

# No-Reference Image Quality Assessment Based on Statistics of Local Ternary Pattern

Pedro Garcia Freitas\*, Welington Y.L. Akamine<sup>†</sup> and Mylène C.Q. Farias<sup>‡</sup>

\*Department of Computer Science,

<sup>†‡</sup>Department of Electrical Engineering,

University of Brasília, Brazil

Email: \*sawp@sawp.com.br, <sup>†</sup>welingtonylakamine@gmail.com, <sup>‡</sup>mylene@ieee.org

**Abstract**—In this paper, we propose a new no-reference image quality assessment (NR-IQA) method that uses a machine learning technique based on Local Ternary Pattern (LTP) descriptors. LTP descriptors are a generalization of Local Binary Pattern (LBP) texture descriptors that provide a significant performance improvement when compared to LBP. More specifically, LTP is less susceptible to noise in uniform regions, but no longer rigidly invariant to gray-level transformation. Due to its insensitivity to noise, LTP descriptors are not able to detect milder image degradation. To tackle this issue, we propose a strategy that uses multiple LTP channels to extract texture information. The prediction algorithm uses the histograms of these LTP channels as features for the training procedure. The proposed method is able to blindly predict image quality, i.e., the method is no-reference (NR). Results show that the proposed method is considerably faster than other state-of-the-art no-reference methods, while maintaining a competitive image quality prediction accuracy.

## I. INTRODUCTION

In recent years, multimedia applications, including image and video services, have become widely popular. As a consequence, the interest in objective methods that are able to estimate the perceived quality of a multimedia content has increased considerably, both in academia and industry. However, modeling a user reaction to a multimedia content is still a challenging problem. Among the conceivable user reactions, one of the most important issues concerns the image quality assessment (IQA).

Objective image quality assessment (IQA) methods can be classified in three categories, according to the amount of the reference image required by the algorithm. Full reference (FR) methods estimate the quality of a test image performing some type of comparison with the reference. Reduced reference (RR) methods use only partial information about the reference image to estimate quality. Since requiring the reference image or even partial reference information is an obstacle for many multimedia applications, the solution is to use no-reference (NR) methods that do not require any information about the reference image.

Although a lot has been done in the area of multimedia quality assessment, most of the achievements have been in the development of FR methods and there is, still, much to be done in no-reference image quality assessment (NR-IQA) methods [1, 2]. A popular NR-IQA approach consists of estimating the strength of the most relevant image distortions and,

then, predicting image quality using a combination of these distortion measures. This approach is known as distortion-specific (DS) because it requires the knowledge of one or more types of distortions. Among the DS-NR-IQA methods, we can cite the works of Chabard *et al.* [3], Li *et al.* [4], Wang *et al.* [5], and Manap & Shao [6].

Methods that do not require a prior knowledge of image distortions are described as non-distortion-specific (NDS). Although more complex, NDS methods are more adequate for diverse multimedia scenarios where several different types of distortions are present. The most common NDS approaches either use the statistics of natural images [7, 8] or machine learning techniques. Among the several NR-IQA methods based on machine learning, we can cite the works of Ye *et al.* [9], Zhang *et al.* [2], and Liu *et al.* [10]. It is worth pointing out that, although machine learning techniques show promising results, they may present limitations in terms of computational complexity and prediction performance.

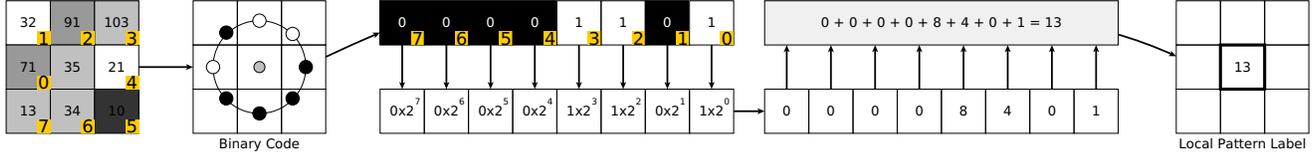
In this paper, we present a method that tackles the aforementioned limitations. The proposed method is a NDS-NR-IQA method that uses machine learning techniques. For training the machine learning algorithm, the proposed method uses the histograms of the local ternary pattern (LTP) as features [11]. This approach enables to blindly predict the image quality, without making any assumptions about the type of distortions that the image may contain. The main advantages of the proposed method are: (1) a high computational efficiency and (2) a quality prediction performance that is comparable to the performance of state-of-the-art NR-IQA methods.

