selected binarized feature (see Section 2.4). The results of the performed simulations on the LIVE2, CSIQ, and TID2013 databases are depicted in Table 3 respectively. Based on results of Table 3, we notice that BSIF is a valuable descriptor for IQA. In the LIVE2 database, the BSIF performs well for almost all configurations. However, the results are better for smaller patch sizes. In these cases, the average SROCC values are higher and have a low variance. As shown in Table 3, the performance of BSIF decreases for the CSIQ database. When compared with the LIVE2 database, the average SROCC values are lower and the variance is higher. The values in both Table 3 indicate that there is a relationship between the patch size and the number of bits. More specifically, the larger the patch size, the higher the number of bits required to obtain a good quality prediction. For example, in both LIVE2 and CSIQ databases, using a 3×3 patch, the best performance is obtained using 8 bits and the worst performance is obtained when only 5 bits are used.

Table 4 shows the results of simulations using seven different LPQ configurations, corresponding to different LPQ parameters. The main parameters of the LPQ descriptor are the size of the local window and the method used for local frequency estimation. The size of the local window was fixed on 3×3 and the tests were performed by varying the method used for local frequency estimation. The LPQ configurations are the following:

- C1: Short-term Fourier transform (STFT) with uniform window (basic version of LPQ);
- C2: STFT with Gaussian window;
- C3: Gaussian derivative quadrature filter pair;
- C4: STFT with uniform window + STFT with Gaussian window (concatenation of feature vectors produced by C1 and C2);
- C5: STFT with uniform window + STFT with Gaussian derivative quadrature filter pair (concatenation of feature vectors produced by C1 and C3);

- C6: STFT with Gaussian window + Gaussian derivative quadrature filter pair (concatenation of feature vectors produced by C2 and C3);
- C7: Concatenation of feature vectors produced by C1, C2, and C3.

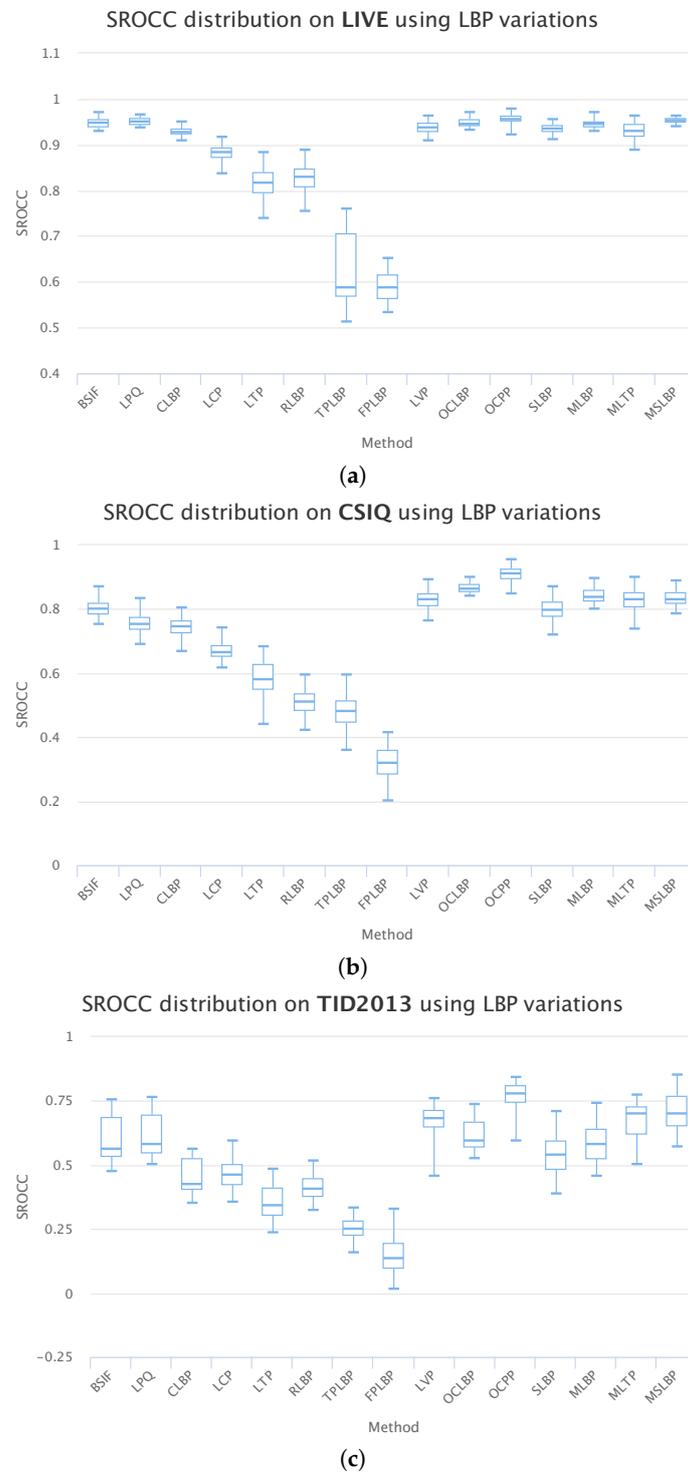


Figure 26. Distribution of average SROCC after 100 simulations using different LBP variations. (a) LIVE2. (b) CSIQ. (c) TID2013.

Table 3. Average SROCC of 1000 runs of simulations on tested databases using BSIF variations.

SIZE		3 × 3			5 × 5			7 × 7		Average	STD	MAX	MIN	Δ
BITS		5	6	7	8	9	10	11	12					
DB	DIST													
LIVE2	JPEG	0.8864	0.9015	0.8857	0.8931	0.8799	0.8969	0.8874	0.8670	0.8872	0.0107	0.9015	0.8670	0.0345
	JPEG2k	0.8803	0.9046	0.9019	0.8585	0.9059	0.8865	0.9138	0.8620	0.8892	0.0209	0.9138	0.8585	0.0553
	WN	0.9440	0.9590	0.9609	0.9614	0.9620	0.9630	0.9639	0.9469	0.9576	0.0077	0.9639	0.9440	0.0200
	GB	0.8644	0.9293	0.9203	0.9330	0.9369	0.9264	0.9505	0.9213	0.9228	0.0255	0.9505	0.8644	0.0862
	FF	0.8270	0.8208	0.8174	0.8075	0.8486	0.8587	0.8674	0.7728	0.8275	0.0306	0.8674	0.7728	0.0946
	ALL	0.8887	0.9116	0.9127	0.9099	0.9236	0.9251	0.9308	0.8947	0.9121	0.0147	0.9308	0.8887	0.0422
	Average	0.8818	0.9045	0.8998	0.8939	0.9095	0.9095	0.9190	0.8775					
	STD	0.0380	0.0462	0.0475	0.0549	0.0407	0.0365	0.0370	0.0605					
	MAX	0.9440	0.9590	0.9609	0.9614	0.9620	0.9630	0.9639	0.9469					
	MIN	0.8270	0.8208	0.8174	0.8075	0.8486	0.8587	0.8674	0.7728					
CSIQ	JPEG	0.8541	0.8638	0.8662	0.8780	0.8825	0.8800	0.8800	0.8687	0.8717	0.0100	0.8825	0.8541	0.0284
	JPEG2k	0.8040	0.8111	0.8549	0.8564	0.8105	0.8282	0.8237	0.8182	0.8259	0.0200	0.8564	0.8040	0.0525
	WN	0.5468	0.6649	0.7668	0.7882	0.8089	0.8217	0.8193	0.7754	0.7490	0.0959	0.8217	0.5468	0.2749
	GB	0.6983	0.7965	0.7871	0.8081	0.8799	0.8762	0.8790	0.8724	0.8247	0.0648	0.8799	0.6983	0.1816
	PN	0.3325	0.5391	0.6538	0.7990	0.7875	0.7663	0.7699	0.7460	0.6743	0.1633	0.7990	0.3325	0.4665
	CD	0.1550	0.1443	0.2428	0.2952	0.0741	0.0907	0.0771	0.0978	0.1471	0.0819	0.2952	0.0741	0.2210
	_ALL	0.5977	0.6904	0.7234	0.7664	0.7317	0.7325	0.7311	0.7074	0.7101	0.0504	0.7664	0.5977	0.1687
	Average	0.5698	0.6443	0.6993	0.7416	0.7107	0.7137	0.7115	0.6980					
	STD	0.2523	0.2459	0.2142	0.2007	0.2856	0.2799	0.2849	0.2716					
	MAX	0.8541	0.8638	0.8662	0.8780	0.8825	0.8800	0.8800	0.8724					
MIN	0.1550	0.1443	0.2428	0.2952	0.0741	0.0907	0.0771	0.0978						
TID	AGC	0.2599	0.2273	0.3868	0.4888	0.4042	0.3931	0.3923	0.4787	0.3789	0.0928	0.4888	0.2273	0.2615
	AGN	0.5046	0.5388	0.7250	0.7462	0.6615	0.6400	0.6808	0.6954	0.6490	0.0859	0.7462	0.5046	0.2415
	CA	0.5727	0.6720	0.6729	0.6771	0.5079	0.5057	0.5351	0.5824	0.5907	0.0741	0.6771	0.5057	0.1714
	CC	0.1219	0.0946	0.1185	0.1362	0.0815	0.0965	0.0885	0.0838	0.1027	0.0202	0.1362	0.0815	0.0546
	CCS	0.1431	0.1415	0.1435	0.1192	0.1881	0.1806	0.2246	0.1762	0.1646	0.0339	0.2246	0.1192	0.1054
	CN	0.1338	0.3165	0.2735	0.3769	0.2942	0.4215	0.4600	0.4838	0.3450	0.1149	0.4838	0.1338	0.3500
	GB	0.7546	0.8277	0.8154	0.8416	0.8832	0.8862	0.9051	0.8953	0.8511	0.0512	0.9051	0.7546	0.1505
	HFN	0.6580	0.7628	0.8047	0.8313	0.7757	0.7878	0.8131	0.8068	0.7800	0.0539	0.8313	0.6580	0.1733
	ICQ	0.7117	0.7777	0.7700	0.7939	0.7632	0.7742	0.7854	0.8123	0.7735	0.0293	0.8123	0.7117	0.1006
	ID	0.5627	0.6804	0.6946	0.6823	0.7338	0.7446	0.7538	0.7785	0.7038	0.0672	0.7785	0.5627	0.2158
	IN	0.4100	0.7385	0.6712	0.7008	0.7762	0.7592	0.7714	0.6995	0.6909	0.1196	0.7762	0.4100	0.3662
	IS	0.1142	0.1092	0.1146	0.0938	0.1291	0.1435	0.1689	0.1519	0.1281	0.0250	0.1689	0.0938	0.0750
	JPEG	0.7625	0.8168	0.7697	0.7708	0.8177	0.8026	0.8091	0.8215	0.7963	0.0246	0.8215	0.7625	0.0591
	JPEGTE	0.1048	0.3770	0.4231	0.4723	0.4750	0.4908	0.5752	0.5000	0.4273	0.1424	0.5752	0.1048	0.4704
	JPEG2k	0.7622	0.8362	0.7923	0.8208	0.8208	0.8246	0.8115	0.8445	0.8141	0.0262	0.8445	0.7622	0.0823
	JPEG2kTE	0.3362	0.4045	0.3646	0.5250	0.5476	0.6743	0.6922	0.7746	0.5399	0.1635	0.7746	0.3362	0.4385
	LBD	0.2808	0.2864	0.2968	0.3775	0.3342	0.3300	0.3343	0.3652	0.3257	0.0355	0.3775	0.2808	0.0967
	LC	0.2569	0.2731	0.4777	0.5323	0.5796	0.6315	0.6300	0.6565	0.5047	0.1590	0.6565	0.2569	0.3996
	MGN	0.3754	0.5173	0.6692	0.6924	0.6508	0.6469	0.6527	0.6919	0.6121	0.1105	0.6924	0.3754	0.3170
	MN	0.1833	0.3278	0.2337	0.3591	0.1812	0.1862	0.1812	0.1658	0.2273	0.0748	0.3591	0.1658	0.1933
	NEPN	0.1201	0.1344	0.1456	0.1400	0.2383	0.2610	0.2569	0.2091	0.1882	0.0593	0.2610	0.1201	0.1410
	QN	0.6454	0.7046	0.7615	0.7469	0.7281	0.7001	0.7296	0.7585	0.7218	0.0383	0.7615	0.6454	0.1162
	SCN	0.4627	0.6238	0.6904	0.7138	0.8215	0.8069	0.8131	0.8815	0.7267	0.1356	0.8815	0.4627	0.4188
	SSR	0.7231	0.7962	0.8700	0.9108	0.8823	0.8938	0.9192	0.9008	0.8620	0.0679	0.9192	0.7231	0.1962
	_ALL	0.4252	0.5364	0.5809	0.6177	0.5965	0.6126	0.6247	0.5964	0.5738	0.0661	0.6247	0.4252	0.1995
Average	0.4154	0.5009	0.5306	0.5667	0.5549	0.5678	0.5843	0.5924						
STD	0.2372	0.2559	0.2560	0.2495	0.2566	0.2501	0.2509	0.2624						
MAX	0.7625	0.8362	0.8700	0.9108	0.8832	0.8938	0.9192	0.9008						
MIN	0.1048	0.0946	0.1146	0.0938	0.0815	0.0965	0.0885	0.0838						

