APSIPA Transactions on Signal and Information Processing, 2024, 13, e7

Original Paper

Lightweight High-Performance Blind Image Quality Assessment

Zhanxuan Mei[1*], Yun-Cheng Wang[1] and C.-C. Jay Kuo[1]

[1] *University of Southern California, USA*

ABSTRACT

Blind image quality assessment (BIQA) is a task that predicts the perceptual quality of an image without its reference. Research on BIQA attracts growing attention due to the increasing amount of user-generated images and emerging mobile applications where reference images are unavailable. The problem is challenging due to the wide range of content and mixed distortion types. Many existing BIQA methods use deep neural networks (DNNs) to achieve high performance. However, their large model sizes hinder their applicability to edge or mobile devices. To meet the need, a novel BIQA method with a small model, low computational complexity, and high performance is proposed and named "GreenBIQA" in this work. GreenBIQA includes five steps: 1) image cropping, 2) unsupervised representation generation, 3) supervised feature selection, 4) distortion-specific prediction, and 5) regression and decision ensemble. Experimental results show that the performance of GreenBIQA is comparable with that of state-of-the-art deep learning (DL) solutions while demanding a much smaller model size and significantly lower computational complexity.

Keywords: Image quality assessment, Blind image quality assessment, Green learning

*Corresponding author: Zhanxuan Mei, zhanxuan@usc.edu

Received 29 November 2023; Revised 13 March 2024
ISSN 2048-7703; DOI 10.1561/116.00000179

1 Introduction

Objective image quality assessment (IQA) can be classified into three categories: full-reference IQA (FR-IQA), reduced-reference IQA (RR-IQA), and no-reference IQA (NR-IQA). FR-IQA methods evaluate the quality of images by comparing distorted images with their reference images. Quite a few image quality metrics, such as PSNR, SSIM [39], FSIM [51], and MMF [23] have been proposed in the last two decades. RR-IQA methods (e.g., RR-SSIM [31]) utilize part of the information from reference images to evaluate the quality of underlying images. RR-IQA is more flexible than FR-IQA. NR-IQA, also called blind image quality assessment (BIQA), is needed in two scenarios. First, reference images may not be available to users (e.g., at the receiver). Second, most user-generated images do not have references. The need for BIQA grows rapidly due to the popularity of social media platforms and multi-party video conferencing.

Research on BIQA has received a lot of attention in recent years. Existing BIQA methods can be categorized into two types: conventional methods and deep-learning-based (DL-based) methods. Most conventional methods adopt a standard pipeline: a quality-aware feature extraction followed by a regressor that maps from the feature space to the quality score space. To give an example, methods based on natural scene statistics (NSS) analyze the statistical properties of distorted images and compute the distortion degree as quality-aware features. These quality-aware features can be represented by discrete wavelet transform (DWT) coefficients [28], discrete cosine transform (DCT) coefficients [33], luminance coefficients in the spatial domain [26], and so on. Codebook-based methods [41, 44, 45, 49] generate features by extracting representative codewords from distorted images. After that, a regressor is trained to project from the feature domain to the quality score domain.

Inspired by the success of deep neural networks (DNNs) in computer vision, researchers have developed DL-based methods to solve the BIQA problem. On the one hand, the DL-based methods achieve high performance because of their strong feature representation capability and efficient regression fitting. On the other hand, existing annotated IQA datasets may not have sufficient content to train large DNN models. Given that collecting large-scale annotated IQA datasets is expensive and time-consuming and that DL-based BIQA methods tend to overfit the training data from IQA datasets of limited sizes, it is critical to address the overfitting problem caused by small-scale annotated IQA datasets. Effective DL-based solutions adopt a large pre-trained model that was trained on other datasets, e.g. ImageNet [7].

The transferred prior information from a pre-trained model improves the test performance. Nevertheless, it is difficult to implement a large pre-trained model of high complexity on mobile or edge devices. As social media contents are widely accessed via mobile terminals, it is desired to conduct BIQA

with limited model sizes and computational complexity. A lightweight, high-performance BIQA solution is in great need. To address this void, we study the BIQA problem in depth and propose a new solution called "GreenBIQA". This work has the following three main contributions.

- A novel GreenBIQA method is proposed for images with synthetic and real-world (or authentic) distortions. It offers a transparent and modularized design with a feedforward training pipeline. The pipeline includes unsupervised representation generation, supervised feature selection, distortion-specific prediction, regression, and ensembles of prediction scores.
- We conduct experiments on four IQA datasets to demonstrate the prediction performance of GreenBIQA. It outperforms all conventional BIQA methods and DL-based BIQA methods without pre-trained models in prediction accuracy. Compared to state-of-the-art BIQA methods with pre-trained networks, the prediction performance of GreenBIAQ is still quite competitive yet demands a much smaller model size and significantly lower inference complexity.
- We carry out experiments under the weakly supervised learning setting to demonstrate the robust performance of GreenBIQA as the number of training samples decreases. Also, we show how to exploit active learning in selecting images for labeling.

It is worthwhile to point out that the preliminary results of our research were presented in [25]. This work is its extension. The additional content includes a more thorough literature review in Section 2, an elaborative description of the GreenBIQA method and more exemplary images to illustrate key discussed ideas in Section 3, improved and extended experimental results in Section 4. In particular, we have added new experimental results on memory/latency tradeoff, cross-domain learning, ablation study, weak-supervision, and active learning.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. GreenBIQA is described in Section 3. Experimental results are shown in Section 4. Finally, concluding remarks are given in Section 5.

2 Related Work

2.1 Conventional BIQA Methods

Conventional BIQA methods adopt a two-step processing pipeline: 1) extracting quality-aware features from input images, and 2) using a regression model to predict the quality score based on extracted features. The support

vector regressor (SVR) [1] or the XGBoost regressor [4] is often employed in the second step. According to the differences in the first step, we categorize conventional BIQA methods into two main types.

