Training Quality-Aware Filters for No-Reference Image Quality Assessment

Lin Zhang
Tongji University and Nanjing University of Science and Technology, China

Zhongyi Gu, Xiaoxu Liu, Hongyu Li, and Jianwei Lu
Tongji University, China

The proposed general-purpose, no-reference image quality assessment (NR-IQA) model does not require prior knowledge about nondistorted reference images or the types of image distortions.

Objective image quality assessment (IQA) aims to develop computational models that can measure image quality consistently in subjective evaluations. With the now widespread use of digital imaging for representation and communication, IQA has become an essential, yet challenging problem. Potential applications exist in the image processing and computer vision fields, such as image acquisition, transmission, compression, restoration, and enhancement. Because subjective IQA methods cannot be readily and routinely used in many scenarios (such as real-time and automated systems), it is necessary to develop objective IQA metrics to automatically and robustly measure an image's quality. Based on the availability of nondistorted reference images, IQA approaches can be classified into three categories: full-reference IQA (FR-IQA), reduced-reference IQA (RR-IQA), and no-reference IQA (NR-IQA).

In this article, we propose a general-purpose opinion-aware NR-IQA method that utilizes quality-aware filters (QAFs). Our method follows a training-test framework. In the training stage, we first construct a filter dictionary by performing sparse filtering[1] on a set of local descriptor vectors extracted from image patches with different quality levels. We then use a training dataset consisting of a set of distorted images and their associated subjective scores to train the regression model. For any training image, we first represent it as a set of local descriptors and then, by performing local descriptor encoding and feature pooling, finally obtain a quality-aware feature vector to represent the image. Next, we adopt a random forest[2] to train the regression model, mapping from the feature vectors to the subjective scores. In the test stage, given a test image, a quality-aware feature vector is extracted and then fed into the learned regression model to predict the image's perceptual quality.

The key components of QAF are the sparse filtering and random forest. Sparse filtering is typically used to create a filter dictionary, which maps the original data to good feature representations for classification tasks by optimizing exclusively for sparsity in the feature distribution. Here, we apply sparse filtering to a set of local descriptors extracted from image patches to develop a QAF dictionary. The term "quality aware" means that one filter in the dictionary exists for each feature with a certain distortion degree and vice versa. Using this dictionary for image encoding with max pooling, we can obtain an effective image representation for quality prediction. The random forest is an ensemble learning method that operates by constructing a multitude of decision trees during the training stage and predicting new data by averaging the predictions of all the trees. Because this prediction process is close to the subjective IQA process, we use the random forest instead of a support vector machine (SVM) as the regression model to map the feature vectors to the subjective quality scores. SVMs with various kernels are commonly used in existing NR-IQA methods for regression, but in our experiments, we found that the random forest achieved significantly better results.

1070-986X/14/$31.00/© 2014 IEEE

Previous Image Quality Assessment Approaches

For simplicity, most early NR-IQA algorithms assume that the image under consideration is affected by one or several types of distortion, such as blocking, ringing, blur, and compression. Generally, these approaches extract distortion-specific features that relate to the loss of visual quality. They can perform blind IQA only when the type of distortion is known beforehand. Hence, the applicability of these kinds of methods is limited.

In contrast, the goal of general-purpose non-distortion-specific (NDS) NR-IQA approaches is to predict an image's quality without any prior knowledge of the distortion type. Depending on the availability of subjective scores when training the prediction model, existing NDS NR-IQA approaches can be classified as either opinion-aware or opinion-unaware.

A majority of the existing NR-IQA methods are opinion-aware approaches, which means that they are trained on a dataset consisting of both distorted images and the associated subjective scores. These works generally share a similar architecture. In the training stage, feature vectors are extracted from the distorted images, and then a regression model is learned to map the feature vectors to the associated human subjective scores. In the testing stage, a feature vector is extracted from the test image and then fed into the learned regression model to predict an image's quality score.