Table 4 shows that the performance of LPQ is high for the LIVE2 database, with mean SROCC values above 0.9 for all distortions, independently of the configuration. The low variance and the high average value of the SROCC values for the LIVE2 indicate that LPQ is a suitable descriptor for measuring the quality of JPEG, JPEG2k, WN, GB, and FF distortions. However, the performance of the prediction decreases for the CSIQ and TID2013 databases. This is probably due to the presence of the contrast and color distortions on the CSIQ and TID2013 databases.

Table 4. Average SROCC of 1000 runs of simulations on tested databases using LPQ variations.

DB	DIST	C1	C2	C3	C4	C5	C6	C7	Average	STD	MAX	MIN	Δ
LIVE2	JPEG	0.8999	0.9324	0.9140	0.9186	0.9130	0.9197	0.9174	0.9164	0.0097	0.9324	0.8999	0.0325
	JPEG2k	0.8865	0.8916	0.8832	0.8797	0.8900	0.8828	0.8841	0.8854	0.0042	0.8916	0.8797	0.0119
	WN	0.9445	0.9605	0.9323	0.9566	0.9422	0.9565	0.9556	0.9498	0.0102	0.9605	0.9323	0.0282
	GB	0.8924	0.9126	0.9042	0.9116	0.9037	0.9258	0.9231	0.9105	0.0116	0.9258	0.8924	0.0333
	FF	0.8659	0.8536	0.8394	0.8596	0.8523	0.8450	0.8577	0.8533	0.0090	0.8659	0.8394	0.0265
	ALL	0.9051	0.9141	0.8998	0.9149	0.9047	0.9163	0.9167	0.9102	0.0069	0.9167	0.8998	0.0169
	Average	0.8991	0.9108	0.8955	0.9068	0.9010	0.9077	0.9091					
	STD	0.0261	0.0363	0.0319	0.0337	0.0295	0.0387	0.0339					
	MAX	0.9445	0.9605	0.9323	0.9566	0.9422	0.9565	0.9556					
	MIN	0.8659	0.8536	0.8394	0.8596	0.8523	0.8450	0.8577					
CSIQ	JPEG	0.8415	0.8801	0.8651	0.8701	0.8538	0.8812	0.8706	0.8660	0.0142	0.8812	0.8415	0.0397
	JPEG2k	0.7323	0.7677	0.8172	0.7698	0.8029	0.8029	0.7948	0.7839	0.0291	0.8172	0.7323	0.0849
	WN	0.3586	0.5805	0.5658	0.6008	0.5554	0.6505	0.6372	0.5641	0.0972	0.6505	0.3586	0.2919
	GB	0.8047	0.8483	0.8343	0.8647	0.8305	0.8723	0.8687	0.8462	0.0246	0.8723	0.8047	0.0676
	PN	0.6681	0.7210	0.7867	0.7328	0.7740	0.8125	0.7944	0.7556	0.0507	0.8125	0.6681	0.1444
	CD	0.3017	0.3248	0.3264	0.4328	0.3321	0.4462	0.4436	0.3725	0.0648	0.4462	0.3017	0.1445
	ALL	0.6858	0.7223	0.7360	0.7449	0.7308	0.7623	0.7604	0.7347	0.0261	0.7623	0.6858	0.0765
	Average	0.6275	0.6921	0.7045	0.7165	0.6971	0.7468	0.7385					
	STD	0.2128	0.1891	0.1938	0.1546	0.1888	0.1534	0.1519					
	MAX	0.8415	0.8801	0.8651	0.8701	0.8538	0.8812	0.8706					
MIN	0.3017	0.3248	0.3264	0.4328	0.3321	0.4462	0.4436						
TID	AGC	0.3526	0.2115	0.3792	0.3635	0.3900	0.4083	0.4304	0.3622	0.0715	0.4304	0.2115	0.2189
	AGN	0.5149	0.5300	0.5292	0.6693	0.5399	0.7145	0.7092	0.6010	0.0918	0.7145	0.5149	0.1996
	CA	0.6601	0.6451	0.6256	0.6444	0.6410	0.6455	0.6413	0.6433	0.0101	0.6601	0.6256	0.0344
	CC	0.0996	0.0992	0.0946	0.1100	0.1027	0.1015	0.1037	0.1016	0.0047	0.1100	0.0946	0.0154
	CCS	0.1344	0.1088	0.1300	0.1362	0.1342	0.1258	0.1346	0.1292	0.0096	0.1362	0.1088	0.0273
	CN	0.3846	0.2571	0.3696	0.3131	0.3881	0.3531	0.3604	0.3466	0.0467	0.3881	0.2571	0.1310
	GB	0.6762	0.7720	0.7431	0.8162	0.7257	0.8431	0.8204	0.7709	0.0599	0.8431	0.6762	0.1669
	HFN	0.7963	0.7862	0.8254	0.8123	0.8235	0.8462	0.8362	0.8180	0.0213	0.8462	0.7862	0.0600
	ICQ	0.7662	0.7808	0.8200	0.7831	0.8023	0.8165	0.8085	0.7968	0.0203	0.8200	0.7662	0.0538
	ID	0.6277	0.7192	0.7231	0.7877	0.7062	0.8088	0.7938	0.7381	0.0637	0.8088	0.6277	0.1812
	IN	0.7677	0.6023	0.7269	0.7158	0.7426	0.6992	0.7523	0.7153	0.0548	0.7677	0.6023	0.1654
	IS	0.1031	0.0950	0.1189	0.0823	0.1115	0.0831	0.0900	0.0977	0.0141	0.1189	0.0823	0.0365
	JPEG	0.6985	0.8077	0.7399	0.8360	0.7248	0.8472	0.8312	0.7836	0.0609	0.8472	0.6985	0.1487
	JPEGTE	0.5204	0.3885	0.4938	0.4512	0.5115	0.4354	0.4581	0.4655	0.0466	0.5204	0.3885	0.1319
	JPEG2k	0.7915	0.7692	0.7968	0.8120	0.8072	0.8082	0.8085	0.7991	0.0150	0.8120	0.7692	0.0427
	JPEG2kTE	0.4554	0.4419	0.5158	0.3823	0.5165	0.4931	0.5038	0.4727	0.0493	0.5165	0.3823	0.1342
	LBD	0.3362	0.3627	0.3560	0.3591	0.3548	0.3862	0.3635	0.3598	0.0148	0.3862	0.3362	0.0499
	LC	0.7212	0.3088	0.6808	0.4996	0.7169	0.5838	0.5815	0.5847	0.1464	0.7212	0.3088	0.4123
	MGN	0.6406	0.5823	0.6707	0.7346	0.6753	0.7705	0.7713	0.6922	0.0703	0.7713	0.5823	0.1890
	MN	0.3678	0.6290	0.4174	0.4681	0.4095	0.5466	0.5137	0.4789	0.0907	0.6290	0.3678	0.2612
	NEPN	0.1639	0.1329	0.1758	0.2146	0.1968	0.2073	0.2189	0.1872	0.0313	0.2189	0.1329	0.0860
	QN	0.8362	0.8146	0.8127	0.8469	0.8308	0.8323	0.8442	0.8311	0.0133	0.8469	0.8127	0.0342
	SCN	0.7492	0.7038	0.8265	0.7269	0.8131	0.7635	0.8027	0.7694	0.0462	0.8265	0.7038	0.1227
	SSR	0.7231	0.8846	0.7265	0.8815	0.7469	0.8835	0.8831	0.8185	0.0811	0.8846	0.7231	0.1615
	ALL	0.5910	0.5593	0.6031	0.6358	0.6075	0.6519	0.6545	0.6147	0.0347	0.6545	0.5593	0.0951
	Average	0.5391	0.5197	0.5561	0.5633	0.5608	0.5862	0.5886					
	STD	0.2361	0.2593	0.2408	0.2573	0.2388	0.2588	0.2563					
	MAX	0.8362	0.8846	0.8265	0.8815	0.8308	0.8835	0.8831					
	MIN	0.0996	0.0950	0.0946	0.0823	0.1027	0.0831	0.0900					