The paper is organized as follows. Section II summarizes the basic concepts of the use of LTP operators for feature extraction. Section III describes the proposed NR-IQA method. Sections IV and V present the experimental setup and results, respectively. Finally, in Section VI we present our conclusions.

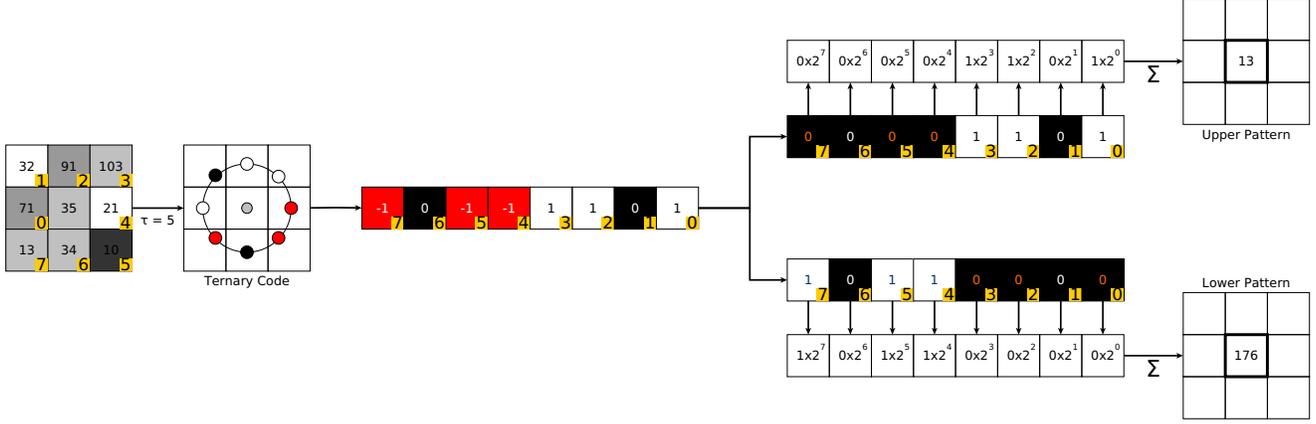
## II. BRIEF REVIEW OF LOCAL TERNARY PATTERN

The Local Ternary Pattern (LTP) operator is a generalization of the Local Binary Pattern (LBP) [12] patchwise texture feature extractor. The LBP operator is formally defined as:

$$LBP_{R,P}(t_c) = \sum_{p=0}^{P-1} S(t_p - t_c)2^p, \quad (1)$$



(a) Illustration of the basic Local Binary Pattern operator.



(b) Illustration of the basic Local Ternary Pattern operator.

Figure 1: Scheme of pattern extraction of a central pixel using LBP (a) and LTP (b) operators with  $R = 1$ ,  $P = 8$ ,  $t_c = 35$ , and  $t_p = \{71, 32, 91, 103, 21, 10, 34, 13\}$ .

where  $t_p$  is the neighboring pixel,  $t_c$  is the grayscale value of the centering pixel,  $R$  is the radius of the neighborhood,  $P$  is the total number of considered neighbors, and

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

The LTP operator extends the definition of the LBP operator in order to generate a code that can assume up to 3 values (-1, 0, or 1). This is achieved by changing the step function  $S$  in the following manner:

$$\hat{S}(t) = \begin{cases} 1, & t \geq \tau, \\ 0, & -\tau < t < \tau \\ -1, & t < -\tau, \end{cases} \quad (3)$$

where  $\tau$  is a threshold which determines how sharp an intensity change should be in order to be considered as an edge. After computing the ternary codes using the above equation, each ternary pattern is split into two codes: a positive (upper pattern) and a negative (lower pattern) code. These codes are treated as two separate channels of LBP descriptors.

Fig. 1 illustrates the basic feature extraction procedure for a single pixel, using LBP and LTP operators. The numbers in yellow squares represent the order in which the step function is computed (Eqs. 2 and 3). Fig. 1-(a) depicts the steps for using the LBP operator, considering a unitary neighborhood radius ( $R = 1$ ) and eight neighboring pixels ( $P = 8$ ). In the LBP case, the binary code takes only two values: 0 (black) or 1 (white). Using the Eq. 1, we obtain a LBP label for the central pixel, taking its neighborhood.