One representative opinion-aware NR-IQA method is the two-step framework.[3,4] Anush Moorthy and Alan Bovik proposed a two-step framework called BIQI (Blind Image Quality Index).[3] In BIQI, given a distorted image, scene statistics are at first extracted to explicitly classify the distorted image into one of several known distortions. Then, the same set of statistics is used to evaluate the distortion-specific quality. Following the same paradigm, the authors later extended BIQI to develop DIIVINE (Distorted Identification-Based Image Verity and Integrity Evaluation Index).[4] Both BIQI and DIIVINE assume that the distortion types in the test images should be covered by the training dataset, which is obviously not true in many practical applications.

Opinion-aware approaches that do not follow a two-step framework typically follow two trends:[3] natural scene statistics (NSS) -based and training-based methods. NSS methods are based on the assumption that the quality distortions in images will affect certain statistical properties in natural scenes. For example, Michele Saad, Alan Bovik, and Christophe Charrier proposed BLIINDS (Blind Image Integrity Notator Using DCT Statistics) by assuming that the statistics of discrete cosine transform (DCT) features will vary in a predictable way as image quality changes.[6] Later, they proposed BLIINDS-II[7] as an extension of their previous work. Anish Mittal, Anush Moorthy, and Alan Bovik proposed using locally normalized luminance coefficients in the spatial domain to predict image quality.[8] Their design rationale was that the presence of distortion will affect the regular structure of the image's coefficients.

Training-based methods aim to design quality-relevant features that can capture the factors that may impact distortion. Most of these training-based approaches need to design numerous handcrafted features. Peng Ye and David Doermann proposed a codebook-based framework called CBIQ (Codebook Based Image Quality Index) to train the regression model.[5] Their codebook-based framework was commonly applied to image classification tasks. Later, using the same framework, Ye and his colleagues improved CBIQ by employing features trained via unsupervised feature learning, and the resulting NR-IQA metric was named CORNIA (Codebook Representation for No-Reference Image Assessment).[9] CORNIA has proven effective in dealing with many distortion types. Another proposed model extracts three sets of features based on the statistics of natural images, distortion textures, and blur/noise.[10] That approach then trains three regression models for each feature set, and finally a weighted combination of the models is used to estimate image quality.

Unlike opinion-aware methods, opinion-unaware approaches do not require training on databases associated with human scores. Wufeng Xue, Lei Zhang, and Xuanqin Mou simulated a virtual dataset[11] in which the quality scores of distorted images are estimated by the full reference IQA algorithm feature similarity (FSIM).[12] They then proposed a quality-aware clustering method to train a set of centroids for each quality level and used these centroids as a codebook to infer the quality of each patch in a given image. However, their method can only deal with four specific types of distortions: Gaussian noise, Gaussian blur, JPEG compression, and JPEG2000 compression. Thus, strictly speaking, their work is not a