2.1.1 Natural Scene Statistics (NSS)

The first type relies on natural scene statistics (NSS). These methods predict image quality by evaluating the distortion of the NSS information. For example, DIIVINE [29] proposed a two-stage framework, including a classifier to identify different distortion types, followed by a distortion-specific quality assessment. Instead of computing distortion-specific features, NIQE [27] evaluated the quality of distorted images by computing the distance between the model statistics and those of distorted images. BRISQUE [26] used NSS to quantify the loss of naturalness caused by distortions, which is operated in the spatial domain with low complexity. BLINDS-II [33] proposed an NSS model using the discrete cosine transform (DCT) coefficients and then adopted the Bayesian inference approach to predict image quality using features extracted from the model. NBIQA [30] developed a refined NSS mode by collecting competitive features from existing NSS models in both spatial and transform domains. Histogram counting and the Weibull distribution were employed in [42] and [50], respectively, to analyze the statistical information and build the distribution models. Although the methods mentioned above utilized the NSS information in a wide variety, they are still not powerful enough to handle a broad range of distortion types, especially for datasets with authentic distortions.

2.1.2 Codebook-based Methods

The second type extracts representative codewords from distorted images. The common framework of codebook-based methods includes local feature extraction, codebook construction, feature encoding, spatial pooling, and quality regression. CBIQ [44] constructed visual codebooks from training images by quantizing features, computed the codeword histogram, and fed the histogram data to the regressor. Following the same framework, CORNIA [45] extracted image patches from unlabeled images as features, built a codebook (or a dictionary) based on clustering, converted an image into a set of non-linear features, and trained a linear support vector machine to map the encoded quality-aware features to quality scores. Non-linear features in this pipeline were obtained from the dictionary using soft-assignment coding with spatial pooling. However, the codebook needs a large number of codewords to achieve good performance. The high order statistics aggregation (HOSA) was exploited in [41] to design a codebook of a smaller size. That is, besides the mean of each cluster, the high-order statistical information (e.g., dimension-wise variance

and skewness) inside each cluster can be aggregated to reduce the codebook size. Generally speaking, codebook-based methods rely on high-dimensional handcrafted feature vectors, and they are not effective in handling diversified distortion types.

2.2 DL-based BIQA Methods

DL-based methods have been intensively studied to solve the BIQA problem. A solution based on the convolutional neural network (CNN) was first proposed in [14]. It includes one convolutional layer with max and min pooling and two fully connected layers. To alleviate the accuracy discrepancy between FR-IQA and NR-IQA, a local quality map was derived using CNN to imitate the behaviors of FR-IQA in BIECON [15]. Then, a statistical pooling strategy is adopted to capture the holistic properties and generate fixed-size feature vectors. A DNN model was proposed in WaDIQaM [2] by including ten convolutional layers as well as five pooling layers for feature extraction, and two fully connected layers for regression. MEON [24] proposed two sub-networks to achieve better performance on synthetic datasets. The first sub-network classifies the distortion types, while the second sub-network predicts the final quality. By sharing their earlier layers, the two sub-networks can solve their sub-tasks jointly for better performance.

Quality assessment of images with authentic (i.e., real-world) distortions is challenging due to mixed distortion types and high content variety. Recent DL-based methods all adopt advanced DNNs. Feature extraction using a pre-trained ResNet [11] was adopted in [46]. A probabilistic quality representation was proposed in PQR [47], which employed a more robust and optimal loss function to describe the score distribution generated by different subjects. It improved the accuracy of quality prediction and sped up the training process. A self-adaptive hyper network architecture was utilized by HyperIQA [36] to adjust the quality prediction parameters. It can handle a broad range of distortions with a local distortion-aware module and deal with wide content variety with perceptual quality patterns based on recognized content adaptively. DBCNN [54] adopted DNN models pre-trained by large datasets to facilitate quality prediction on both synthetic and authentic datasets. A network pre-trained by synthetic-distortion datasets was used to classify distortion types and levels. Another pre-trained network based on the ImageNet [7] was used as the classifier. The two feature sets from two models were integrated into one representation for final quality prediction through bilinearly pooling. The absence of the ground truth reference was compensated in Hallucinated-IQA [22], which generated a hallucinated reference using generative adversarial networks (GANs) [10].

Instead of predicting the mean opinion score (MOS) generated by subjects, NIMA [38] predicted the MOS distribution using a CNN. To balance the trade-

off between performance accuracy and number of model parameters, NIMA had three models with different architectures, namely, VGG16 [35], Inception-v2 [37], and MobileNet [13]. NIMA (VGG16) gave the best performance but with the longest inference time and the largest model size. NIMA (MobileNet) was the smallest one with the fewest model parameters but the worst accuracy. Although NIMA (MobileNet) has a small model size, it is still difficult to deploy it on mobile/edge devices.

2.3 Green Machine Learning

Green learning [17] has been proposed recently as an alternative machine learning paradigm that targets efficient models of low carbon footprint. They are characterized by small model sizes and low training and inference computational complexities. An additional advantage is its mathematical transparency through a modularized design principle. Green learning was originated by efforts in understanding the functions of various components of CNNs such as nonlinear activation [16], convolutional layers and fully-connected layers [18]. Its development path has started to deviate from neural networks by giving up the basic neuron unit and the network architecture since 2020. Examples of green learning models include PixelHop [5] and PixelHop++ [6] for object classification and PointHop [53] and PointHop++ [52] for 3D point cloud classification. Green learning techniques have been developed for many applications, such as deepfake detection [3], anomaly detection [48], image generation [20], etc. We propose a lightweight BIQA method in this work by following this path.

3 Proposed GreenBIQA Method

An overview of the proposed GreenBIQA method is depicted in Figure 1. As shown in the figure, GreenBIQA has a modularized solution that consists of five modules: (1) image cropping, (2) unsupervised representation generation, (3) supervised feature selection, (4) distortion-specific prediction, and (5) regression and decision ensemble. They are elaborated below.

3.1 Image Cropping

Image cropping is implemented to standardize the input size and enlarge the number of training samples. It is achieved by cropping sub-images of fixed size from raw images in datasets. All cropped sub-images are assigned the same mean opinion score (MOS) as their source image. MOS is a commonly used metric in the field of quality assessment. It is a numerical value that represents the average opinion of a group of human observers, who have rated

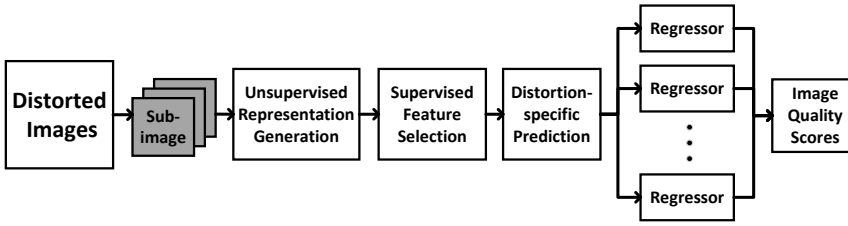


Figure 1: An overview of the proposed GreenBIQA method.

the quality of a particular image or video. In the BIQA problem, MOS serves as the label of the learning objective. To ensure the high correlation between sub-images and their assigned MOS, we adopt different cropping strategies for synthetic-distortion and authentic-distortion datasets, as shown in Figure 2 and Figure 3, respectively.