Fig. 1-(b) depicts the steps of the LTP operator, considering a unitary neighborhood radius ( $R = 1$ ) and eight neighboring pixels ( $P = 8$ ). Here, the threshold  $\tau$  is set to five. The LTP operator generates three possible values (see Eq. 3), which are represented by the colors black ( $\hat{S}(t) = 1$ ), white ( $\hat{S}(t) = 0$ ), and red ( $\hat{S}(t) = -1$ ). Following the same counterclockwise order used for the LTP operator, we split the ternary code into two LBP codes with only positive values. First, we create the upper pattern by converting the negative codes to zero and using Eq. 1. Next, we create the lower pattern by setting the positive values to zero, converting the negative values to positive, and using Eq. 1.

By comparing Figs. 1-(a) and (b), we can notice that the LTP operator extends the LBP operator and, consequently, generates two texture information maps. Treating these maps as two separate channels of LBP descriptors, we compute independent histograms and similarity measures. Results are combined at the end of the process to generate texture features. In the next section, we present how to use this texture information to design an image quality assessment method.

### III. IMAGE QUALITY ASSESSMENT USING LTP

In general, LTP parameters must be adjusted to the target application. One important parameter that needs to be adequately chosen is the threshold  $\tau$  in Eq. 3. In this section, we discuss how to choose this threshold and how to use the LTP operator for blindly estimating image quality.

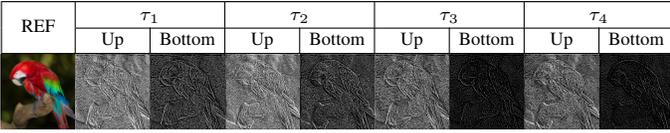


Figure 2: Reference image and its upper and lower patterns generated using the Local Ternary Pattern (LTP) operator with four different threshold values.

### A. Threshold Choice

The threshold  $\tau$  defined in Eq. 3 was proposed by Opitz *et al.* [11], who estimated local thresholds from the directional gradient magnitude image. Anthimopoulos *et al.* [13] demonstrated that the  $\tau$  values actually correspond to the gradient of the image. We can assume that the probability density function (PDF) of an image gradient is Laplacian. For this reason, the PDF of the absolute gradient values is exponential.

According to Anthimopoulos *et al.* [13], the choice of the threshold  $\tau$  affects the discrimination between edge and non-edge pixels, which is a necessary step to generate edge patterns. Choosing an optimal set of thresholds for the multilevel edge description operation makes it possible to group gradient PDFs in clusters. With this goal, the image gradients are fit using an exponential distribution:

$$PDF_e(z) = \lambda e^{-\lambda z}, \quad (4)$$

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

$$F_e(\Delta_i) = \lambda^{-1} \ln(1 - \Delta_i), \quad (5)$$

where

$$\Delta_i = \frac{i}{L+1}. \quad (6)$$

Since  $i \in \{1, 2, \dots, L\}$  and  $L$  is the total number of levels,  $\Delta_i \in [0, 1)$ . To select a threshold, we make

$$\tau_i = F_e(\Delta_i) \quad (7)$$

for equally spaced values.

### B. Feature Extraction

The feature extraction process is illustrated in Fig. 3. First, we decompose the image into LTP channels. These channels are generated by varying the  $\tau$  values according to Eq. 5, 6, and 7. As described in Section II and depicted in Fig. 1-(b), for a single image, the LTP operator produces two channels, one corresponding to the upper patterns and the other one to the lower patterns. Therefore, for  $L$  numbers of  $\tau_i$ , we have  $2L$  LTP channels. These channels are illustrated in Figure 2. In this figure, we use  $L = 4$ , what generates eight distinct LTP channels. In the proposed LTP approach, instead of computing the differences between  $t_c$  and its neighbors on a grayscale image, we take the maximum difference on R, G, or B channels.

After the aforementioned steps are completed, we obtain a set of LTP channels with  $2 \times L$  elements:

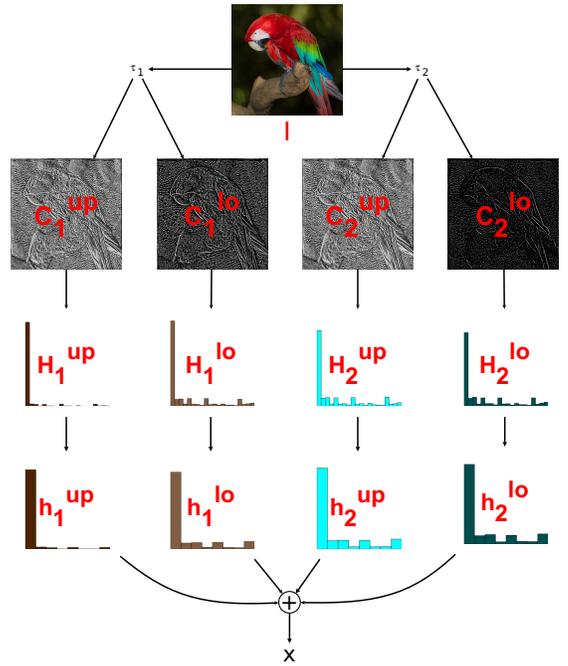


Figure 3: Illustration of process of extracting the feature vector  $x$  with  $L = 2$ .