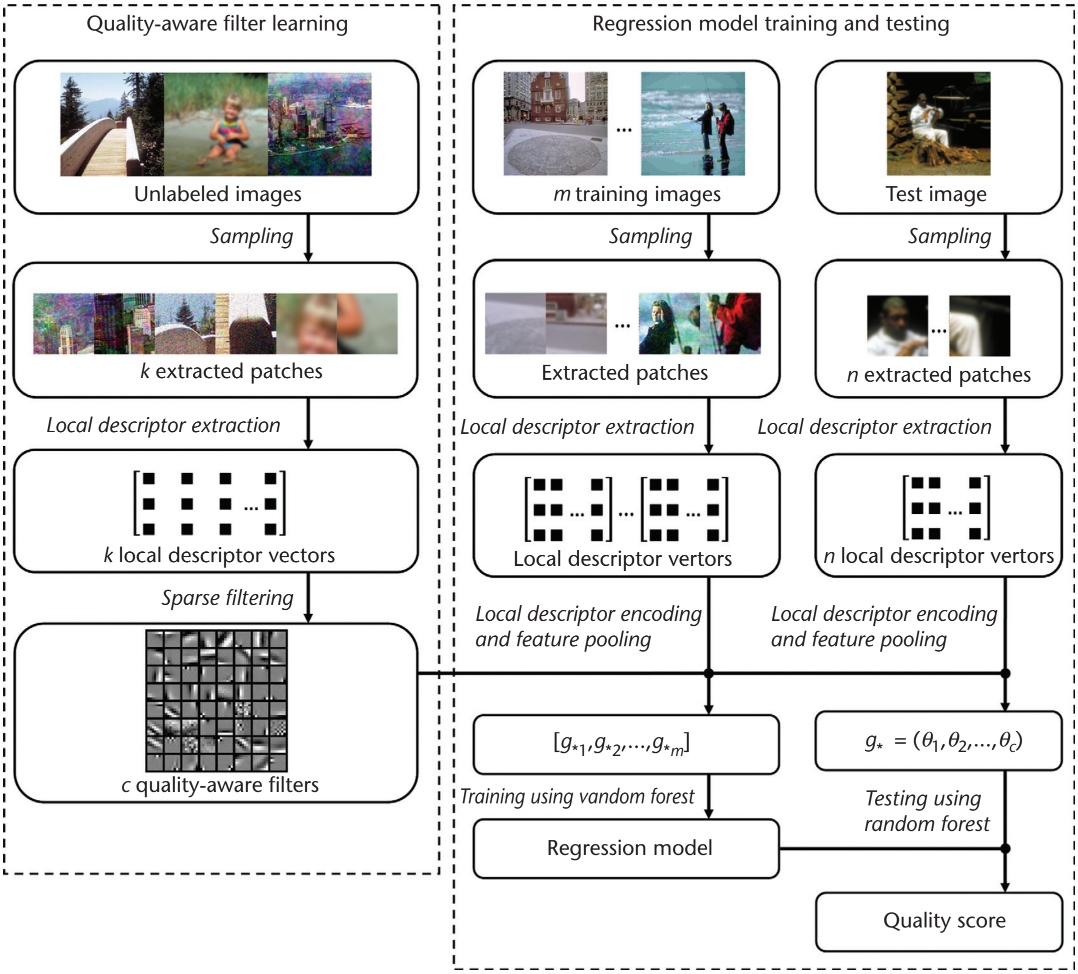

Figure 1. Pipeline of the proposed no-reference image quality assessment (NR-IQA) metric quality-aware filter (QAF). The key components are local descriptor extraction, QAF dictionary construction, local feature encoding and pooling, and regression.

general-purpose NR-IQA metric. The NIQE method extracts a set of local features from an image and fits the feature vectors by a multivariate Gaussian (MVG) model.[13] A test image's quality is predicted by the distance between its MVG model and the MVG model learned from a set of pristine images. However, NIQE uses a single MVG model to describe an image and loses much useful information when characterizing image quality. Until now, according to the experiments conducted on LIVE[14] and CSIQ[15] databases, the prediction accuracy of opinion-unaware methods was lower than that of opinion-aware methods.

QAF Framework

Figure 1 illustrates the QAF pipeline. The key components in our proposed NR-IQA framework are the extraction of local descriptors, the construction of a QAF dictionary, local feature encoding and pooling, and regression.

Extracting Local Descriptors

In our approach, an image is represented by a set of local descriptors, each of which is extracted from a randomly sampled patch. With respect to the local descriptor extraction, we resort to mean subtracted contrast normalized (MSCN) coefficients,[16] the products of pairs of adjacent MSCN coefficients,[13] and the responses of Gabor filters.[5]

The MSCN procedure can be seen as a normalization process with respect to local brightness and contrast, which can be described as

$$\rho(x, y) = \frac{I(x, y) - \mu(x, y)}{\sigma(x, y) + 1},$$

where I is a given grayscale image, x and y are spatial coordinates, and

$$\mu(x, y) = \sum_{\alpha=-K}^{K} \sum_{\beta=-L}^{L} w_{\alpha,\beta} I(x + \alpha, y + \beta)$$

$$\sigma(x,y) = \sqrt{\sum_{\alpha=-K}^{K}\sum_{\beta=-L}^{L} w_{\alpha,\beta}[I(x+\alpha,y+\beta)-\mu(x,y)]^2}$$

estimate the local mean and contrast, respectively, where $w=\{w_{\alpha,\beta}|\alpha=-K,\ldots,K,\beta=-L,\ldots,L\}$ is a 2D circularly-symmetric Gaussian weighting function sampled out to three standard deviations and rescaled to unit volume.