Table 5 shows the average SROCC of simulations using CLBP as the texture descriptor. For this descriptor, we tested the influence of each combination of feature set (see $CLBP_S$, $CLBP_M$, and $CLBP_C$ in Figure 9) on the image quality prediction. From Table 5, we can notice that the feature sets, $CLBP_M$ and $CLBP_C$, are individually unsatisfactory for measuring image quality. This is due to the low SROCC scores obtained for the three tested databases. On the other hand, $CLBP_S$ is the dominant feature set for quality description, since it presents the higher SROCC values in almost all cases.

Interestingly, the combination of CLBP feature sets produces a better performance, as indicated by the performances of $CLBP_{SM}$ ($CLBP_S + CLBP_M$) and $CLBP_{SMC}$ ($CLBP_S + CLBP_M + CLBP_C$). From Table 5, we can observe that the mean SROCC value of overall case increases from 0.91 ($CLBP_S$) to 0.93 ($CLBP_{MC}$ and $CLBP_{SMC}$) for the LIVE2 database. The combination of feature sets also improves the average SROCC values of the TID2013 database, increasing from 0.35 ($CLBP_S$) to 0.44 ($CLBP_{MC}$ and $CLBP_{SMC}$). The average values on CSIQ database show that the best performance is obtained using $CLBP_{MC}$. Based on these SROCC values, we can conclude that $CLBP_{MC}$ is the best combination of

features to assess image quality since the incorporation of $CLBP_C$ does not improve or even deteriorate the general prediction performance.

Table 5. Average SROCC of 1000 runs of simulations on tested databases using CLBP variations.

Radius		1				2				Average	STD	MAX	MIN	Δ	
Sampled Points	DIST	4	8	12	16	4	8	12	16						
LIVE2	DB														
	JPEG	0.9043	0.9074	0.9056	0.9049	0.8889	0.8826	0.9086	0.8617	0.8955	0.0166	0.9086	0.8617	0.0469	
	JPEG2k	0.9095	0.9107	0.9251	0.9009	0.9144	0.8980	0.9164	0.8888	0.9080	0.0115	0.9251	0.8888	0.0363	
	WN	0.9747	0.9714	0.9799	0.9730	0.9554	0.9585	0.9836	0.9515	0.9685	0.0119	0.9836	0.9515	0.0321	
	GB	0.9187	0.9343	0.9263	0.9383	0.9157	0.9168	0.9285	0.9151	0.9242	0.0090	0.9383	0.9151	0.0232	
	FF	0.8964	0.8520	0.8406	0.8577	0.8261	0.8003	0.8553	0.7634	0.8365	0.0404	0.8964	0.7634	0.1329	
	ALL	0.9264	0.9227	0.9252	0.9189	0.9053	0.8990	0.9242	0.8799	0.9127	0.0166	0.9264	0.8799	0.0465	
	Average	0.9217	0.9164	0.9171	0.9156	0.9010	0.8925	0.9194	0.8767						
	STD	0.0280	0.0391	0.0450	0.0387	0.0427	0.0522	0.0411	0.0637						
	MAX	0.9747	0.9714	0.9799	0.9730	0.9554	0.9585	0.9836	0.9515						
	MIN	0.8964	0.8520	0.8406	0.8577	0.8261	0.8003	0.8553	0.7634						
	TID	AGC	0.4642	0.2396	0.2892	0.3196	0.4247	0.1642	0.1648	0.1638	0.2788	0.1185	0.4642	0.1638	0.3004
		AGN	0.7827	0.6831	0.6758	0.7396	0.6565	0.5192	0.5546	0.5373	0.6436	0.0971	0.7827	0.5192	0.2635
		CA	0.5275	0.6265	0.5736	0.4519	0.6640	0.6035	0.4916	0.5340	0.5591	0.0711	0.6640	0.4519	0.2121
CC		0.1258	0.0989	0.0865	0.1192	0.1138	0.0912	0.1204	0.0954	0.1064	0.0151	0.1258	0.0865	0.0393	
CCS		0.1704	0.1496	0.1623	0.1415	0.1853	0.1541	0.1138	0.1317	0.1511	0.0225	0.1853	0.1138	0.0715	
CN		0.2373	0.2685	0.3096	0.4012	0.1658	0.4262	0.4035	0.3369	0.3186	0.0914	0.4262	0.1658	0.2604	
GB		0.8708	0.8915	0.8846	0.8631	0.8867	0.8777	0.8656	0.8722	0.8765	0.0103	0.8915	0.8631	0.0285	
HFN		0.8496	0.8263	0.8184	0.8232	0.8412	0.7654	0.7768	0.7220	0.8029	0.0439	0.8496	0.7220	0.1276	
ICQ		0.8205	0.8365	0.8277	0.8185	0.8208	0.8103	0.8476	0.8392	0.8277	0.0125	0.8476	0.8103	0.0372	
ID		0.5692	0.5992	0.6138	0.6000	0.7296	0.5815	0.6485	0.6731	0.6269	0.0537	0.7296	0.5692	0.1604	
IN		0.6150	0.6892	0.7328	0.7792	0.5831	0.5058	0.6253	0.6154	0.6432	0.0870	0.7792	0.5058	0.2735	
IS		0.1066	0.2123	0.1879	0.1246	0.1408	0.1471	0.1046	0.1144	0.1423	0.0393	0.2123	0.1046	0.1077	
JPEG		0.8101	0.8345	0.8088	0.8046	0.7746	0.8054	0.8281	0.8026	0.8086	0.0180	0.8345	0.7746	0.0599	
JPEGTE		0.2487	0.3846	0.3518	0.4125	0.2464	0.3762	0.4462	0.4015	0.3585	0.0738	0.4462	0.2464	0.1998	
JPEG2k		0.8200	0.8569	0.8515	0.8223	0.8786	0.8442	0.8615	0.8638	0.8499	0.0203	0.8786	0.8200	0.0586	
JPEG2kTE		0.5696	0.5538	0.5723	0.5812	0.7015	0.6977	0.6477	0.6623	0.6233	0.0608	0.7015	0.5538	0.1477	
LBD		0.1908	0.2723	0.1894	0.1787	0.1473	0.1875	0.2654	0.2881	0.2149	0.0522	0.2881	0.1473	0.1407	
LC		0.6585	0.5569	0.5415	0.5565	0.5846	0.5335	0.4827	0.4681	0.5478	0.0593	0.6585	0.4681	0.1904	
MGN		0.7275	0.6946	0.6758	0.7291	0.7091	0.5607	0.5761	0.5583	0.6539	0.0757	0.7291	0.5583	0.1707	
MN		0.4185	0.4234	0.3570	0.3937	0.1883	0.1598	0.1946	0.1908	0.2908	0.1170	0.4234	0.1598	0.2635	
NEPN		0.1452	0.2268	0.2801	0.3613	0.1486	0.2097	0.2360	0.2485	0.2320	0.0700	0.3613	0.1452	0.2160	
QN		0.7646	0.8108	0.8323	0.8154	0.7204	0.8103	0.7618	0.7800	0.7869	0.0370	0.8323	0.7204	0.1119	
SCN		0.7877	0.7927	0.7492	0.7646	0.7123	0.6408	0.6512	0.6577	0.7195	0.0629	0.7927	0.6408	0.1519	
SSR		0.8842	0.8608	0.8838	0.8519	0.8731	0.7892	0.7677	0.7962	0.8384	0.0467	0.8842	0.7677	0.1165	
_ALL	0.5925	0.6092	0.6070	0.5983	0.5747	0.5626	0.5904	0.5846	0.5899	0.0158	0.6092	0.5626	0.0466		
Average	0.5503	0.5599	0.5545	0.5621	0.5389	0.5129	0.5211	0.5175							
STD	0.2702	0.2598	0.2601	0.2541	0.2810	0.2602	0.2574	0.2578							
MAX	0.8842	0.8915	0.8846	0.8631	0.8867	0.8777	0.8656	0.8722							
MIN	0.1066	0.0989	0.0865	0.1192	0.1138	0.0912	0.1046	0.0954							

Table 6 depicts the mean SROCC values of simulations using other LBP variants. From this table, we can notice that almost all variants present an acceptable performance for the LIVE2 database. The exceptions are TPLBP and FPLBP that presented mean SROCC below 0.65, which is poorer than other methods. Based on the average values of mean SROCC on LIVE2, the methods LTP, RLBP, LCP, LVP, MLTP, SLBP, OCLBP, MLBP, MSLBP, and OCPP are in ascending order of performance. For the CSIQ and TID2013 databases, the methods perform similarly, but RLBP performs worse than LTP on CSIQ.