$\{C_1^{up}, C_1^{lo}, C_2^{up}, C_2^{lo}, \dots, C_L^{up}, C_L^{lo}\}$ . In this set, the subscript index corresponds to the  $i$ -th  $\tau$  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 are reduced according to the following formula:

$$k_i^j = \left\lfloor \frac{\max H_i^j - \min H_i^j}{n} \right\rfloor, \quad (8)$$

where  $\lfloor \cdot \rfloor$  represents the operation of rounding to the nearest integer,  $n$  defines the number of equal-width bins, and  $k_i^j$  is the resulting reduced number of bins of the histogram  $H_i^j$ . Therefore, 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 to the image  $I$ . More specifically, the feature vector  $x$  is generated by concatenating the quantized histograms  $h_i^j$ , i.e.:

$$x = h_1^{up} \oplus h_1^{lo} \oplus h_2^{up} \oplus h_2^{lo} \oplus \dots \oplus h_L^{up} \oplus h_L^{lo}, \quad (9)$$

where  $\oplus$  is the concatenation operator and  $x$  is the feature vector used as input to the support vector regression algorithm.

### C. Support Vector Regression

A support vector regression (SVR) algorithm is used to predict quality from the feature vector  $x$ . The SVR machine

Method	JPEG			JPEG2k			WN			GB			FF			ALL		
	SROCC	LCC	KRCC															
PSNR	0.8515	0.8544	0.6439	0.8822	0.8668	0.6937	<b>0.9856</b>	0.9778	0.8933	0.7818	0.7749	0.5862	0.8869	0.8725	0.7002	0.8013	0.7633	0.5964
SSIM	<b>0.9480</b>	0.8091	<b>0.8087</b>	<b>0.9438</b>	0.8225	<b>0.8065</b>	0.9793	0.9372	0.8816	0.8889	0.7990	0.7330	0.9335	0.8084	0.7893	0.8902	0.9060	0.7228
BRISQUE	0.9026	0.9277	0.7440	0.9109	0.9177	0.7504	0.9734	0.9850	0.8818	0.9618	<b>0.9671</b>	0.8522	0.8830	0.9093	0.7021	0.9336	0.9333	0.7775
CORNIA	0.9240	0.9437	0.7739	0.9326	0.9344	0.7867	0.9672	0.9685	0.8626	<b>0.9738</b>	0.9690	<b>0.8798</b>	0.9335	0.9277	0.7929	<b>0.9530</b>	0.9445	<b>0.8156</b>
CQA	0.8867	0.9022	0.7183	0.8967	0.8992	0.7255	0.9823	<b>0.9912</b>	0.9064	0.9131	0.9227	0.7537	0.8704	0.8920	0.7044	0.9102	0.9049	0.7403
SSEQ	0.8848	0.9087	0.7037	0.9086	0.9145	0.7398	0.9616	0.9727	<b>0.9261</b>	0.9362	0.9403	0.8030	0.8527	0.8630	0.6897	0.9006	0.8945	0.7356
PROPOSED	<i>0.9422</i>	<b>0.9485</b>	<i>0.8011</i>	<i>0.9418</i>	<b>0.9483</b>	<i>0.8010</i>	0.9435	0.9495	0.8028	0.9423	0.9485	0.8012	<b>0.9420</b>	<b>0.9484</b>	<b>0.8012</b>	0.9420	<b>0.9487</b>	0.8022