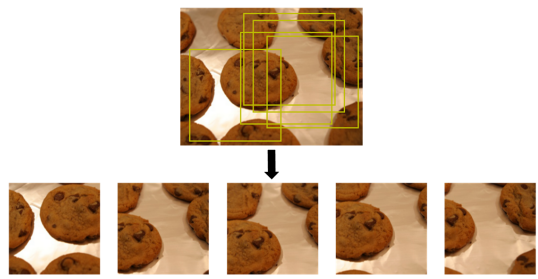


Figure 2: An exemplary image from KonIQ-10K [12] and its five cropped sub-images for authentic-distortion datasets.

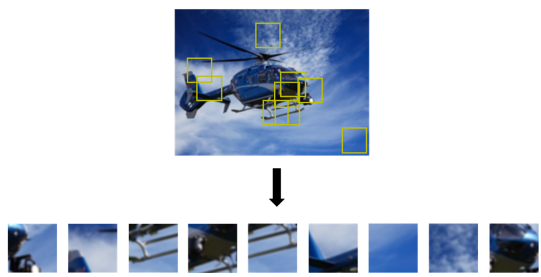


Figure 3: An exemplary image from KADID-10K [21] and its nine cropped sub-images for synthetic-distortion datasets.

For images in an authentic-distortion dataset such as KonIQ-10K [12], they contain distortions in unknown regions. Thus, we crop a smaller number of sub-images of a larger size (e.g., 256×256 out of 384×512) to ensure the assigned MOS for each sub-image is reasonable. The cropped sub-images can overlap with one another. Figure 2 shows five randomly cropped sub-images from one source image.

For images in a synthetic-distortion dataset such as KADID-10K [21], all distortions are applied to the reference images uniformly with few exceptions (e.g., color distortion in localized regions in KADID-10K). Only one distortion type is added to one image at a time. Therefore, cropping sub-images of a smaller size is sufficient to capture distortion characteristics. Furthermore, we can crop more sub-images to enlarge the number of training samples and conduct decision ensembles in the inference stage. An example of image cropping from the KADID-10K dataset is shown in Figure 3, where nine sub-images of size of 64×64 are randomly selected.

3.2 Unsupervised Representation Generation

Given sub-images from the image cropping module, we extract a set of representations from sub-images in an unsupervised manner. We consider two types of representations.

1. Spatial representations. They are extracted from the Y, U, and V channels of sub-images individually.
2. Joint spatio-color representations. They are extracted from a 3D cuboid of size $H \times W \times C$, where H and W are the height and width of a sub-image and $C = 3$ is the number of color channels, respectively.

3.2.1 Spatial Representations

Figure 4 shows the procedure of spatial representation generation. The representations are derived from 8×8 block DCT coefficients. The Discrete Cosine Transform (DCT) is a mathematical transform widely used in signal processing and image compression [40]. It converts a sequence of data points, often in one or more dimensions, into a collection of coefficients, which encapsulate data characteristics in terms of cosine functions across varying frequencies. The rationale for our utilization of DCT coefficients is rooted in its capacity to enhance overall efficiency, given that a substantial proportion of compressed images uses the DCT transform. By integrating the compression scheme and image quality assessment models, the direct extraction of DCT coefficients from cropped images proves more resourceful than employing raw image pixels as the input.

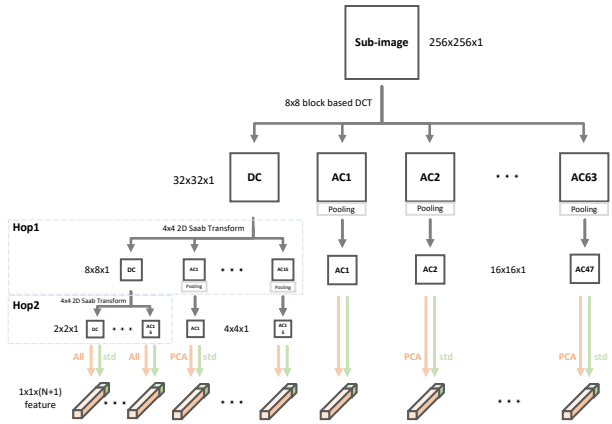


Figure 4: Unsupervised spatial representations generation.

The input sub-images are first partitioned into non-overlapping blocks of size, 8×8, and DCT coefficients are generated by the block DCT transform. DCT coefficients of each block are scanned in the zigzag order, leading to one DC coefficient and 63 AC coefficients, denoted by AC1-AC63. We split them into 64 channels. Generally, the amount of energy decreases from the DC channel to the AC63 channel. There are correlations among DC coefficients of spatially adjacent blocks. We apply the Saab transform [18] to them. The Subspace approximation with adjusted bias (Saab) transform is a variant of the principal component analysis (PCA) method, characterized by the inclusion of an added bias vector. In the Saab transform, a constant-element kernel is employed to compute the average value of image patches, commonly referred to as the DC (Direct Current) component of the Saab transform. Subsequently, PCA is applied to these patches after the removal of the computed mean, resulting in the generation of data-driven kernels known as AC (Alternating Current) kernels. The utilization of these AC kernels on individual patches leads to the derivation of AC coefficients associated with the Saab transform. In this context, for the purpose of decorrelating DC coefficients and producing elevated-level representations, a dual-stage approach employing two successive Saab transforms, referred to as Hop1 and Hop2, is employed.

- Hop1 Processing: We partition 32×32 DC coefficients into non-overlapping blocks of size 4×4 and conduct the Saab transform on each block, leading to one DC channel and 15 AC channels in Hop1. We feed the 8×8 DC coefficients to the next hop.
- HOP2 Processing: We apply another 4×4 Saab transform on each of non-overlapping blocks of size 4×4, leading to DC and 15 AC channels

in Hop2. We collect all the representations from Hop2 and append them to the final representation set to preserve low-frequency details.