It is noticeable that multiscale approaches (MLBP, MLTP, and MSLBP) present the best results. For the three tested databases, the results are in agreement with the assumptions made by Freitas et al. [91], who demonstrated that combining multiple LBP descriptor parameters increases the prediction performance. However, we can observe that the OCPP descriptor presents the best performance when compared with any other tested descriptor, even when compared with the multiscale approaches. Although for the LIVE2 database the performance of the OCPP descriptor is similar to the performance of the MSLBP descriptor, this good performance is not achieved for the other databases. While MSLBP presents average SROCC values of 0.8147 for the CSIQ database, the OCPP presents an average SROCC value of 0.9140 for the same database. Similarly, for the TID2013 database, the average SROCC values obtained with MSLBP and OCPP are 0.5919 and 0.7035, respectively.

Table 6. Average SROCC of 100 runs of simulations on tested image databases using other LBP variations.

DB	DIST	LCP	LTP	RLBP	TPLBP	FPLBP	LVP	OCLBP	OCPP	SLBP	MLBP	MLTP	MSLBP
LIVE2	JPEG	0.8921	0.8278	0.8052	0.7047	0.6626	0.9363	0.9312	0.9678	0.9151	0.9249	0.9395	0.9373
	JPEG2k	0.8913	0.8029	0.8299	0.6491	0.5552	0.9461	0.9411	0.9597	0.9334	0.9342	0.9372	0.9406
	WN	0.9628	0.9358	0.9225	0.6354	0.6774	0.9764	0.9731	0.9861	0.9825	0.9822	0.9646	0.9831
	GB	0.9304	0.8824	0.9111	0.5923	0.5884	0.9531	0.9571	0.9612	0.9432	0.9524	0.9530	0.9619
	FF	0.8034	0.7004	0.7821	0.6724	0.6443	0.8848	0.8936	0.9141	0.9079	0.9487	0.8758	0.9364
	ALL	0.9006	0.8251	0.8487	0.6308	0.6171	0.9376	0.9418	0.9562	0.9405	0.9238	0.9316	0.9528
	Average	0.8968	0.8291	0.8499	0.6475	0.6242	0.9391	0.9397	0.9575	0.9371	0.9444	0.9336	0.9520
	STD	0.0534	0.0794	0.0566	0.0384	0.0465	0.0303	0.0269	0.0238	0.0263	0.0220	0.0307	0.0182
	MAX	0.9628	0.9358	0.9225	0.7047	0.6774	0.9764	0.9731	0.9861	0.9825	0.9822	0.9646	0.9831
	MIN	0.8034	0.7004	0.7821	0.5923	0.5552	0.8848	0.8936	0.9141	0.9079	0.9238	0.8758	0.9364
CSIQ	JPEG	0.8412	0.8011	0.7186	0.7524	0.7179	0.9221	0.8943	0.9596	0.8754	0.8847	0.9292	0.9064
	JPEG2k	0.7746	0.6371	0.6552	0.5699	0.6118	0.8946	0.8865	0.9331	0.7913	0.8095	0.8877	0.8156
	WN	0.8152	0.5057	0.6064	0.1931	0.3599	0.7063	0.8441	0.9186	0.8495	0.9014	0.6454	0.8939
	GB	0.7724	0.7901	0.7939	0.8517	0.6972	0.9137	0.9203	0.9390	0.8539	0.9159	0.9244	0.8816
	PN	0.7049	0.5356	0.2078	0.0815	0.3367	0.7091	0.8361	0.9471	0.7502	0.8872	0.7828	0.8431
	CD	0.1382	0.2246	0.1072	0.3174	0.1025	0.2659	0.4914	0.7753	0.4515	0.5172	0.2082	0.5299
	ALL	0.6672	0.5804	0.5109	0.4815	0.3214	0.8238	0.8421	0.9253	0.7971	0.8399	0.8280	0.8324
	Average	0.6734	0.5821	0.5143	0.4639	0.4496	0.7479	0.8164	0.9140	0.7670	0.8223	0.7437	0.8147
	STD	0.2435	0.1958	0.2607	0.2847	0.2300	0.2312	0.1468	0.0626	0.1457	0.1395	0.2559	0.1300
	MAX	0.8412	0.8011	0.7939	0.8517	0.7179	0.9221	0.9203	0.9596	0.8754	0.9159	0.9292	0.9064
MIN	0.1382	0.2246	0.1072	0.0815	0.1025	0.2659	0.4914	0.7753	0.4515	0.5172	0.2082	0.5299	
TID 2013	AGC	0.3683	0.3654	0.2273	0.1942	0.1207	0.4688	0.5315	0.8308	0.3999	0.5708	0.5963	0.6018
	AGN	0.3903	0.4211	0.5903	0.1731	0.2111	0.6069	0.7253	0.8634	0.6369	0.7884	0.6631	0.7811
	CA	0.2844	0.2267	0.3356	0.2884	0.1604	0.6944	0.4254	0.8821	0.2379	0.3144	0.6749	0.3891
	CC	0.1089	0.1857	0.0816	0.0953	0.1331	0.1756	0.0846	0.4785	0.1261	0.0881	0.1886	0.2161
	CCS	0.1251	0.1503	0.1934	0.2148	0.1296	0.1997	0.5704	0.5577	0.1402	0.1375	0.2384	0.2757
	CN	0.4769	0.2896	0.2682	0.1101	0.1942	0.2101	0.5849	0.5309	0.2725	0.3249	0.3880	0.5229
	GB	0.8455	0.5795	0.8084	0.8072	0.4096	0.8551	0.8607	0.8914	0.8215	0.8769	0.7465	0.8721
	HFN	0.6226	0.6678	0.7125	0.2735	0.3503	0.8181	0.8118	0.9445	0.7361	0.8676	0.7626	0.9031
	ICQ	0.7273	0.6334	0.4951	0.5592	0.5123	0.8261	0.7849	0.8350	0.8329	0.8134	0.7603	0.8302
	ID	0.5307	0.2249	0.4969	0.3623	0.2738	0.8694	0.7719	0.9102	0.5684	0.6434	0.7063	0.7488
	IN	0.4342	0.4257	0.4649	0.1107	0.1534	0.2866	0.5069	0.6696	0.1842	0.4551	0.6484	0.5838
	IS	0.0746	0.0821	0.1058	0.0757	0.0527	0.1406	0.1061	0.1699	0.0992	0.1165	0.3291	0.2092
	JPEG	0.6823	0.6914	0.6653	0.3506	0.5738	0.8961	0.8201	0.9158	0.7123	0.7964	0.6631	0.7907
	JPEGTE	0.4361	0.1138	0.2523	0.1024	0.0896	0.2925	0.5153	0.3795	0.2511	0.2131	0.2314	0.4353
	JPEG2k	0.8057	0.5692	0.7138	0.6557	0.3661	0.9099	0.8769	0.9407	0.8661	0.8507	0.7780	0.9369
	JPEG2kTE	0.6015	0.7531	0.3476	0.3769	0.1531	0.4394	0.5984	0.6552	0.5046	0.6711	0.6594	0.7388
	LBD	0.0969	0.1046	0.1453	0.1215	0.1135	0.1944	0.1311	0.1885	0.2374	0.1464	0.3813	0.2365
	LC	0.3242	0.1819	0.3226	0.2776	0.0876	0.5289	0.5692	0.8326	0.2565	0.3711	0.6533	0.3819
	MGN	0.4211	0.1281	0.5488	0.3085	0.1541	0.5324	0.6753	0.8471	0.6335	0.6666	0.6209	0.7512
	MN	0.1436	0.1988	0.1981	0.1546	0.2959	0.4168	0.5146	0.7290	0.3329	0.1535	0.4243	0.1638
	NEPN	0.1583	0.1009	0.1207	0.2603	0.0908	0.1534	0.2198	0.1545	0.3026	0.2558	0.1256	0.3712
	QN	0.7961	0.7711	0.6524	0.3618	0.5676	0.7869	0.8207	0.7890	0.8769	0.8623	0.7361	0.9173
	SCN	0.6546	0.6576	0.7911	0.1331	0.1126	0.6584	0.7192	0.8914	0.5803	0.7434	0.7015	0.6042
	SSR	0.7588	0.5781	0.6569	0.6623	0.5988	0.9088	0.8892	0.9391	0.6638	0.8488	0.8457	0.8357
ALL	0.4631	0.3437	0.4072	0.2512	0.1377	0.6997	0.6417	0.7621	0.5901	0.6339	0.6078	0.7012	
Average	0.4532	0.3778	0.4241	0.2912	0.2417	0.5428	0.5902	0.7035	0.4746	0.5284	0.5652	0.5919	
STD	0.2460	0.2353	0.2308	0.1958	0.1705	0.2767	0.2418	0.2524	0.2562	0.2873	0.2098	0.2530	
MAX	0.8455	0.7711	0.8084	0.8072	0.5988	0.9099	0.8892	0.9445	0.8769	0.8769	0.8457	0.9369	
MIN	0.0746	0.0821	0.0816	0.0757	0.0527	0.1406	0.0846	0.1545	0.0992	0.0881	0.1256	0.1638	

When we observe the results obtained per distortion for the CSIQ database, we can notice that the superiority of OCPP is due to the good performance obtained for the contrast distortions. While the quality prediction of contrast-distorted images has a mean SROCC value equal to 0.5299 when using MSLBP, the mean SROCC value for these same images are 0.7753 when using OCPP. Similarly, for the TID2013 database, the OCPP presents a superior performance for several types of distortions, especially for color and contrast-related distortions (AGC, AGN, CA, CC, CCS, etc.).