(a) Median SROCC, LCC, and KRCC of simulations on the LIVE2 database

Method	JPEG		JPEG2k		WN		GB		PN		CD		ALL	
	SROCC	LCC												
PSNR	0.9009	0.8936	<b>0.9309</b>	0.9326	<b>0.9345</b>	<b>0.9428</b>	<b>0.9358</b>	0.9082	<b>0.9315</b>	<b>0.9546</b>	<b>0.8862</b>	<b>0.8992</b>	0.8088	0.7857
SSIM	<b>0.9309</b>	0.8746	0.9251	0.8752	0.8760	0.8549	0.9089	0.8097	0.8870	0.8375	0.8128	0.8187	0.8116	0.7219
BRISQUE	0.7121	0.8037	0.7738	0.8104	0.6425	0.6854	0.6021	0.7324	0.7746	0.7974	0.5080	0.5919	0.6877	0.7605
CORNIA	0.8743	0.9228	0.9033	0.9303	0.8381	0.8565	<i>0.9150</i>	0.9481	0.6687	0.6711	0.6131	0.6659	0.7865	0.8322
CQA	0.6316	0.7535	0.8274	0.8702	0.6547	0.6843	0.6180	0.6871	0.7319	0.7613	0.5114	0.5285	0.6529	0.7013
SSEQ	0.8756	0.9041	0.8667	0.9046	<i>0.9164</i>	<i>0.9239</i>	0.8972	<b>0.9288</b>	<i>0.8253</i>	0.8115	<i>0.7031</i>	<i>0.7545</i>	0.8428	0.8578
PROPOSED	<i>0.9170</i>	<b>0.9379</b>	<i>0.9172</i>	<b>0.9371</b>	0.8319	0.8337	0.9017	0.9212	0.8063	<i>0.8192</i>	0.0539	0.0560	<b>0.8636</b>	<b>0.8795</b>

(b) Median SROCC and LCC of simulations on the CSIQ database.

	SROCC							KRCC						
	PSNR	SSIM	BRISQUE	CORNIA	CQA	SSEQ	PROPOSED	PSNR	SSIM	BRISQUE	CORNIA	CQA	SSEQ	PROPOSED
AGC	0.8568	0.7912	<b>0.9289</b>	0.7362	0.9192	0.9269	0.7523	0.7719	0.5989	<b>0.7990</b>	0.5467	0.7723	0.7767	0.5631
AGN	<b>0.9337</b>	0.6421	0.8559	0.4046	0.6436	0.8321	0.8623	0.6599	0.4780	<b>0.7023</b>	0.2867	0.4721	0.6578	0.6867
CA	0.7759	0.7158	0.8460	0.7450	0.6504	0.7665	<b>0.8739</b>	0.5716	0.5390	0.7033	0.1333	0.4933	0.5967	<b>0.7133</b>
CC	<b>0.4608</b>	0.3477	0.0143	<i>0.1992</i>	0.3093	0.0204	0.1196	<b>0.3090</b>	0.2499	0.0150	0.1200	<i>0.2200</i>	0.0134	0.0800
CCS	0.6892	<b>0.7641</b>	0.1265	0.1781	0.1562	0.3098	<i>0.7223</i>	0.4959	0.5579	0.0933	<b>0.5800</b>	0.1000	0.2218	0.5421
CN	<b>0.8838</b>	0.6465	0.6357	0.7577	0.1025	0.3211	<i>0.7685</i>	<b>0.6940</b>	0.4750	0.5154	0.5588	0.0701	0.2174	<i>0.5867</i>
GB	0.8905	0.8196	0.8844	0.9008	0.9169	0.8715	<b>0.9423</b>	0.7877	0.6746	0.7690	0.7400	0.7600	0.6933	<b>0.8133</b>
HFN	0.9165	0.7962	0.8689	0.8531	0.9238	<b>0.9270</b>	0.9000	0.7294	0.6056	0.7390	0.6633	<b>0.7800</b>	0.7790	0.7379
ICQ	<b>0.9087</b>	0.7271	0.8436	0.7931	0.8196	<i>0.9008</i>	0.8743	0.7315	0.5498	0.6800	0.5933	0.6333	<b>0.7400</b>	0.7000
ID	<b>0.9457</b>	0.8327	<i>0.8854</i>	0.7665	0.8403	0.8438	0.8726	<b>0.8032</b>	0.6501	<i>0.7513</i>	0.5733	0.6833	0.6733	0.7133
IN	<b>0.9263</b>	0.8055	0.8663	0.6962	0.6531	0.8310	<i>0.8962</i>	0.7719	0.5944	0.7089	0.4967	0.4942	0.6522	<b>0.7333</b>
IS	<b>0.7647</b>	0.7410	<i>0.4497</i>	0.0896	0.1396	0.3704	0.3158	<b>0.5560</b>	0.5369	<i>0.3167</i>	0.0733	0.0935	0.2552	0.2200
JPEG	<b>0.9252</b>	0.8275	0.8145	0.8467	0.7805	0.8315	<i>0.8877</i>	<b>0.7698</b>	0.6582	0.6567	0.6622	0.5922	0.6533	<i>0.7133</i>
JPEGTE	0.7874	0.6144	0.6709	0.7450	0.3842	0.5096	<b>0.8138</b>	0.5886	0.4474	0.5221	0.5733	0.2733	0.3753	<b>0.6311</b>
JPEG2k	0.8934	0.7531	<b>0.9119</b>	0.8788	0.8881	0.8819	0.9015	0.7112	0.5924	<b>0.7656</b>	0.7067	0.7156	0.7067	0.7400
JPEG2kTE	<b>0.8581</b>	0.7067	0.6151	<i>0.7085</i>	0.6496	0.6878	0.6815	<b>0.6721</b>	0.5425	0.4621	<i>0.5221</i>	0.4733	0.5188	0.5000
LBD	0.1300	0.6213	0.6915	0.2258	0.3250	0.5343	<b>0.6969</b>	0.0965	0.4356	0.5267	0.1703	0.2433	0.3920	<b>0.5268</b>
LC	<b>0.9386</b>	0.8310	<i>0.9144</i>	0.8269	0.5691	0.7019	0.8607	<b>0.7729</b>	0.6482	<i>0.7600</i>	0.6544	0.4187	0.5354	0.6867
MGN	0.9085	0.7863	<b>0.9287</b>	0.6692	0.8870	0.7938	0.9202	0.7207	0.5315	<b>0.7857</b>	0.4900	0.7233	0.6155	0.7723
MN	<b>0.8385</b>	0.7388	<i>0.8332</i>	0.4846	0.6858	0.7807	0.7928	0.6413	0.5899	<b>0.6544</b>	0.3267	0.5000	0.5933	0.6133
NEPN	<b>0.6930</b>	0.5326	<i>0.4708</i>	0.3893	0.0777	0.0700	0.0750	<b>0.4938</b>	0.3780	<i>0.3253</i>	0.2800	0.0517	0.0517	0.0600
QN	0.8636	0.7428	0.7945	0.6550	0.6777	<b>0.8867</b>	0.8223	<b>0.6810</b>	0.5562	0.6155	0.4900	0.4867	<i>0.7179</i>	0.6377
SCN	0.9152	0.7934	0.8634	<b>0.9231</b>	0.8722	0.8704	0.9054	0.7417	0.5731	<b>0.8204</b>	0.5600	0.6467	0.7533	0.7467
SSR	<b>0.9241</b>	0.7774	<i>0.9487</i>	0.7657	0.8403	0.9158	0.9015	0.7351	0.6122	0.6989	<b>0.7733</b>	0.6989	0.6900	0.7356
ALL	0.6869	0.5758	0.7939	0.7181	0.6296	0.7441	<b>0.8408</b>	0.4958	0.4079	0.6162	0.5344	0.4557	0.5611	<b>0.6600</b>