Other Saab coefficients in Hop1 and other DCT coefficients at the top layer contain mid- and high-frequency information. We need to aggregate them spatially to reduce the representation number. First, we take their absolute values and apply the maximum pooling to lower their dimension as indicated by the down-ward gray arrow. Next, we adopt the following operations to yield two sets of values:

- Compute the maximum value, the mean value, and the standard deviation of the same coefficients across the spatial domain.
- Conduct the PCA transform on spatially adjacent regions for further dimension reduction (except the coefficients in Hop2).

These values are concatenated to form spatial representations of interest. The same process is applied to the Y, U, and V channels of all sub-images.

3.2.2 Joint Spatio-Color Representations

We first convert sub-images from the YUV to RGB color space. The corresponding spatio-color cuboids have a size of $H \times W \times C$, where H and W are the height and width of the sub-image, respectively, and $C = 3$ is the number of color channels. They serve as input cuboids to a two-hop hierarchical structure, as shown in Figure 5. In Hop1, we split the input cuboids into non-overlapping cuboids of size $4 \times 4 \times 3$ and apply the 3D Saab transform to them individually - leading to one DC channel and 47 AC channels, denoted by AC1-AC47. Each channel has a spatial dimension of 64×64. Since the DC coefficients are spatially correlated, we apply the 2D Saab transform in Hop2, where the DC channel of size 64×64 is decomposed into 16×16 non-overlapping blocks of size 4×4. For other 47 AC coefficients in the output of Hop1, we take their absolute values and conduct the 4x4 max pooling, leading to 47 channels of spatial dimension 16×16. In total, we obtain $16 + 47 = 63$ channels of the same spatial size 16×16. We use the following two steps to extract joint spatio-color features.

- Flatten blocks to vectors, conduct PCA, and select coefficients from the first N principal components.
- Compute the standard deviation of the coefficients in the same channel.

The above two sets of representations are concatenated to form the joint spatio-color representations.

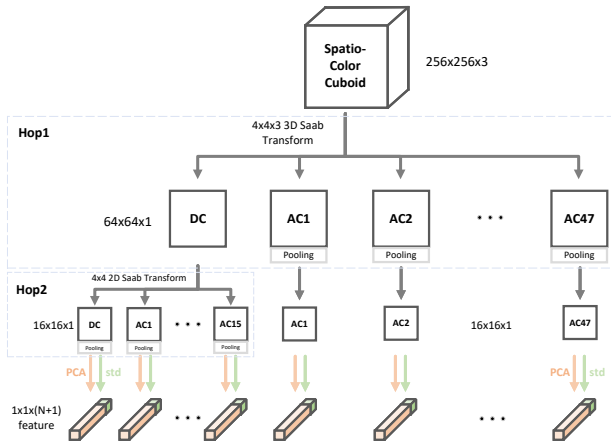


Figure 5: Unsupervised joint spatio-color representations generation.

3.3 Supervised Feature Selection

It is desired to select more discriminant features from a large number of representations obtained from the second module. A powerful tool, called the relevant feature test (RFT) [43], is adopted to achieve this objective.

RFT involves partitioning a feature dimension into two segments, left and right, and assessing the total mean-squared error (MSE) or root-MSE (RMSE) from them. The resulting approximation error serves as the RFT loss function, where a smaller RFT loss indicates a more powerful feature dimension. Given a dataset with N data samples and P features, let f_i, $1 \leq i \leq P$, represents a feature dimension with a minimum and maximum range of f^i_{min} and f^i_{max}, respectively. The three steps of deploying RFT are elaborated below.

- Training Sample Partitioning. For each feature, f^i, we aim to find the optimal threshold, f^i_{op}, within the range $[f^i_{min}, f^i_{max}]$, which allows us to partition the training samples into two subsets: S^i_L and S^i_R. If the ith feature value, x^i_n, for the nth training sample x_n is less than f^i_{op}, then x_n belongs to S^i_L; otherwise, x_n belongs to S^i_R. To narrow down the search space for f^i_{op}, we divide the entire feature range, $[f^i_{min}, f^i_{max}]$, into B uniform segments and search the optimal threshold among $B-1$ candidates.

- RFT Loss Measured by Estimated Regression MSE. Denoting the regression target value as y, let y^i_L and y^i_R represent the mean target values in S^i_L and S^i_R, respectively. These values are used as the estimated

regression values for all samples in S_L^i and S_R^i. The RFT loss is defined as the sum of estimated regression MSEs of S_L^i and S_R^i, given by

$$R_t^i = \frac{N_{L,t}^i R_{L,t}^i + N_{R,t}^i R_{R,t}^i}{N}, \tag{1}$$

where $N_{L,t}^i$, $N_{R,t}^i$, $R_{L,t}^i$, and $R_{R,t}^i$ represent the sample numbers and estimated regression MSEs in subsets S_L^i and S_R^i, respectively. Each feature f^i is characterized by its optimized estimated regression MSE over a set, T, of candidate partition points:

$$R_{op}^i = \min_{t \in T} R_t^i. \tag{2}$$

- Feature Selection based on the Optimized Loss. The optimized estimated regression MSE value, R_{op}^i, is calculated for each feature dimension, f_i. These values are then sorted in ascending order, representing the relevance of each feature dimension. The lower the R_{op}^i value, the more relevant the ith-dimensional feature, f^i.

Following this process, after computing the R_{op}^i value for each dimension of feature, f^i, we sort representation indices i, according to their RMSE values from the smallest to the largest in Figure 6. There are two curves, one for the spatial representations and the other for the spatio-color representations. We can use the elbow point on each curve to select a subset of representations. In the experiment, we use RFT to select 2048-dimensional spatial features and 2000-dimensional spatio-color features. The former is a concatenation of spatial features from Y, U, and V channels.

3.4 Distortion-specific Prediction

Enhancing prediction accuracy in image quality assessment often entails the classification or clustering of distorted images into distinct categories based on their respective distortion types. This approach recognizes the inherent difficulty of utilizing a single regressor to address the diverse range of distortion types. To mitigate this challenge, we employ a divide-and-conquer strategy, wherein various distortion types are classified into homogeneous groups. We perform this analysis for synthetic-distortion and authentic-distortion datasets independently, as illustrated in Figure 7 due to their different properties.