3.5. Comparison with Other IQA Methods

Figure 27 depicts the SROCC box plots for different no-reference IQA methods. Moreover, Table 7 depicts the results of six IQA methods, including two established full-reference metrics (PSNR and SSIM) and four state-of-the-art no-reference metrics (BRISQUE, CORNIA, CQA, and SSEQ). From this table, we can notice that CORNIA and SSEQ present the best performance on LIVE2

database, even when compared with full-reference approaches, such as PSNR and SSIM. On the LIVE2 database, the average SROCC values of CORNIA and SSEQ is 0.92, a score similar to some LBP-based descriptors, such as CLBP_{SM} and BSIF. However, several LBP-based descriptors (LPQ, MLBP, MSLBP, and OCPP) present a notable performance, being superior to the state-of-the-art methods, achieving average SROCC values above 0.94 for the LIVE2 database.

Table 7. Average SROCC of 100 runs of simulations on tested image databases using state-of-the-art IQA methods.

DB	DISTORTION	PSNR	SSIM	BRISQUE	CORNIA	CQA	SSEQ
LIVE 2	JPEG	0.8515	0.9481	0.8641	0.9002	0.8257	0.9122
	JPEG2k	0.8822	0.9438	0.8838	0.9246	0.8366	0.9388
	WN	0.9851	0.9793	0.9750	0.9500	0.9764	0.9544
	GB	0.7818	0.8889	0.9304	0.9465	0.8377	0.9157
	FF	0.8869	0.9335	0.8469	0.9132	0.8262	0.9038
	ALL	0.8013	0.8902	0.9098	0.9386	0.8606	0.9356
	Average	0.8648	0.9306	0.9017	0.9289	0.8605	0.9268
	STD	0.0726	0.0353	0.0469	0.0197	0.0582	0.0192
	MAX	0.9851	0.9793	0.9750	0.9500	0.9764	0.9544
	MIN	0.7818	0.8889	0.8469	0.9002	0.8257	0.9038
CSIQ	JPEG	0.9009	0.9309	0.8525	0.8319	0.6506	0.8066
	JPEG2k	0.9309	0.9251	0.8458	0.8405	0.8214	0.7302
	WN	0.9345	0.8761	0.6931	0.6187	0.7276	0.7876
	GB	0.9358	0.9089	0.8337	0.8526	0.7486	0.7766
	PN	0.9315	0.8871	0.7740	0.5340	0.5463	0.6661
	CD	0.8862	0.8128	0.4255	0.4458	0.5383	0.4172
	ALL	0.8088	0.8116	0.7597	0.6969	0.6369	0.7007
	Average	0.9041	0.8789	0.7406	0.6886	0.6671	0.6979
	STD	0.0462	0.0495	0.1502	0.1624	0.1053	0.1335
	MAX	0.9358	0.9309	0.8525	0.8526	0.8214	0.8066
MIN	0.8088	0.8116	0.4255	0.4458	0.5383	0.4172	
TID 2013	AGC	0.8568	0.7912	0.4166	0.2605	0.3964	0.3949
	AGN	0.9337	0.6421	0.6416	0.5689	0.6051	0.6040
	CA	0.7759	0.7158	0.7310	0.6844	0.4380	0.4366
	CC	0.4608	0.3477	0.1849	0.1400	0.2043	0.2006
	CCS	0.6892	0.7641	0.2715	0.2642	0.2461	0.2547
	CN	0.8838	0.6465	0.2176	0.3553	0.1623	0.1642
	GB	0.8905	0.8196	0.8063	0.8341	0.7019	0.7058
	HFN	0.9165	0.7962	0.7103	0.7707	0.7104	0.7061
	ICQ	0.9087	0.7271	0.7663	0.7044	0.6829	0.6834
	ID	0.9457	0.8327	0.5243	0.7227	0.6711	0.6716
	IN	0.9263	0.8055	0.6848	0.5874	0.4231	0.4272
	IS	0.7647	0.7411	0.2224	0.2403	0.2011	0.2013
	JPEG	0.9252	0.8275	0.7252	0.7815	0.6317	0.6284
	JPEGTE	0.7874	0.6144	0.3581	0.5679	0.2221	0.2195
	JPEG2k	0.8934	0.7531	0.7337	0.8089	0.7219	0.7205
	JPEG2kTE	0.8581	0.7067	0.7277	0.6113	0.6529	0.6529
	LBD	0.1301	0.6213	0.2833	0.2157	0.2382	0.2290
	LC	0.9386	0.8311	0.5726	0.6682	0.4561	0.4460
	MGN	0.9085	0.7863	0.5548	0.4393	0.4969	0.4897
	MN	0.8385	0.7388	0.2650	0.2342	0.2506	0.2575
	NEPN	0.6931	0.5326	0.1821	0.2855	0.1308	0.1275
	QN	0.8636	0.7428	0.5383	0.4922	0.7242	0.7214
	SCN	0.9152	0.7934	0.7238	0.7043	0.7121	0.7064
	SSR	0.9241	0.7774	0.7101	0.8594	0.8115	0.8084
ALL	0.6869	0.5758	0.5416	0.6006	0.4925	0.4900	
Average	0.8126	0.7172	0.5238	0.5361	0.4794	0.4779	
STD	0.1814	0.1135	0.2145	0.2258	0.2191	0.2186	
MAX	0.9457	0.8327	0.8063	0.8594	0.8115	0.8084	
MIN	0.1301	0.3477	0.1821	0.1400	0.1308	0.1275	

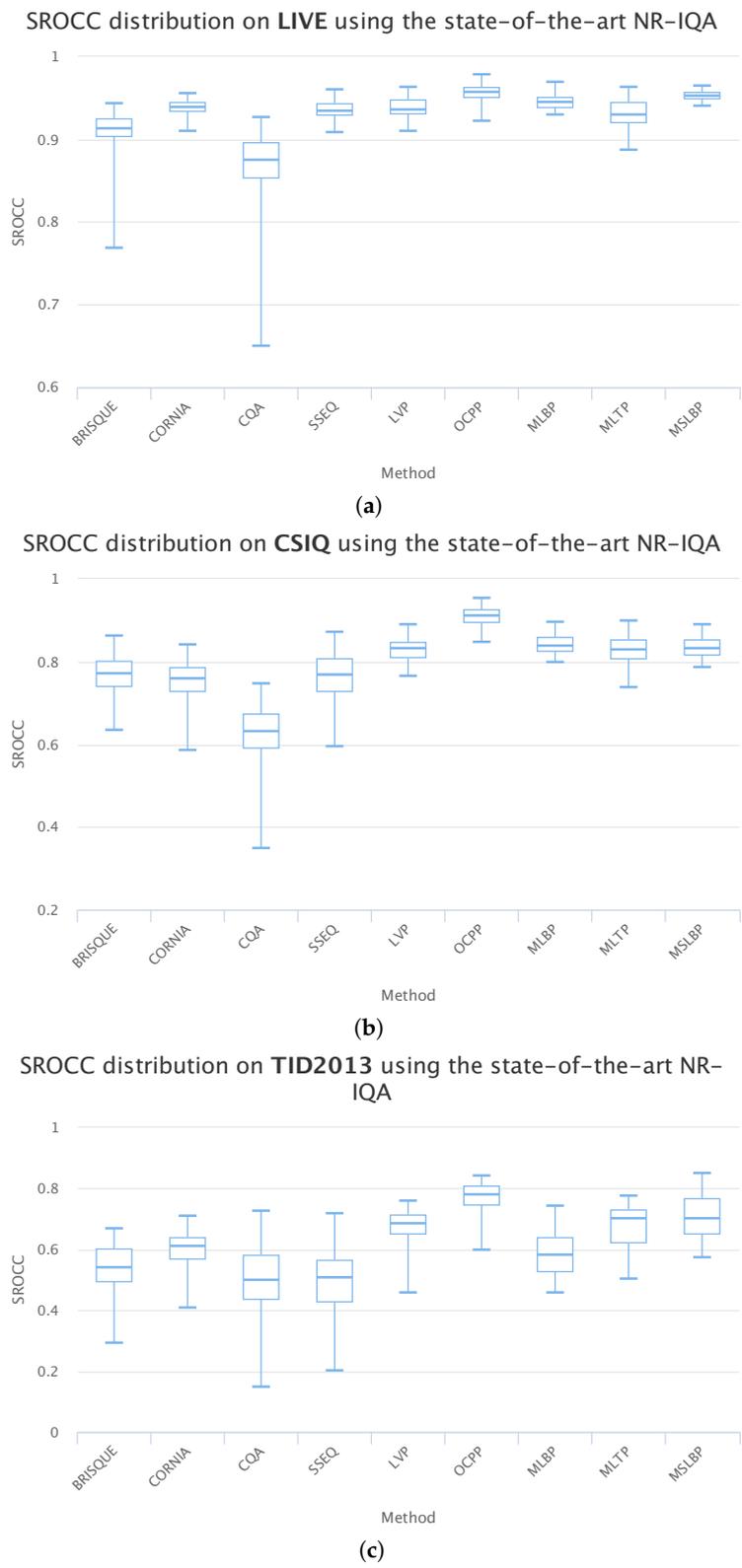


Figure 27. Distribution of average SROCC after 100 simulations using different state-of-the-art methods. (a) LIVE2. (b) CSIQ. (c) TID2013.

By comparing Table 7 with Tables 4–6, we can notice that LBP-based NR-IQA approaches present better performance also for the CSIQ and TID2013 databases. For the CSIQ database, we can observe that, on average, the best state-of-the-art NR-IQA method is BRISQUE, followed by SSEQ and CORNIA. The average SROCC scores are 0.7406, 0.6979, and 0.6886 for BRISQUE, SSEQ, and CORNIA, respectively. However, the LPQ, BSIF, LVP, OCLBP, OCPP, SLBP, MLBP, MLTP, and MSLBP descriptors present better results for this CSIQ database. Similarly, for the TID2013 database, the best state-of-the-art method is CORNIA, which presents an average SROCC of 0.5361. This value is outperformed by several LBP-based descriptors, such as LVP (0.5428), OCLBP (0.5902), OCPP (0.7035), MLBP (0.5284), MLTP (0.5652), MSLBP (0.5919), and LPQ (0.5518).