In addition to MSCN coefficients, we also extract the products of adjacent MSCN coefficient pairs as features. Specifically, we extract four products along horizontal, vertical, and diagonal orientations: $\rho(x,y)\rho(x,y+1)$, $\rho(x,y)\rho(x+1,y)$, $\rho(x,y)\ \rho(x+1,y+1)$, and $\rho(x,y)\ \rho(x+1,y-1)$.

Research has shown that an image's multiscale and multiorientation filtering responses are useful for NR-IQA tasks.[12] For the multiscale and multiorientation filter, we adopt the Gabor filter. A 2D Gabor filter has a real component and an imaginary component, which we can define as

$$g_e(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x'^2}{\sigma^2}+\frac{y'^2}{(\gamma\sigma)^2}\right)\right)\cos\left(\frac{2\phi}{\lambda}x'\right)$$

$$g_o(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x'^2}{\sigma^2}+\frac{y'^2}{(\gamma\sigma)^2}\right)\right)\sin\left(\frac{2\pi}{\lambda}x'\right)$$

where $x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta$, λ is the wavelength of the sinusoid factor, θ is the orientation of the normal to the parallel stripes of the Gabor function, σ is the standard deviation of the Gaussian envelop, and γ is its aspect ratio.

Given an image patch P, the modulus of its response to the Gabor filter $g(x,y;\lambda,\theta)$ can be written as

$$G(x,y;\lambda,\theta) = abs(P(x,y) * g(x,y;\lambda,\theta)).$$

Then, we use the mean and a variance of $\{G(x,y;\lambda,\theta)\}_{(x,y)\in P}$ as the features for P. To capture the multiscale image properties, we use Gabor filters with S different wavelengths and J different orientations. Thus, for each patch P, we can generate a feature vector that is $2\times S\times J$ long.

For a given image, we sample from it n patches of size $M\times N$. For each patch $P\in R^{M\times N}$, after we compute and vectorize its MSCN coefficients, its four products of adjacent MSCN coefficient pairs, and Gabor-based features, we can obtain a patch-level feature vector as $\mathbf{f} = (p_1, p_2, \ldots, p_{M\times N\times(1+4)+J\times S\times 2})^T$. Therefore, each image can be represented by a set of descriptor vectors $F = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n]$, where each $\mathbf{f}_i$ is a local descriptor vector and n is the number of patches sampled.

Constructing a Quality-Aware Filter Dictionary

Unsupervised feature learning can be effective in training representations that perform well on classification tasks, such as image, video, and audio classification. In our approach, we treat the patch-level quality estimation as a classification problem. After we classify each patch-level feature into one category, we can obtain an image-level feature for regression. The key to this implementation is the construction of a QAF dictionary.

First, we randomly sample a collection of image patches from unlabeled training images that suffer from quality distortions at different degrees. For each patch, a corresponding local descriptor is extracted (as described in the last section). Thus, we can obtain a set of local descriptors $Y = [y_1, y_2, \ldots, y_k]$, where $Y\in R^{d\times k}$, d is the dimension of each local descriptor, and k is the number of patches. Then, we initialize a filter dictionary $X\in R^{v\times d}$ with random small values, where v is the number of filters that we aim to learn. Finally, the objective of sparse filtering can be formulated as

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \sum_{i=1}^{\Omega} \varphi|\mathbf{Z}|,$$

where $Z = XY$, Ω is the number of entries in Z and $|\cdot|$ represents the soft-absolute operation defined as

$$\mathbf{Z} = \sqrt{\varepsilon + \mathbf{Z}^2},$$

where ε is a small constant and φ stands for the operation of performing twice normalization on $|Z|$ by rows and by columns using l_2 norm, respectively.

An off-the-shelf L-BFGS[17] package can be used to optimize this objective function until convergence. Because the normalization operation introduces competition,[1] some values in $\varphi|Z|$ must be large while the others are small (close to 0) after optimization. Each row of X only gives a strong response to certain kinds of quality-aware features. This turns X into the QAF dictionary that we need. (For more implementation details and a theoretical proof of sparse filtering, please refer to earlier work.[1])

To make the filter dictionary more stable, we repeat this process u times, and after we get a filter collection of $v\times u$ filters, we perform a

K-means clustering on them to get the final filter set $C \in R^{c \times d}$, consisting of c filters.