3.4.1 Synthetic Distortions

Images in synthetic-distortion datasets are usually associated with one specific distortion type with multiple severity levels. For example, CSIQ [19] has 6

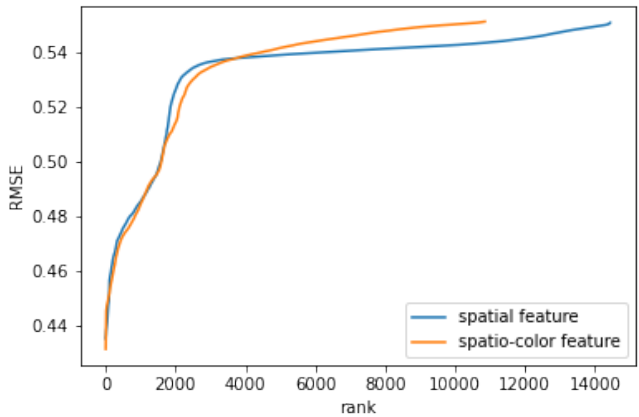


Figure 6: RFT results of spatial and spatio-color representations.

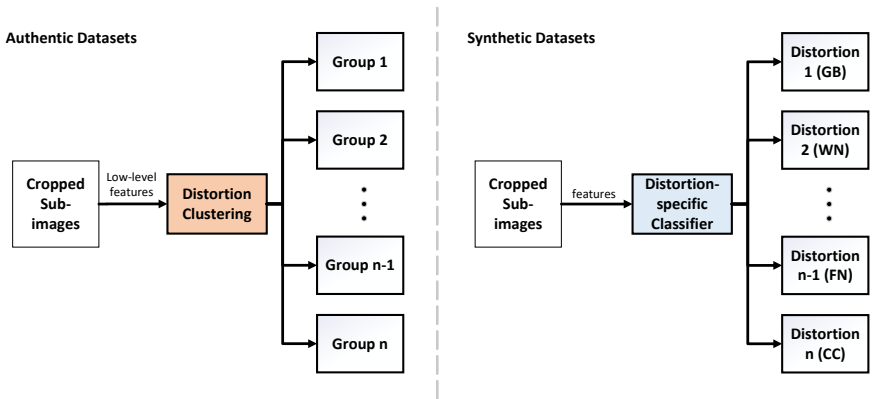


Figure 7: The diagrams of distortion-specific classifiers and distortion clustering for synthetic and authentic datasets, respectively, where GB, WN, FN, and CC denote Gaussian blur, white Gaussian noise, pink Gaussian noise, and contrast decrements, respectively.

distortion types with 4 to 5 different levels, as shown in Figure 8. We can leverage the known distortion types by first training a distortion classifier to separate images accordingly. Then, we design an individual pipeline to handle each distortion type. We can use distortion labels of training images to train a multi-class distortion classifier based on the selected features in Section 3.3. There are multiple sub-images from one image, and each of them may have a different predicted distortion type. We adopt majority voting to determine

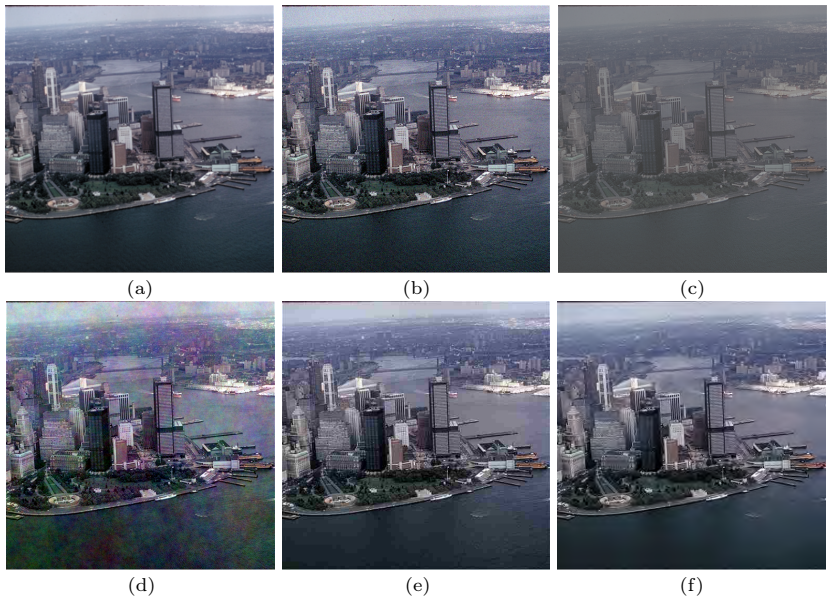


Figure 8: Six synthetic distortion types in CSIQ: (a) Gaussian blur, (b) Gaussian noise, (c) Contrast decrements, (d) Pink Gaussian noise, (e) JPEG, and (f) JPEG-2000.

the image-level distortion type. Note that some distortion types are easily confused with each other (e.g., JPEG and JPEG2000). We can simply merge them into a single type. As a result, the class number can be reduced.

3.4.2 Authentic Distortions

Images from authentic-distortion datasets may contain mixed distortion types introduced in image capture or transmission. Three distorted images from KonIQ-10K are shown in Figure 9. It is difficult to define each as one specific type. For example, the underwater image contains blurriness, noise, and color distortion. Thus, instead of training a specific distortion classifier, we cluster images into multiple groups using some low-level features in an unsupervised manner (e.g., the K-means algorithm). The low-level features include statistical information in the spatial and color domains. For spatial features, we apply the Laplacian and Sobel edge filters to all pixels in each sub-image, take their absolute values, and compute the mean, variance, and maximum. For color features, we compute the variance of each color channel (such as Y, U, and V). In addition, higher-order statistics are also collected as color features. All these extracted low-level features are concatenated into a feature vector for unsupervised clustering. Although unsupervised clustering does not assign a

(a) (b) (c)

Figure 9: Three distorted images in KonIQ-10k: (a) Dark environment, (b) Underwater, and (c) Smeared light.

distortion type to a cluster, it reduces the content diversity of sub-images in the same cluster. The rationale behind our utilization of low-level features, as opposed to the extracted features detailed in Sections 3.2 and 3.3 is to reduce the overall complexity. Our investigations reveal that low-level feature extraction is usually fast and consumes less computation power. Furthermore, these low-level features are effective enough to classify or cluster distortion types.