3.6. Prediction Performance on Cross-Database Validation

To investigate the generalization capability of the studied methods, we performed a cross-database validation. This validation consists of training the ML algorithm using all images of one database and testing the them on the other databases. Table 8 depicts the SROCC values obtained using LIVE2 as the training database and TID2013 and CSIQ as the testing databases. To perform a straightforward cross-database comparison, only the shared subset of distortions are selected from each database.

Table 8. SROCC cross-database validation, when models are trained on LIVE2 and tested on CSIQ and TID2013.

Database	Distortion	BRISQUE	CORNIA	CQA	SSEQ	LVP	OCPP	MLBP	MLTP	MSLBP
TID2013	JPEG	0.8058	0.7423	0.8071	0.7823	0.7827	0.8875	0.8378	0.8472	0.8779
	JPEG2k	0.8224	0.8837	0.7724	0.8258	0.8718	0.9246	0.9219	0.9046	0.9293
	WN	0.8621	0.7403	0.8692	0.6959	0.7781	0.9001	0.8351	0.6881	0.8766
	GB	0.8245	0.8133	0.8214	0.8624	0.8873	0.8651	0.8849	0.8693	0.8958
	ALL	0.7965	0.7599	0.8214	0.7955	0.8365	0.8814	0.8661	0.8137	0.8776
CSIQ	JPEG	0.8209	0.7062	0.7129	0.8141	0.8334	0.9091	0.9012	0.8784	0.9151
	JPEG2k	0.8279	0.8459	0.6957	0.7862	0.7716	0.9101	0.8744	0.8914	0.8846
	WN	0.6951	0.8627	0.6596	0.4613	0.8229	0.9107	0.8498	0.7739	0.8809
	GB	0.8311	0.8815	0.7648	0.7758	0.8753	0.9188	0.9047	0.8712	0.9115
	ALL	0.8022	0.7542	0.7114	0.7403	0.8359	0.8921	0.8608	0.8628	0.8723

Based on the results in Table 8, we can notice that OCPP outperforms other methods for almost all types of distortions. For TID2013, the OCPP outperforms the other methods for 3 out of the 5 distortions, while for CSIQ it outperforms the other methods for 4 distortions out of the 5 distortions. OCPP is followed by MSLBP, which achieves the best results in the cases where OCPP is not the best. The cross-database validation test indicates that, in general, texture descriptors have a better generalization capacity, when compared to the tested state-of-the-art methods.

3.7. Simulation Statistics

In order to investigate the stability of the mean over the simulations, we generated some box plots depicted in Figure 28. We chosen the BSIF, LCP, CLBP, and LPQ because these descriptors were among the best analyzed in the last section. Based on this figure, we can notice that the mean changes over the simulations. More specifically, the inter-quartile ranges increase over the simulations for BSIF and LPQ on LIVE2 database. On the other hand, this behavior is not the same for LCP and CLBP descriptors. The pattern on CSIQ and TID2013 are more similar. Further studies concerning the number of simulations that generates a stable distribution are suggested as future works.

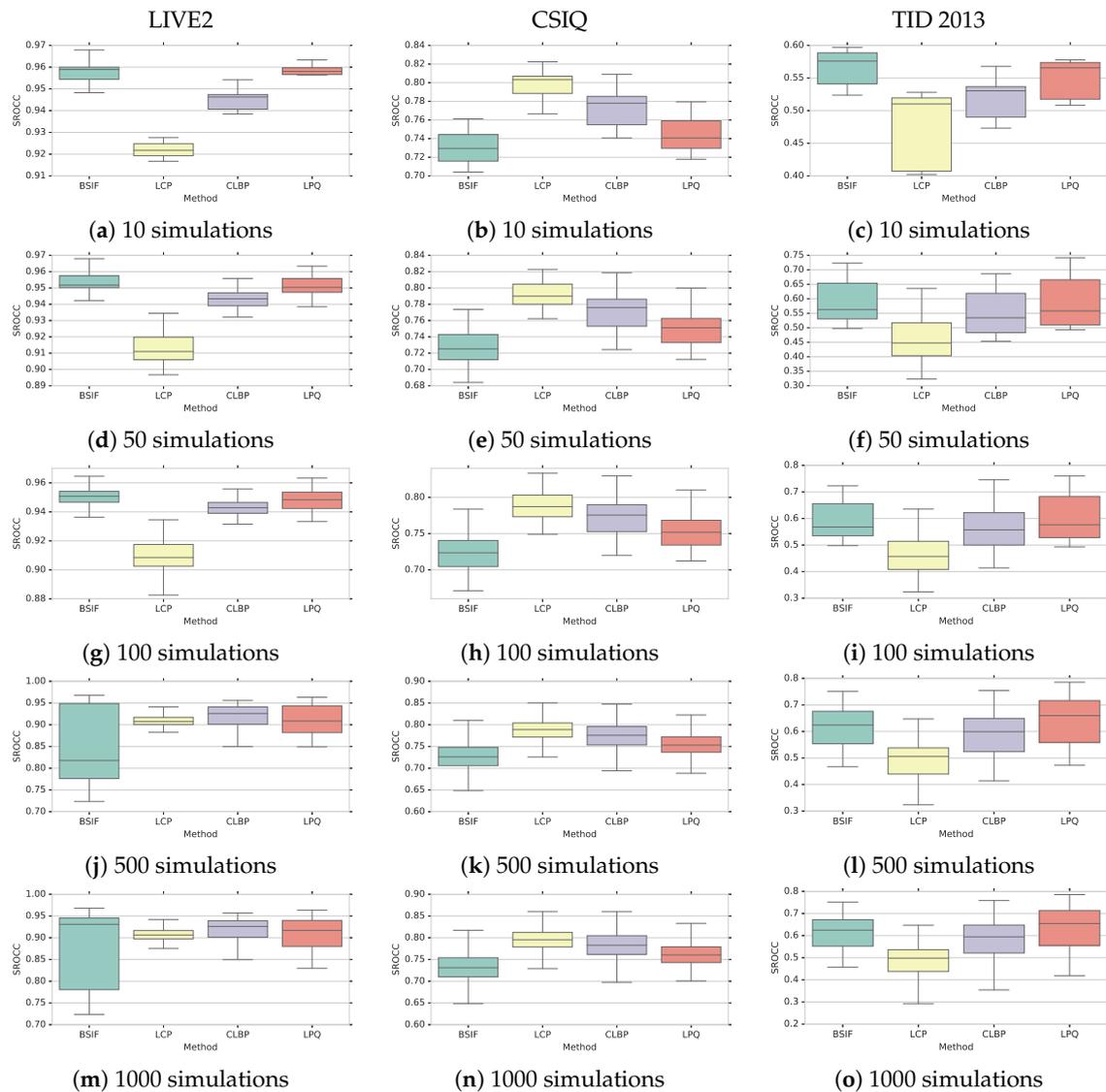


Figure 28. Stability of the mean SROCC over the simulations after 10, 50, 100, 500, and 1000 simulations using different descriptors.

4. Conclusions

In this paper, we compared three basic LBP (LBP^{ri} , LBP^u , and LBP^{riu2}) with eight different parameter combinations each. This comparison was performed to verify whether LBP can be used as a feature descriptor in image quality assessment applications. Preliminary results show that, although LBP can be used in image quality assessment, the performance varies greatly for each distortion and the its parameters. Based on these results, we investigated other 14 texture descriptors, which are variants of the basic LBP. When tested using the proposed framework, BSIF, LPQ, LVP, and CLBP present a good mean correlation value for the LIVE2 database, but their performances decrease for the CSIQ and TID2013 datasets due to color and contrast distortions. Results show that multiscale approaches have a substantially better quality prediction performance. Among the tested multiscale approaches, the MSLBP descriptor, which incorporates visual saliency, has the best performance. While MSLBP has a performance that is similar to the performance obtained with the OCPP descriptor for the LIVE2 database, the OCPP presents the best performance for the remaining databases.

Author Contributions: P.G.F. wrote most of the text, figures, and analysis and developed the methods described in this manuscript. L.P.d.E. and S.S.S. were responsible to perform the experiments. M.C.Q.d.F. is the principal investigator in this project and has guided the research and helped write and revise this manuscript.

Funding: No funding information.