Local Descriptor Encoding and Feature Pooling

As we described in the "Extracting Local Descriptors" section, in our approach, each image is first represented by a set of local descriptors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n] \in R^{d \times n}$, where $\mathbf{f}_i$ represents a patch-level descriptor and n is the number of sampled patches.

To obtain a quality-aware feature vector for each image, we need to use the trained filter dictionary C to encode $\mathbf{F}$. At first, we apply a soft absolute on $G = CF \in R^{c \times n}$ and in turn normalize G by rows and by columns. G has c rows and n columns, where c is the number of filters in C. Then, we apply max pooling to matrix G to get the final feature vector. Typically, there are two types of pooling strategies: average and max. In our experiment, we find that max pooling can achieve better results. Specifically, for each column $\mathbf{g}_j = (\theta_1, \theta_2, \ldots, \theta_c)^{\mathrm{T}} (j = 1, 2, \ldots, n)$ in G, the max pooling performed on $\mathbf{g}_j$ can be written as

$$\theta_i = \begin{cases} 1, & \text{if } \theta_i = \max(\theta_1, \theta_2, \theta_3, \ldots, \theta_c) \\ 0, & \text{else} \end{cases}, \quad i = 1, 2, \ldots, c.$$

Then, image level feature can be obtained by summing up all the columns, which can be expressed as

$$\mathbf{g}_* = \mathbf{g}_1 + \mathbf{g}_2 +, \ldots, + \mathbf{g}_n,$$

where $\mathbf{g}_* \in R^{c \times 1}$.

For any given image, after applying the aforementioned operations, we can obtain a quality-aware feature vector $\mathbf{g}_*$ to represent it, and the prediction of its perceptual quality will be based on $\mathbf{g}_*$.

Regression

Given a set of training images $\{I_i\}$ with quality distortions at various levels and their corresponding subjective scores $\{s_i\}$, we can treat NR-IQA as a regression problem. Specifically, at the training stage, we first obtain a set of quality-aware feature vectors $\{\mathbf{g}_{*i}\}$ from $\{I_i\}$ using the schemes we have described here. Then, $\{\mathbf{g}_{*i}\}$ and its corresponding scores $\{s_i\}$ are used to construct a regression model M. At the test stage, a feature vector t is extracted from the test image, and then we feed t into the trained regression model M to predict its quality score.

Table 1. Benchmark IQA image datasets.

Dataset	No. of reference images	No. of distorted images	No. of distortion types
LIVE	29	779	5
CSIQ	30	866	6
TID2013	25	3,000	24

For regression, most existing NR-IQA approaches use SVM for simplicity.[3–5,7] However, in subjective IQA experiments, the ultimate quality score is obtained by averaging the evaluation from different subjects. Inspired by this fact, we adopted the random forest,[2] the training and test procedures of which are similar to the subjective IQA process, as the regression model in our QAF approach. This method combines the "bagging" theory and the random selection of features. In our experiments, we found that the random forest achieved significantly better results than a SVM in our framework.

Experimental Results and Discussion

In our experiments, we attempted to validate QAF's performance in terms of its ability to predict the perceptual quality of a given image. To do so, we used three widely used IQA datasets: LIVE,[14] CSIQ,[15] and TID2013.[18] These databases provided us with reference images, their distorted versions, and associated subjective scores. Table 1 lists brief information about each dataset.

The distortion types in the LIVE database include JPEG2000, JPEG, white noise (WN), Gaussian blur (BLUR), and simulated fast fading (FF) Rayleigh channel. In the CSIQ database, the distortion types include JPEG2000, JPEG, WN, BLUR, global contrast decrements, and Gaussian pink noise. The TID2013 database consists of 24 different types of distortions at five levels.

To evaluate the performance of the IQA metrics, we adopted two correlation coefficients: the Spearman rank order correlation coefficient (SROCC), which is related to the prediction monotonicity, and the Pearson linear correlation coefficient (PLCC), which is related to the prediction linearity and can be considered a measure of prediction accuracy. A value close to 1 for SROCC or PLCC indicates a good quality prediction performance.