(c) Median SROCC and KRCC of simulations on the TID2013 database.

Table I: Correlation indexes across 100 train-test simulations on (a) LIVE2, (b), CSIQ, and (c) TID2013 databases.

learning algorithm was chosen because its use in other NR-IQA approaches [7, 9] has provided a good performance. Also, SVR is a robust algorithm feature spaces with high dimensions [14, 15]. To train the quality model with SVR, the feature vectors are mapped to subjective quality scores provided in the quality databases:

$$Q(I) = SVR(x, \mathcal{M}), \quad (10)$$

where  $\mathcal{M}$  is the trained model for regression and  $Q(I)$  is the objective quality score predicted using the model.

#### IV. EXPERIMENTAL SETUP

In our experiments, the implementation of SVR uses Lib-SVM on a Python interface provided by Scikit library [16]. The kernel, penalty parameter, epsilon, and other meta-parameters of SVR are found using exhaustive grid search methods provided by Sklearn’s API [16]. The parameters of the proposed algorithm are  $L = 4$ , which gives 8 LTP channels,  $R = 1$ , and  $P = 8$ . To generate the set of quantized histograms  $h_i^j$ , each histogram  $H_i^j$  is reduced from 256 bins to 18 bins.

The proposed method is tested using the CSIQ [17], LIVE2 [18], and TID2013 [19] image quality databases. The CSIQ database has a total of 866 test images, consisting of 30 originals and 6 different types of distortions. The LIVE2 database has 982 test images, including 29 originals and 5 types of distortions. The distortions included in these databases are JPEG, JPEG 2000 (JPEG2k), white noise (WN), Gaussian blur (GB), fast fading (FF), global contrast decrements (CD), and additive Gaussian pink noise (PN). The TID2013 database contains 25 reference images with the following distortion types: 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 with transmission errors (JPEGTE), JPEG2k with 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).