3.5 Regression and Decision Ensemble

For each of 6 distortions, 19 distortions, and 4 clusters for CSIQ, KADID-10K, and authentic-distortion datasets, we train an XGBoost regressor [4] that maps from the feature space to the MOS score, respectively. In the experiment, we set hyper-parameters of the XGBoost regressor to the following: 1) the max depth of each tree is 5, 2) the subsampling ratio is 0.6, 3) the maximum tree number is 2000, and 4) the early stop is adopted. Given the predicted MOS scores of all sub-image from the same source image, a median filter is applied to generate the ultimate predicted MOS score of the input image.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We evaluate GreenBIQA on two synthetic IQA datasets and two authentic IQA datasets. Their statistics are given in Table 1. The two synthetic-distortion datasets are CSIQ [19] and KADID-10K [21]. Multiple distortions of various levels are applied to a set of reference images to yield distorted images. CSIQ has six distortion types with four to five distortion levels. KADID-10K contains 25 distortion types with five levels for each distortion type. LIVE-C [8] and KonIQ-10K [12] are two authentic-distortion datasets. They contain a broad

Table 1: Four benchmarking IQA datasets, where the number of distorted images, the number of reference images, the number of distortion types and collection methods of each dataset are listed.

Datasets	Dist.	Ref.	Dist. Types	Scenario
CSIQ	866	30	6	Synthetic
KADID-10K	10,125	81	25	Synthetic
LIVE-C	1,169	N/A	N/A	Authentic
KonIQ-10K	10,073	N/A	N/A	Authentic

range of distorted real-world images captured by users. No reference image and specific distortion type are available for each image. LIVE-C and KonIQ-10K have 1,169 and 10,073 distorted images, respectively.

4.1.2 Evaluation Metrics

The performance is measured by two popular metrics: the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Order Correlation Coefficient (SROCC). PLCC evaluates the correlation between predicted scores from an objective method and user's subjective scores (e.g., MOS) in form of

$$PLCC = \frac{\sum_i (p_i - p_m)(\hat{p_i} - \hat{p_m})}{\sqrt{\sum_i (p_i - p_m)^2}\sqrt{\sum_i (\hat{p_i} - \hat{p_m})^2}}, \tag{3}$$

where p_i and $\hat{p_i}$ represent predicted and subjective scores while p_m and $\hat{p_m}$ are their means, respectively. SROCC measures the monotonicity between predicted scores from an objective method and the user's subjective scores via

$$SROCC = 1 - \frac{6\sum_{i=1}^{L}(m_i - n_i)^2}{L(L^2-1)}, \tag{4}$$

where m_i and n_i denote the ranks of the prediction and the ground truth label, respectively, and L denotes the total number of samples or the number of images in our current case.

4.1.3 Implementation Details

In the training stage, we crop 15 sub-images of size 224×224 for each image in the two authentic datasets. This design choice is based on the SROCC performance of validation sets, as shown in Figure 10, where the best performance under different crop sizes is highlighted. Similarly, we crop 25 sub-images of size 32×32 for each image in the two synthetic datasets. In the testing (or inference) stage, we crop 25 sub-images of size 224×224 and 32×32 for images in authentic and synthetic datasets, respectively.

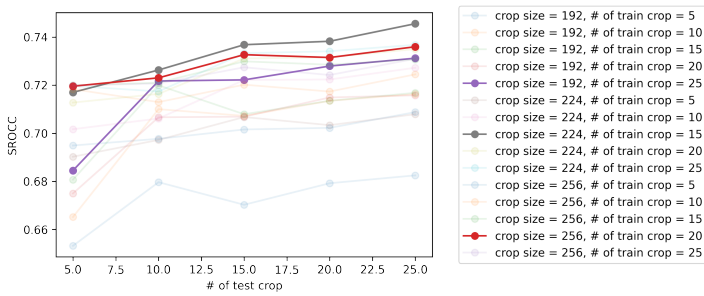


Figure 10: Performance curve on the validation dataset of LIVE-C with different crop numbers and sizes.

We adopt the standard evaluation procedure by splitting each dataset into 80% for training and 20% for testing. Furthermore, 10% of training data is used for validation. We run experiments 10 times and report median PLCC and SROCC values. For synthetic-distortion datasets, splitting is implemented on reference images to avoid content overlap.

4.2 Experimental Results

4.2.1 Benchmarking Methods

We compare the performance of GreenBIQA with eleven benchmarking methods in Table 2. They include four conventional and seven DL-based BIQA methods. We divide them into four categories.

- NIQE [27] and BRISQUE [26]. They are conventional BIQA methods using NSS features.
- CORNIA [45] and HOSA [41]. They are conventional BIQA methods using codebooks.
- BIECON [15] and WaDIQaM [2]. They are DL-based BIQA methods without pre-trained models (or simple DL methods).
- PQR [47], DBCNN [54], HyperIQA [36], TReS [9], and QPT [55]. They are DL-based BIQA methods with pre-trained models (or advanced DL methods).

4.2.2 Comparison Among Benchmarking Methods

We first compare the performance among the eleven benchmarks. Although some conventional BIAQ methods have comparable performance with simple

Table 2: Performance comparison in PLCC and SROCC metrics between our GreenBIQA method and eleven benchmarking methods on four IQA databases, where the eleven benchmarking methods are categorized into four groups as discussed in Section 4.2.1 and the best performance numbers are shown in boldface.

	CSIQ		LIVE-C		KADID-10K		KonIQ-10K		
Model	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	Model size (MB)
NIQE [27]	0.627	0.712	0.455	0.483	0.374	0.428	0.531	0.538	-
BRISQUE [26]	0.746	0.829	0.608	0.629	0.528	0.567	0.665	0.681	-
CORNIA [45]	0.678	0.776	0.632	0.661	0.516	0.558	0.780	0.795	7.4
HOSA [41]	0.741	0.823	0.661	0.675	0.618	0.653	0.805	0.813	0.23
BIECON [15]	0.815	0.823	0.595	0.613	-	-	0.618	0.651	35.2
WaDIQaM [2]	0.844	0.852	0.671	0.680	-	-	0.797	0.805	25.2
PQR [47]	0.872	0.901	0.857	0.882	-	-	0.880	0.884	235.9
DBCNN [54]	0.946	**0.959**	0.851	0.869	0.851	0.856	0.875	0.884	54.6
HyperIQA [36]	0.923	0.942	0.859	0.882	0.852	0.845	0.906	0.917	104.7
TReS [9]	0.922	0.942	0.846	0.877	0.859	0.858	0.915	0.928	582
QPT-ResNet50 [55]	-	-	**0.894**	**0.914**	-	-	**0.927**	**0.941**	-
GreenBIQA (Ours)	**0.952**	**0.959**	0.801	0.809	**0.886**	**0.893**	0.858	0.870	1.82

DL methods (without pre-trained models), we see a clear performance gap between conventional BIQA methods and advanced DL methods (with pre-trained models). On the other hand, the model size of advanced DL methods is significantly larger. We comment on the performance of GreenBIQA against other benchmarking methods, as shown below.