For comparison, we used five opinion-aware approaches—BIQI,[3] BRISQUE,[8] BLIINDS-II,[7]

Table 2. Performance evaluation on LIVE.

	80% partition ratio		50% partition ratio		10% partition ratio	
Method	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC
BIQI	0.825	0.840	0.739	0.764	0.547	0.623
BRISUQE	0.933	0.931	0.917	0.919	0.806	0.816
BLIIDNS-II	0.924	0.927	0.901	0.901	0.836	0.834
DIIVINE	0.884	0.893	0.858	0.866	0.695	0.701
CORNIA	0.940	0.944	0.933	0.934	0.893	0.894
NIQE	0.908	0.908	0.905	0.904	0.905	0.903
QAC	0.874	0.868	0.869	0.864	0.866	0.86
QAF	**0.948**	**0.953**	**0.949**	**0.949**	**0.942**	**0.945**

DIIVINE,[4] and CORNIA[9]—and two opinion-unaware approaches—NIQE[13] and QAC.[11] NIQE and QAC did not require training, so we reported their results on test images only to ensure a fair comparison across methods. The codes for these methods were provided by their authors, and we tuned the parameters to achieve the best results.

Implementation Details

The proposed framework contains a number of parameters that can be tuned and that may greatly impact the results of our approach:

- N is number of patches sampled from each image;
- M and N are width and height of the raw patch, respectively;
- S and J are the number of center frequencies and orientations, respectively;
- $u \times v$ is total number of filters trained through sparse filtering;
- c is number of QAFs (the size of the filter dictionary);
- *ntree* is number of trees constructed; and
- *mtry* is the number of variables to split on at each node.

The two most important parameters used in the random forest are *ntree* and *mtry*. In our experiment, we set $n = 10,000$, $M = N = 7$, $S = 5\left(1, 1/\sqrt{2}, 1/2, 1/2\sqrt{2}, \text{and} 1/4\right)$, $J = 4(0^\circ, 45^\circ, 90^\circ, \text{and} 135^\circ)$, $c = 10,000$, $ntree = 1,500$ and $mtry = 250$. There are no explicit constraints on the number of $u \times v$, however, empirically, it should not be smaller than 100,000.

Performance Evaluation on a Single Database

In the current literature, most NR-IQA algorithms are only evaluated on the LIVE IQA database using the experimental strategy described in earlier work.[4,7–9] For the training-based algorithms, 23 reference images along with their distorted images are randomly selected for training, and the remaining six reference images along with their degraded versions are used for testing. Such an experimental strategy mainly has some deficiencies. Specifically, the LIVE database only contains 779 distorted images, so the size of the test image set is only approximately $150(779 \times 0.2)$, which is too small to elicit definitive conclusions. However, with a large training set, over-fitting is likely to occur, so we cannot properly evaluate the generalization ability of the algorithms.

Therefore, in this article, we adopted a different experimental strategy similar to the one proposed in earlier work.[11] Specifically, we present the training-based method results under three partition settings: 80, 50, and 10 percent of the reference images are used for training and the remainders are used for testing. The partition was randomly conducted 1,000 times, and we report the median result.

Tables 2, 3, and 4 present the performance results for QAF and the other NR-IQA methods for the LIVE, CSIQ, and TID2013 datasets, respectively. The results show that QAF outperforms its counterparts under different partition ratios in the training samples. Additionally, although the performance of most existing NR-IQA methods decreases rapidly as the number of training

Table 3. Performance evaluation on CSIQ.

	80% partition ratio		50% partition ratio		10% partition ratio	
Method	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**
BIQI	0.092	0.237	0.092	0.396	0.020	0.311
BRISQUE	0.775	0.817	0.736	0.781	0.545	0.596
BLIIDNS-II	0.780	0.832	0.749	0.806	0.628	0.688
DIIVINE	0.757	0.795	0.652	0.716	0.441	0.492
CORNIA	0.714	0.781	0.678	0.754	0.638	0.732
NIQE	0.627	0.725	0.626	0.716	0.624	0.714
QAC	0.486	0.654	0.494	0.706	0.490	0.707
QAF	**0.786**	**0.847**	**0.771**	**0.815**	**0.742**	**0.755**

Table 4. Performance evaluation on TID2013.