Acknowledgments: This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and by the University of Brasília (UnB).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Chen, Q.H.; Xie, X.F.; Cao, J.; Cui, X.C. Research of ROI image compression based on visual attention model. In Proceedings of the International Conference on Image Processing and Pattern Recognition in Industrial Engineering, Xi'an, China, 7–8 August 2010; Volume 7820, p. 78202W.
- Wang, Z.; Li, Q.; Shang, X. Perceptual image coding based on a maximum of minimal structural similarity criterion. In Proceedings of the IEEE International Conference on Image Processing, ICIP 2007, San Antonio, TX, USA, 16–19 September 2007; Volume 2, p. II-121.
- Chen, Z.; Guillemot, C. Perceptually-friendly H. 264/AVC video coding based on foveated just-noticeable-distortion model. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 806–819. [[CrossRef](#)]
- Ou, T.S.; Huang, Y.H.; Chen, H.H. SSIM-based perceptual rate control for video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 682–691.
- Wang, Z.; Baroud, Y.; Najmabadi, S.M.; Simon, S. Low complexity perceptual image coding by just-noticeable difference model based adaptive downsampling. In Proceedings of the Picture Coding Symposium (PCS), Nuremberg, Germany, 4–7 December 2016; pp. 1–5.
- Wu, H.R.; Rao, K.R. *Digital Video Image Quality and Perceptual Coding*; CRC Press: Boca Raton, FL, USA, 2017.
- Zhang, F.; Liu, W.; Lin, W.; Ngan, K.N. Spread spectrum image watermarking based on perceptual quality metric. *IEEE Trans. Image Process.* **2011**, *20*, 3207–3218. [[CrossRef](#)] [[PubMed](#)]
- Urvoy, M.; Goudia, D.; Autrusseau, F. Perceptual DFT watermarking with improved detection and robustness to geometrical distortions. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 1108–1119. [[CrossRef](#)]
- Conviva. Viewer Experience Report. 2015. Available online: <http://www.conviva.com/convivaviewer-experience-report/vxr-2015/> (accessed on 19 July 2017).
- Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; Matas, J. DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. *arXiv* **2017**, arXiv:1711.07064.
- Dodge, S.; Karam, L. Understanding how image quality affects deep neural networks. In Proceedings of the 2016 IEEE Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, 6–8 June 2016; pp. 1–6.
- Bhogal, A.P.S.; Söllinger, D.; Trung, P.; Hämmerle-Uhl, J.; Uhl, A. Non-reference Image Quality Assessment for Fingervein Presentation Attack Detection. In Proceedings of the Scandinavian Conference on Image Analysis, Tromsø, Norway, 12–14 June 2017; pp. 184–196.
- Söllinger, D.; Trung, P.; Uhl, A. Non-reference image quality assessment and natural scene statistics to counter biometric sensor spoofing. *IET Biom.* **2018**, *7*, 314–324. [[CrossRef](#)]
- Karahan, S.; Yildirum, M.K.; Kirtac, K.; Rende, F.S.; Butun, G.; Ekenel, H.K. How image degradations affect deep CNN-based face recognition? In Proceedings of the 2016 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 21–23 September 2016; pp. 1–5.
- Chernov, T.S.; Razumnuy, N.P.; Kozharinov, A.S.; Nikolaev, D.P.; Arlazarov, V.V. Image quality assessment for video stream recognition systems. In Proceedings of the Tenth International Conference on Machine Vision (ICMV 2017), Vienna, Austria, 13–15 November 2017; Volume 10696, p. 106961U.
- Jeelani, H.; Martin, J.; Vasquez, F.; Salerno, M.; Weller, D.S. Image quality affects deep learning reconstruction of MRI. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 357–360.

17. Choi, J.H.; Cheon, M.; Lee, J.S. Influence of Video Quality on Multi-view Activity Recognition. In Proceedings of the 2017 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 11–13 December 2017; pp. 511–515.
18. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
19. Nah, S.; Kim, T.H.; Lee, K.M. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 257–265.
20. Seshadrinathan, K.; Bovik, A.C. Automatic prediction of perceptual quality of multimedia signals—A survey. *Multimed. Tools Appl.* **2011**, *51*, 163–186. [[CrossRef](#)]
21. Ferzli, R.; Karam, L.J. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE Trans. Image Process.* **2009**, *18*, 717–728. [[CrossRef](#)] [[PubMed](#)]
22. Maheshwary, P.; Shirvaikar, M.; Grecos, C. Blind image sharpness metric based on edge and texture features. In Proceedings of the Real-Time Image and Video Processing 2018, Orlando, FL, USA, 15–19 April 2018; Volume 10670, p. 1067004.
23. Li, L.; Xia, W.; Lin, W.; Fang, Y.; Wang, S. No-reference and robust image sharpness evaluation based on multiscale spatial and spectral features. *IEEE Trans. Multimed.* **2017**, *19*, 1030–1040. [[CrossRef](#)]
24. Ong, E.; Lin, W.; Lu, Z.; Yao, S.; Yang, X.; Jiang, L. No-reference JPEG-2000 image quality metric. In Proceedings of the 2003 International Conference on Multimedia and Expo, ICME'03, Baltimore, MD, USA, 6–9 July 2003; Volume 1, pp. 1–545.
25. Barland, R.; Saadane, A. Reference free quality metric for JPEG-2000 compressed images. In Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, Sydney, Australia, 28–31 August 2005; Volume 1, pp. 351–354.
26. Li, L.; Zhou, Y.; Lin, W.; Wu, J.; Zhang, X.; Chen, B. No-reference quality assessment of deblocked images. *Neurocomputing* **2016**, *177*, 572–584. [[CrossRef](#)]
27. Gu, K.; Lin, W.; Zhai, G.; Yang, X.; Zhang, W.; Chen, C.W. No-Reference Quality Metric of Contrast-Distorted Images Based on Information Maximization. *IEEE Trans. Cybern.* **2017**, *47*, 4559–4565. [[CrossRef](#)] [[PubMed](#)]
28. Gu, K.; Zhai, G.; Yang, X.; Zhang, W.; Chen, C.W. Automatic contrast enhancement technology with saliency preservation. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1480–1494.
29. Chandler, D.M. Seven challenges in image quality assessment: Past, present, and future research. *ISRN Signal Process.* **2013**, *2013*, 905685. [[CrossRef](#)]
30. Chandler, D.M.; Alam, M.M.; Phan, T.D. Seven challenges for image quality research. In Proceedings of the IS&T/SPIE Electronic Imaging, San Francisco, CA, USA, 2–6 February 2014; p. 901402.
31. Hemami, S.S.; Reibman, A.R. No-reference image and video quality estimation: Applications and human-motivated design. *Signal Process. Image Commun.* **2010**, *25*, 469–481. [[CrossRef](#)]
32. Cheng, G.; Huang, J.; Liu, Z.; Lizhi, C. Image quality assessment using natural image statistics in gradient domain. *AEU Int. J. Electron. Commun.* **2011**, *65*, 392–397. [[CrossRef](#)]
33. Appina, B.; Khan, S.; Channappayya, S.S. No-reference Stereoscopic Image Quality Assessment Using Natural Scene Statistics. *Signal Process. Image Commun.* **2016**, *43*, 1–14. [[CrossRef](#)]
34. Zhang, Y.; Wu, J.; Xie, X.; Li, L.; Shi, G. Blind image quality assessment with improved natural scene statistics model. *Digit. Signal Process.* **2016**, *57*, 56–65. [[CrossRef](#)]
35. Fang, Y.; Ma, K.; Wang, Z.; Lin, W.; Fang, Z.; Zhai, G. No-reference quality assessment of contrast-distorted images based on natural scene statistics. *IEEE Signal Process. Lett.* **2015**, *22*, 838–842. [[CrossRef](#)]
36. Saad, M.; Bovik, A.C.; Charrier, C. A DCT statistics-based blind image quality index. *IEEE Signal Process. Lett.* **2010**, *17*, 583–586. [[CrossRef](#)]
37. Ma, L.; Li, S.; Ngan, K.N. Reduced-reference image quality assessment in reorganized DCT domain. *Signal Process. Image Commun.* **2013**, *28*, 884–902. [[CrossRef](#)]
38. Saad, M.A.; Bovik, A.C.; Charrier, C. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Trans. Image Process.* **2012**, *21*, 3339–3352. [[CrossRef](#)] [[PubMed](#)]
39. Moorthy, A.K.; Bovik, A.C. A two-step framework for constructing blind image quality indices. *IEEE Signal Process. Lett.* **2010**, *17*, 513–516. [[CrossRef](#)]

40. Moorthy, A.K.; Bovik, A.C. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Trans. Image Process.* **2011**, *20*, 3350–3364. [[CrossRef](#)] [[PubMed](#)]
41. He, L.; Tao, D.; Li, X.; Gao, X. Sparse representation for blind image quality assessment. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 1146–1153.
42. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [[CrossRef](#)] [[PubMed](#)]
43. Kang, L.; Ye, P.; Li, Y.; Doermann, D. Convolutional neural networks for no-reference image quality assessment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1733–1740.
44. Li, J.; Zou, L.; Yan, J.; Deng, D.; Qu, T.; Xie, G. No-reference image quality assessment using Prewitt magnitude based on convolutional neural networks. *Signal Image Video Process.* **2016**, *10*, 609–616. [[CrossRef](#)]
45. Bosse, S.; Maniry, D.; Wiegand, T.; Samek, W. A deep neural network for image quality assessment. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3773–3777.
46. Kuzovkin, I.; Vicente, R.; Petton, M.; Lachaux, J.P.; Baciú, M.; Kahane, P.; Rheims, S.; Vidal, J.R.; Aru, J. Frequency-Resolved Correlates of Visual Object Recognition in Human Brain Revealed by Deep Convolutional Neural Networks. *bioRxiv* **2017**, 133694. [[CrossRef](#)]
47. Yamins, D.L.; DiCarlo, J.J. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* **2016**, *19*, 356–365. [[CrossRef](#)] [[PubMed](#)]
48. Bianco, S.; Celona, L.; Napoletano, P.; Schettini, R. On the use of deep learning for blind image quality assessment. *Signal Image Video Process.* **2018**, *12*, 355–362. [[CrossRef](#)]
49. Scott, E.T.; Hemami, S.S. No-Reference Utility Estimation with a Convolutional Neural Network. *Electron. Imaging* **2018**, *2018*, 1–6. [[CrossRef](#)]
50. Jia, S.; Zhang, Y. Saliency-based deep convolutional neural network for no-reference image quality assessment. *Multimed. Tools Appl.* **2018**, *77*, 14859–14872. [[CrossRef](#)]
51. Zhang, L.; Shen, Y.; Li, H. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.* **2014**, *23*, 4270–4281. [[CrossRef](#)] [[PubMed](#)]
52. Farias, M.C.; Akamine, W.Y. On performance of image quality metrics enhanced with visual attention computational models. *Electron. Lett.* **2012**, *48*, 631–633. [[CrossRef](#)]
53. Engelke, U.; Kaprykowsky, H.; Zepernick, H.J.; Ndjiki-Nya, P. Visual attention in quality assessment. *IEEE Signal Process. Mag.* **2011**, *28*, 50–59. [[CrossRef](#)]
54. Gu, K.; Wang, S.; Yang, H.; Lin, W.; Zhai, G.; Yang, X.; Zhang, W. Saliency-guided quality assessment of screen content images. *IEEE Trans. Multimed.* **2016**, *18*, 1098–1110. [[CrossRef](#)]
55. You, J.; Perkis, A.; Hannuksela, M.M.; Gabbouj, M. Perceptual quality assessment based on visual attention analysis. In Proceedings of the 17th ACM international conference on Multimedia, Beijing, China, 19–24 October 2009; pp. 561–564.
56. Le Meur, O.; Ninassi, A.; Le Callet, P.; Barba, D. Overt visual attention for free-viewing and quality assessment tasks: Impact of the regions of interest on a video quality metric. *Signal Process. Image Commun.* **2010**, *25*, 547–558. [[CrossRef](#)]
57. Le Meur, O.; Ninassi, A.; Le Callet, P.; Barba, D. Do video coding impairments disturb the visual attention deployment? *Signal Process. Image Commun.* **2010**, *25*, 597–609. [[CrossRef](#)]
58. Akamine, W.Y.; Farias, M.C. Video quality assessment using visual attention computational models. *J. Electron. Imaging* **2014**, *23*, 061107. [[CrossRef](#)]
59. Ciocca, G.; Corchs, S.; Gasparini, F. A complexity-based image analysis to investigate interference between distortions and image contents in image quality assessment. In Proceedings of the International Workshop on Computational Color Imaging, Milan, Italy, 29–31 March 2017; pp. 105–121.
60. Larson, E.C.; Chandler, D.M. Most apparent distortion: full-reference image quality assessment and the role of strategy. *J. Electron. Imaging* **2010**, *19*, 011006.
61. Liu, M.; Wang, M.; Wang, J.; Li, D. Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. *Sens. Actuators B Chem.* **2013**, *177*, 970–980. [[CrossRef](#)]
62. Petrou, M.; Sevilla, P.G. *Image Processing: Dealing with Texture*; John Wiley and Sons: Hoboken, NJ, USA, 2006.