4.2.3 Synthetic-Distortion Datasets

For the two synthetic-distortion datasets, CSIQ and KADID-10K, GreenBIQA achieves the best performance among all. This is attributed to its two characteristics: 1) classification of synthetic distortions to multiple types followed by different processing pipelines, and 2) effective usage of ensemble decisions. For the first point, there are six distortion types in CSIQ, as shown in Figure 8. We show the SROCC performance of the best BIQA method in each of the four categories against each of the six distortion types in the CSIQ dataset in Table 3. GreenBIAQ outperforms all others in four distortion types. It performs especially well for JPEG distortion because it adopts the DCT spatial features, which match the underlying compression distortion well. GreenBIQA is also effective against white Gaussian noise (WN), pink Gaussian noise (FN), and contrast decrements (CC) through the use of joint spatial and spatio-color features. GreenBIQA still works well for Gaussian blur (GB), although no blur detector is employed. For the second point, since the number of reference images is limited and the distortion is uniformly spread out across the whole image, ensemble decision works well in such a setting.

Table 3: Comparison of the SROCC performance for each of six individual distortion types in the CSIQ dataset, where WN, JPEG, JP2K, FN, GB, and CC denote white Gaussian noise, JPEG compression, JPEG-2000 compression, pink Gaussian noise, Gaussian blur, and contrast decrements, respectively. The last column shows the weighted average of the SROCC metrics.

	WN	JPEG	JP2K	FN	GB	CC	Average
BRISQUE	0.723	0.806	0.840	0.378	0.820	0.804	0.728
HOSA	0.604	0.733	0.818	0.500	0.841	0.716	0.702
BIECON	0.902	0.942	0.954	0.884	**0.946**	0.523	0.858
HyperIQA	0.927	0.934	0.960	0.931	0.915	**0.874**	0.923
GreenBIQA (Ours)	**0.943**	**0.980**	**0.969**	**0.965**	0.894	0.857	**0.934**

4.2.4 Authentic-Distortion Datasets

For the two authentic-distortion datasets, LIVE-C and KonIQ-10K, GreenBIQA outperforms conventional BIQA methods and simple DL methods. This demonstrates the effectiveness of its extracted quality-aware features and decision pipeline in handling diversified distortions and contents. There is, however, a performance gap between GreenBIQA and advanced DL methods with pre-trained models. The authentic-distortion datasets are more challenging because of non-uniform distortions across images and a wide variety of content without duplication. Since pre-trained models are trained by a much larger image database, they have advantages in extracting features for non-uniform distortions and unseen contents. Yet, they demand much larger model sizes as a tradeoff.

4.3 Cross-Domain Learning

To evaluate the cross-domain generalizability of BIQA methods, we train models on one dataset and test them on another dataset. Due to the huge differences in synthetic-distortion and authentic-distortion datasets, we focus on authentic-distortion datasets and conduct experiments on LIVE-C and KonIQ-10K only. We consider two experimental settings: I) trained with LIVE-C and tested on KonIQ-10K, and II) trained with KonIQ-10K and tested on LIVE-C. The SROCC performance of GreenBIQA and five benchmarking methods under the two settings are compared in Table 4, where benchmarks include the five best BIQA methods in Table 2 (e.g., PQR, DBCNN, HyperIQA, TReS, and QPT) and two conventional BIQA methods (e.g., BRISQUE and HOSA).

By comparing the performance numbers in Tables 2 and 4, we see a performance drop in the cross-domain condition for all methods. We see that GreenBIQA has a performance gap of 0.019 against the best one, HyperIQA, for Experimental Setting I. GreenBIQA has a performance gap of 0.089 against

Table 4: Comparison of the SROCC performance under the cross-domain learning scenario.

Settings	I	II
Train Dataset	LIVE-C	KonIQ-10K
Test Dataset	KonIQ-10K	LIVE-C
BRISQUE	0.425	0.526
HOSA	0.651	0.648
PQR	0.757	0.770
DBCNN	0.754	0.755
HyperIQA	**0.772**	0.785
TReS	0.733	0.786
QPT-ResNet50	0.749	**0.821**
GreenBIQA(Ours)	0.753	0.732

the best one, QPT, for Experimental Setting II. As shown in Table 1, KonIQ-10K is much larger than LIVE-C. Experimental Setting I provides a more proper environment to demonstrate the robustness (or generalizability) of a learning model. We compare the performance gaps in Table 4 under Setting I with those in the KonIQ-10K/SROCC column in Table 2. The gaps between PQR, DBCNN, HyperIQA, TReS, QPT, and GreenBIQA narrow down from 0.022, 0.017, 0.048, 0.057, and 0.069 to 0.004, 0.001, 0.019, -0.02, and -0.004, respectively. We see a greater potential for GreenBIQA in this direction.

4.4 Model Complexity

A lightweight model is critical to applications on mobile and edge devices. We analyze the model complexity of BIQA methods in four aspects below: model sizes, inference time, computational complexity in terms of floating-point operations (FLOPs), and memory/latency tradeoff.

4.4.1 Model Size

There are two ways to measure the size of a learning model: 1) the number of model parameters, and 2) the actual memory usage. Floating-point and integer model parameters are typically represented by 4 bytes and 2 bytes, respectively. Since a great majority of model parameters are in floating point, the actual memory usage is roughly equal to $4 \times$ (no. of model parameters) bytes (see Table 5). To avoid confusion, we use the "model size" to refer to actual memory usage below.