	80% partition ratio		50% partition ratio		10% partition ratio	
Method	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**
BIQI	0.349	0.366	0.332	0.332	0.199	0.25
BRISQUE	0.573	0.651	0.563	0.645	0.513	0.587
BLIIDNS-II	0.536	0.628	0.458	0.48	0.402	0.447
DIIVINE	0.549	0.654	0.503	0.602	0.330	0.391
CORNIA	0.549	0.613	0.573	0.652	0.508	0.603
NIQE	0.317	0.426	0.317	0.42	0.313	0.398
QAC	0.390	0.495	0.390	0.489	0.372	0.435
QAF	**0.589**	**0.662**	**0.581**	**0.660**	**0.564**	**0.641**

Table 5. Evaluation results when trained on LIVE and tested on CSIQ.

Method	**SROCC**	**PLCC**
BIQI	0.619	0.695
BRISQUE	0.557	0.742
BLIIDNS-II	0.577	0.724
DIIVINE	0.596	0.697
CORNIA	0.663	0.764
NIQE	0.627	0.716
QAC	0.490	0.708
QAF	**0.710**	**0.720**

Table 6. Evaluation results when trained on LIVE and tested on TID2013.

Method	**SROCC**	**PLCC**
BIQI	0.394	0.468
BRISQUE	0.367	0.475
BLIIDNS-II	0.393	0.470
DIIVINE	0.355	0.545
CORNIA	0.429	0.575
NIQE	0.311	0.398
QAC	0.372	0.437
QAF	**0.440**	**0.593**

samples decreases, QAF seems to be robust to the number of training samples.

Cross-Database Evaluation

We also performed a more comprehensive performance evaluation by training on LIVE dataset and testing on the CSIQ and TID2013 datasets. For the five opinion-aware NR-IQA methods, the original authors provided their quality prediction models trained on the entire LIVE dataset. Thus, we directly use them for testing on CSIQ and TID2013.

Tables 5 and 6 show the experimental results. We can see that QAF performs significantly better than all the other state-of-the-art NR-IQA algorithms, indicating that QAF has a better generalization capability than the other methods.

Table 7. Performance of QAFs with different regression models.

		80% partition ratio		50% partition ratio		10% partition ratio	
Dataset	Method	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
LVIE	QAF_{SVM}	0.944	0.949	0.935	0.939	0.919	0.923
	QAF_{RF}	0.948	0.953	0.949	0.949	0.943	0.945
CSIQ	QAF_{SVM}	0.776	0.838	0.731	0.743	0.696	0.703
	QAF_{RF}	0.786	0.847	0.771	0.815	0.742	0.755
TID2013	QAF_{SVM}	0.582	0.655	0.551	0.602	0.529	0.597
	QAF_{RF}	0.589	0.662	0.581	0.66	0.564	0.641

Choosing the Random Forest as the Regression Model

To validate our decision to use the random forest as a regression model, we tested against a SVM in the context of QAF. We denote the method using SVM as a regression model as QAF_{SVM} and the method using random forest as a regression model as QAF_{RF}. The experimental protocol used here is the same as we used in the last section.

Table 7 summarizes the results. We can clearly see that the random forest performs much better than the SVM as a regression model when embedded in the QAF framework, especially when the number of training samples decreases.

Future Work

We will continue our future work on NR-IQA in two directions. First, we will try to find more powerful methods for extracting quality-ware features. Second, we will make efforts to devise more suitable regression models for the NR-IQA problem. **MM**

Acknowledgments

This work is supported by the Natural Science Foundation of China under grant 61201394, the Shanghai Pujiang Program under grants 13PJ1408700 and 13PJ1433200, the National Basic Research Program of China under grant 2013CB967101, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology) under grant 30920140122007.