We compare the proposed method with the fastest state-of-the-art NR-IQA methods: BRISQUE [20], CORNIA [9], CQA [21], and SSEQ [22]. Moreover, we also compare the proposed algorithm with PSNR and SSIM [23], which are two well-established FR-IQA metrics.

The performance of testing methods is measured using three tradition statistical measures: Spearman’s Rank Ordered Correlation (SROCC), Pearson (linear) Correlation Coefficient (LCC), and Kendall’s Rank Correlation Coefficient (KRCC). The correlation coefficients are computed considering the predicted scores obtained using the IQA methods and the corresponding subjective scores provided in the databases. For training the NR-IQA methods, the databases are split into two subsets, randomly selected, with 80% of data used for training and 20% for testing in each simulation. All reported results are the median of the correlation values computed for 100 random combinations of training and testing subsets.

#### V. EXPERIMENTAL RESULTS

Tables I (a)-(c) show the correlation coefficients obtained using the considered IQA methods on the LIVE2, CSIQ, and TID2013 databases, respectively. For each database, the tables show SROCC, LCC, or KRCC values obtained for the sets of images containing each distortion type and for the complete set of images (ALL). In these tables, the numbers in bold depict the best correlation results among all FR and NR metrics. On the other hand, the italicized numbers depict the best correlation results considering only NR metrics.

For LIVE2 (Table I-a), we can notice that the proposed method outperforms most of the NR-IQA methods considered in the test. This is depicted in Fig. 4, which presents the box plot of SROCC, LCC, and KRCC distributions for different NR-IQA methods tested on the LIVE2 database. From these graphs, we can notice that our method and the CORNIA method have similar performances. Nevertheless, CORNIA is the most computationally expensive method tested in our work, as depicted in Table II.

For the CSIQ database, the proposed method achieves a statistically better performance than the other NR-IQA methods (see Table I-b). Interestingly, NR approaches perform worse than PSNR for this database. It is worth pointing out that these PSNR scores are consistent with the values reported by Larson and Chandler [17]. Even though the proposed method presents the best overall results, it has a poor performance for CD distortions. We believe the robustness of the LTP operators to changes in contrast affects its prediction performance. Further studies are needed to determine whether the performance can be improved with the inclusion of additional features that are sensitive to contrast changes.

For the TID2013 (Table I-c) database, the proposed method performs statistically better than the other methods for the general case (ALL). In terms of both SROCC and KRCC, the proposed method outperforms many of the considered NR-IQA methods. But, similarly to what was obtained for the CSIQ database, the performance of the proposed method has a lower performance for images with contrast and intensity distortions.

Since the main goal of this work is to design a NR-IQA method with a good performance and a low computational complexity, we compared the average time each IQA method takes to compute a single image quality score. These average times are reported in Table II. These values were obtained using an Intel i7-4790 processor at 3.60GHz. The average times were computed using images of sizes 512x512, 480x720, 610x488, 618x453, 627x482, 632x505, 634x438, 634x505, 640x512, 768x512, 1280x1600, 1280x1510, or 1280x1506. By comparing the results presented in Tables I and II, we can notice that the proposed methods, BRISQUE and CORNIA methods have similar prediction accuracy performances in terms of correlation values. However, the proposed method is 4 times faster than BRISQUE and 48 times faster than CORNIA.

	PSNR	SSIM	BRISQUE	CORNIA	CQA	SSEQ	PROPOSED
Time	0.0055	0.0447	0.1576	1.8964	1.3691	1.8112	0.0392

Table II: Average computational time (in seconds).

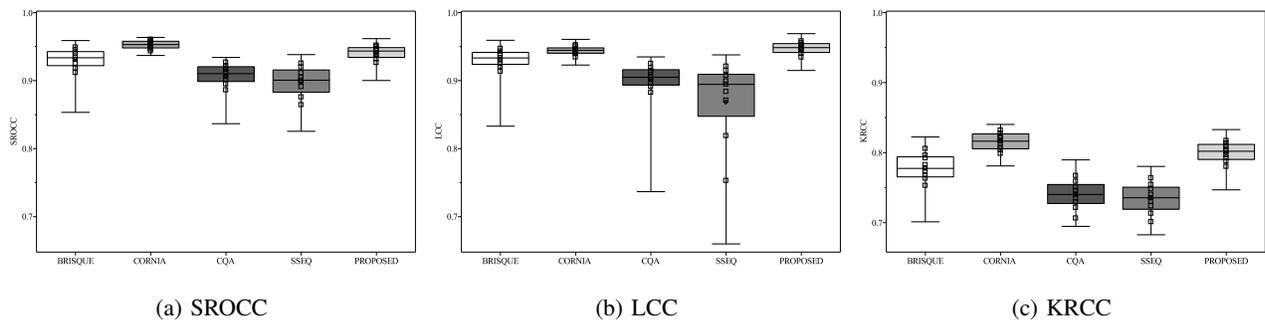


Figure 4: Box plot of (a) SROCC, (b) LCC, and (c) KRCC distributions of NR algorithms from 100 runs of simulations using the LIVE2 database.