63. Davies, E.R. Introduction to Texture Analysis. In *Handbook of Texture Analysis*; Mirmehdi, M., Xie, X., Suri, J., Eds.; Imperial College Press: London, UK, 2008; Chapter 1, pp. 1–31.
64. Galloway, M.M. Texture analysis using gray level run lengths. *Comput. Graph. Image Process.* **1975**, *4*, 172–179. [[CrossRef](#)]
65. Soh, L.K.; Tsatsoulis, C. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 780–795. [[CrossRef](#)]
66. He, D.C.; Wang, L. Texture unit, texture spectrum, and texture analysis. *IEEE Trans. Geosci. Remote Sens.* **1990**, *28*, 509–512.
67. Julesz, B. Textons, the elements of texture perception, and their interactions. *Nature* **1981**, *290*, 91–97. [[CrossRef](#)] [[PubMed](#)]
68. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [[CrossRef](#)]
69. Hadid, A.; Ylioinas, J.; Bengherabi, M.; Ghahramani, M.; Taleb-Ahmed, A. Gender and texture classification: A comparative analysis using 13 variants of local binary patterns. *Pattern Recognit. Lett.* **2015**, *68*, 231–238. [[CrossRef](#)]
70. Brahmam, S.; Jain, L.C.; Lumini, A.; Nanni, L. Introduction to Local Binary Patterns: New Variants and Applications. In *Local Binary Patterns; Studies in Computational Intelligence*; Brahmam, S., Jain, L.C., Nanni, L., Lumini, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 506, pp. 1–13.
71. Pietikäinen, M.; Ojala, T.; Xu, Z. Rotation-invariant texture classification using feature distributions. *Pattern Recognit.* **2000**, *33*, 43–52. [[CrossRef](#)]
72. Ojansivu, V.; Heikkilä, J. Blur insensitive texture classification using local phase quantization. In Proceedings of the International Conference on Image and Signal Processing, Cherbourg-Octeville, France, 1–3 July 2008; pp. 236–243.
73. Kannala, J.; Rahtu, E. Bsf: Binarized statistical image features. In Proceedings of the 2012 21st International Conference on Pattern Recognition (ICPR), Tsukuba, Japan, 11–15 November 2012; pp. 1363–1366.
74. Arashloo, S.R.; Kittler, J. Dynamic texture recognition using multiscale binarized statistical image features. *IEEE Trans. Multimed.* **2014**, *16*, 2099–2109. [[CrossRef](#)]
75. Raja, K.B.; Raghavendra, R.; Busch, C. Binarized statistical features for improved iris and periocular recognition in visible spectrum. In Proceedings of the 2014 International Workshop on Biometrics and Forensics (IWBF), Valletta, Malta, 27–28 March 2014; pp. 1–6.
76. Arashloo, S.R.; Kittler, J.; Christmas, W. Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2396–2407. [[CrossRef](#)]
77. Raghavendra, R.; Busch, C. Robust scheme for iris presentation attack detection using multiscale binarized statistical image features. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 703–715. [[CrossRef](#)]
78. Mehta, R.; Egiazarian, K.O. *Rotated Local Binary Pattern (RLBP)-Rotation Invariant Texture Descriptor*; ICPRAM: Barcelona, Spain, 2013; pp. 497–502.
79. Mehta, R.; Egiazarian, K. Dominant rotated local binary patterns (DRLBP) for texture classification. *Pattern Recognit. Lett.* **2016**, *71*, 16–22. [[CrossRef](#)]
80. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [[CrossRef](#)]
81. Briët, J.; Harremoës, P. Properties of classical and quantum Jensen-Shannon divergence. *Phys. Rev. A* **2009**, *79*, 052311. [[CrossRef](#)]
82. Ye, N.; Borrer, C.M.; Parmar, D. Scalable Chi-Square Distance versus Conventional Statistical Distance for Process Monitoring with Uncorrelated Data Variables. *Qual. Reliab. Eng. Int.* **2003**, *19*, 505–515. [[CrossRef](#)]
83. Guo, Z.; Zhang, L.; Zhang, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.* **2010**, *19*, 1657–1663. [[PubMed](#)]
84. Guo, Y.; Zhao, G.; Pietikäinen, M. Texture Classification using a Linear Configuration Model based Descriptor. In *Proceedings of the British Machine Vision Conference*; BMVA Press: Dundee, UK, 2011; pp. 119.1–119.10.
85. Mäenpää, T. *The Local Binary Pattern Approach to Texture Analysis: Extensions and Applications*; Oulun Yliopisto: Oulu, Finland, 2003.
86. Jain, A.; Healey, G. A multiscale representation including opponent color features for texture recognition. *IEEE Trans. Image Process.* **1998**, *7*, 124–128. [[CrossRef](#)] [[PubMed](#)]

87. Wolf, L.; Hassner, T.; Taigman, Y. Descriptor Based Methods in the Wild. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*; Erik Learned-Miller and Andras Ferencz and Frédéric Jurie: Marseille, France, 2008.
88. Chang, D.J.; Desoky, A.H.; Ouyang, M.; Rouchka, E.C. Compute pairwise manhattan distance and pearson correlation coefficient of data points with gpu. In *Proceedings of the 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, Daegu, Korea, 27–29 May 2009; pp. 501–506.
89. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D.L. The mahalanobis distance. *Chemom. Intell. Lab. Syst.* **2000**, *50*, 1–18. [[CrossRef](#)]
90. Merigó, J.M.; Casanovas, M. A new Minkowski distance based on induced aggregation operators. *Int. J. Comput. Intell. Syst.* **2011**, *4*, 123–133. [[CrossRef](#)]
91. Freitas, P.G.; Akamine, W.Y.; Farias, M.C. Blind Image Quality Assessment Using Multiscale Local Binary Patterns. *J. Imaging Sci. Technol.* **2016**, *60*, 60405-1. [[CrossRef](#)]
92. Anthimopoulos, M.; Gatos, B.; Pratikakis, I. Detection of artificial and scene text in images and video frames. *Pattern Anal. Appl.* **2013**, *16*, 431–446. [[CrossRef](#)]
93. Freitas, P.G.; Akamine, W.Y.; Farias, M.C. No-reference image quality assessment based on statistics of Local Ternary Pattern. In *Proceedings of the 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, 6–8 June 2016; pp. 1–6.
94. Zhang, J.; Sclaroff, S. Exploiting surroundedness for saliency detection: A Boolean map approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 889–902. [[CrossRef](#)] [[PubMed](#)]
95. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
96. Ye, P.; Kumar, J.; Kang, L.; Doermann, D. Unsupervised feature learning framework for no-reference image quality assessment. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012; pp. 1098–1105.
97. Liu, L.; Dong, H.; Huang, H.; Bovik, A.C. No-reference image quality assessment in curvelet domain. *Signal Process. Image Commun.* **2014**, *29*, 494–505. [[CrossRef](#)]
98. Liu, L.; Liu, B.; Huang, H.; Bovik, A.C. No-reference image quality assessment based on spatial and spectral entropies. *Signal Process. Image Commun.* **2014**, *29*, 856–863. [[CrossRef](#)]
99. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
100. Sheikh, H.R.; Wang, Z.; Cormack, L.; Bovik, A.C. LIVE Image Quality Assessment Database Release 2. 2015. Available online: <http://live.ece.utexas.edu/research/quality> (accessed on 30 September 2016).
101. Larson, E.C.; Chandler, D. Categorical Image Quality (CSIQ) Database. 2010. Available online: <http://vision.okstate.edu/csiq> (accessed on 30 September 2016).
102. Ponomarenko, N.; Jin, L.; Ieremeiev, O.; Lukin, V.; Egiazarian, K.; Astola, J.; Vozel, B.; Chehdi, K.; Carli, M.; Battisti, F.; et al. Image database TID2013: Peculiarities, results and perspectives. *Signal Process. Image Commun.* **2015**, *30*, 57–77. [[CrossRef](#)]