References

1. J. Ngiam et al., "Sparse Filtering," *Proc. Advances in Neural Information Processing Systems*, 2011, pp. 1125–1133.
2. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, 2001, pp. 5–32.
3. A.K. Moorthy and A.C. Bovik, "A Two Step Framework for Constructing Blind Image Quality Indices," *IEEE Signal Processing Letters*, vol. 17, no. 5, 2010, pp. 513–516.
4. A.K. Moorthy and A.C. Bovik, "Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality," *IEEE Trans. Image Processing*, vol. 20, no. 12, 2011, pp. 3350–3364.
5. P. Ye and D. Doermann, "No-Reference Image Quality Assessment Using Visual Codebooks," *IEEE Trans. Image Processing*, vol. 21, no. 7, 2012, pp. 3129–3138.
6. M.A. Sadd, A.C. Bovik, and C. Charrier, "A DCT Statistics-Based Blind Image Quality Index," *IEEE Signal Processing Letters*, vol. 17, no. 6, 2010, pp. 583–586.
7. M.A. Sadd, A.C. Bovik, and C. Charrier, "Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain," *IEEE Trans. Image Processing*, vol. 21, no. 8, 2012, pp. 3339–3352.
8. A. Mittal, A.K. Moorthy, and A.C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," *IEEE Trans. Image Processing*, vol. 21, no. 12, 2012, pp. 4695–4708.
9. P. Ye et al., "Unsupervised Feature Learning Framework for No-Reference Image Quality Assessment," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2012, pp. 1098–1105.
10. H. Tang, N. Joshi, and A. Kapoor, "Learning a Blind Measure of Perceptual Image Quality," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2011, pp. 305–312.
11. W. Xue, L. Zhang, and X. Mou, "Learning without Human Scores for Blind Image Quality Assessment," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition* (CVPR), 2013, pp. 995–1002.
12. L. Zhang et al., "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Trans. Image Processing*, vol. 20, no. 8, 2011, pp. 2378–2386.

13. A. Mittal, R. Soundararajan, and A.C. Bovik, "Making a 'Completely Blind' Image Quality Analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, 2013, pp. 209–212.
14. H.R. Sheikh, M.F. Sabir, and A.C. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE Trans. Image Processing*, vol. 15, no. 11, 2006, pp. 3440–3451.
15. E.C. Larson and D.M. Chandler, "Most Apparent Distortion: Full-Reference Image Quality Assessment and the Role of Strategy," *J. Electronic Imaging*, vol. 19, no. 1, 2010, pp. 3440–3451.
16. D. Ruderman, "The Statistics of Natural Images," *Network: Computation in Neural Systems*, vol. 5, no. 4, 1994, pp. 517–548.
17. M. Schmidt, "minfunc," 2005, www.cs.ubc.ca/schmidtm/software/minFunc.html.
18. N. Ponomarenko et al., "Color Image Database TID2013: Peculiarities and Preliminary Results," *Proc. European Workshop on Visual Information Processing*, 2013, pp. 106–111.

Lin Zhang is an assistant professor in the School of Software Engineering at Tongji University, China, and in the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety at the Nanjing University of Science and Technology. His research interests include pattern recognition and computer vision. Zhang has a PhD in computing from the Hong Kong Polytechnic University. Contact him at cslinzhang@tongji.edu.cn.

Zhongyi Gu is a graduate student in the School of Software Engineering at Tongji University. His research interests include perceptual image quality assessment and machine earning. Gu has a BS from the School of Software Engineering at Tongji University. Contact him at zhongyi_gu@126.com.

Xiaoxu Liu is a research assistant in the School of Software Engineering at Tongji University. Her research interests include perceptual image quality assessment and machine earning. Liu has a BS from the School of Software Engineering at Tongji University. Contact her at xiaoxu_liu1990@163.com.

Hongyu Li is an associate professor in the School of Software Engineering at Tongji University. His research interests include computer vision, pattern recognition, and motion analysis. Li has PhDs in computer science from both Fudan University and the University of Eastern Finland. Contact him at hyli@tongji.edu.cn.

Jianwei Lu is a professor in the Advanced Institute of Translational Medicine and the School of Software Engineering at Tongji University. His research interests include vision science. Lu has a PhD from the Department of Computer Science at the University of Southern California. Contact him at jwlu33@gmail.com.