## VI. CONCLUSIONS

In this paper, we proposed a novel method for blindly assessing the quality of images with no previous assumptions about the type of image distortions. It uses a machine learning technique based on texture descriptor features. Results show that the proposed method has a good prediction accuracy when compared to state-of-the-art NR-IQA methods. Moreover, the proposed method is faster than other methods found in literature. Future works include the investigation of the impact of the method parameters on the prediction accuracy. Furthermore, it is worth investigating how the addition of contrast features can improve the prediction performance for images affected by this type of distortion.

## ACKNOWLEDGMENT

This work was supported by the University of Brasília (UnB), the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and the Fundação de Empreendimentos Científicos e Tecnológicos (Finatec).

## REFERENCES

- [1] D. M. Chandler, "Seven challenges in image quality assessment: past, present, and future research," *ISRN Signal Processing*, vol. 2013, 2013.
- [2] M. Zhang, C. Muramatsu, X. Zhou, T. Hara, and H. Fujita, "Blind image quality assessment using the joint statistics of generalized local binary pattern," *Signal Processing Letters, IEEE*, vol. 22, no. 2, pp. 207–210, 2015.
- [3] T. Chabardès and B. Marcotegui, "Local blur estimation based on toggle mapping," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2015, pp. 146–156.
- [4] L. Li, Y. Zhou, W. Lin, J. Wu, X. Zhang, and B. Chen, "No-reference quality assessment of deblocked images," *Neurocomputing*, 2015.
- [5] S. Wang, C. Deng, B. Zhao, G.-B. Huang, and B. Wang, "Gradient-based no-reference image blur assessment using extreme learning machine," *Neurocomputing*, vol. 174, pp. 310–321, 2016.
- [6] R. A. Manap and L. Shao, "Non-distortion-specific no-reference image quality assessment: A survey," *Information Sciences*, vol. 301, pp. 141–160, 2015.
- [7] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3350–3364, 2011.
- [8] Y. Fang, K. Ma, Z. Wang, W. Lin, Z. Fang, and G. Zhai, "No-reference quality assessment of contrast-distorted images based on natural scene statistics," *Signal Processing Letters, IEEE*, vol. 22, no. 7, pp. 838–842, 2015.
- [9] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1098–1105.
- [10] L. Liu, Y. Hua, Q. Zhao, H. Huang, and A. C. Bovik, "Blind image quality assessment by relative gradient statistics and adaboosting neural network," *Signal Processing: Image Communication*, vol. 40, pp. 1–15, 2016.
- [11] M. Opitz, M. Diem, S. Fiel, F. Kleber, and R. Sablatnig, "End-to-end text recognition using local ternary patterns, msr and deep convolutional nets," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. IEEE, 2014, pp. 186–190.
- [12] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [13] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "Detection of artificial and scene text in images and video frames," *Pattern Analysis and Applications*, vol. 16, no. 3, pp. 431–446, 2013.
- [14] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [15] M. Minghui and Z. Chuanfeng, "Application of support vector machines to a small-sample prediction," *Advances in Petroleum Exploration and Development*, vol. 10, no. 2, pp. 72–75, 2016.
- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [17] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of Electronic Imaging*, vol. 19, no. 1, p. 011006, 2010.
- [18] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3440–3451, 2006.
- [19] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti *et al.*, "Image database tid2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57–77, 2015.
- [20] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *Image Processing, IEEE Transactions on*, vol. 21, no. 12, pp. 4695–4708, 2012.
- [21] L. Liu, H. Dong, H. Huang, and A. C. Bovik, "No-reference image quality assessment in curvelet domain," *Signal Processing: Image Communication*, vol. 29, no. 4, pp. 494–505, 2014.
- [22] L. Liu, B. Liu, H. Huang, and A. C. Bovik, "No-reference image quality assessment based on spatial and spectral entropies," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 856–863, 2014.
- [23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.