Figure 11 shows the time and memory analysis on a synthetic-distortion dataset, CSIQ, and an authentic-distortion dataset, KonIQ-10K, for nine BIQA methods. The vertical axis represents the model performance in SROCC. The

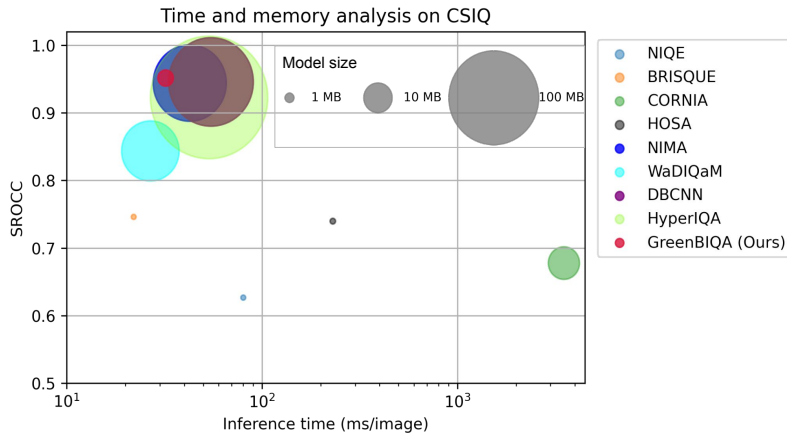
Table 5: Comparison of SROCC/PLCC performance, no. of model parameters, model sizes (memory usage), no. of GigaFlops, and no. of KiloFlops per pixel of several BIQA methods tested on the LIVE-C dataset, where "X" denotes the multiple no.

Model	SROCC	PLCC	Model Parameters (M)	Model Size (MB)	GFLOPs	KFLOPs/pixel
NIMA(Inception-v2)	0.637	0.698	10.16 (22.6X)	37.4 (20.5X)	4.37 (128.5X)	87.10 (128.5X)
BIECON	0.595	0.613	7.03 (15.6X)	35.2 (19.3X)	0.088 (2.6X)	85.94 (126.8X)
WaDIQaM	0.671	0.680	5.2 (11.6X)	25.2 (13.8X)	0.137 (4X)	133.82 (197.4X)
DBCNN	0.851	0.869	14.6 (32.4X)	54.6 (30X)	16.5 (485.3)	328.84 (485.1X)
HyperIQA	**0.859**	**0.882**	28.3 (62.9X)	104.7 (57.5X)	12.8 (376.5X)	255.10 (376.3X)
GreenBIQA (Ours)	0.801	0.809	**0.45**(1X)	**1.82**(1X)	**0.034** (1X)	**0.678**(1X)

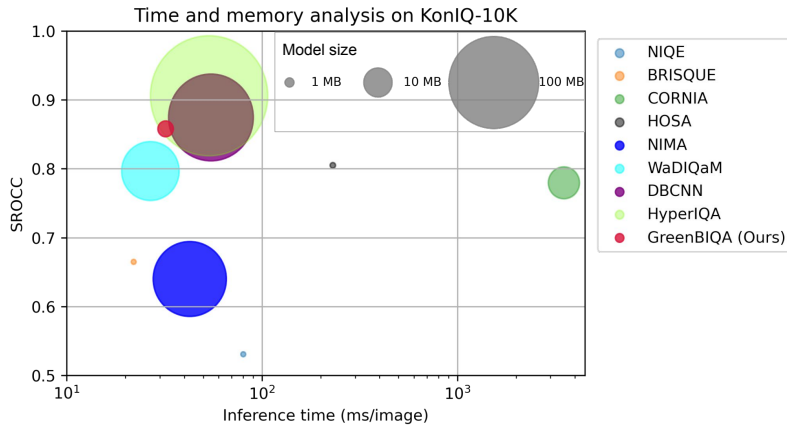
horizontal axis represents the time efficiency of the methods in milliseconds. The maker sizes are proportional to model sizes. The size of the GreenBIQA model includes the feature extractor (600KB), the distortion-specific classifier (50KB), and regressors (1.17MB), leading to a total of 1.82 MB. As compared with the two conventional methods (CORINA and HOSA), GreenBIQA achieves much better performance with comparable model sizes. GreenBIQA outperforms two simple DL methods (BIECON and WaDIQaM), with a smaller model size. As compared with four DL methods, e.g., NIMA, WaDIQaM, DBCNN, and HyperIQA, GreenBIQA achieves the best performance on CSIQ and competitive performance on KonIQ-10k at a significantly smaller model size. Note that advanced DL methods have a huge pre-trained network of size larger than 100MB as their backbones.

4.4.2 Inference Time

Another important factor to consider is running time in inference, which is especially the case for mobile/edge clients. Figure 11 shows the SROCC performance versus the inference time (measured in milliseconds per image) for several benchmarking methods on CSIQ and KonIQ-10K. All methods are tested in the same environment with a single CPU. We compare GreenBIQA with four conventional methods, e.g., NIQE, BRISQUE, CORNIA, and HOSA, and four DL methods, e.g. NIMA, WaDIQaQ, DBCNN, and HyperIQA. GreenBIQA has clear advantages over all benchmarking methods by jointly considering performance and inference time. It is worthwhile to point out that GreenBIQA can process around 31 images per second with a single CPU. In other words, it can meet the real-time requirement by processing videos of 30 fps on a frame-by-frame basis. The inference time of GreenBIQA can be further reduced by code optimization and/or with the support of mature packages.



(a) The SROCC performance on CSIQ dataset.



(b) The SROCC performance on KonIQ-10K dataset.

Figure 11: Illustration of the tradeoff between SROCC (vertical axis), inference time (horizontal axis), and model size (area of the dot) on (a) CSIQ and (b) KonIQ-10K datasets among several BIQA methods.

4.4.3 Computational Complexity

We compare the SROCC and PLCC performance, the numbers of model parameters, model sizes (in terms of memory usage), the numbers of Flops, and Flops per pixel of several BIQA methods tested on the LIVE-C dataset in Table 5. FLOPs is a common metric to measure the computational complexity of a model. For a given hardware configuration, the number of FLOPs is

linearly proportional to energy consumption or carbon footprint. Column "GFLOPs" in Table 5 gives the number of GFLOPs needed to run a model once without considering the patch number and size used in a method. For a fair comparison of FLOPs, we compute the number of FLOPS per pixel defined by

$$FLOPs/pixel = \frac{FLOPs/patch}{H \times W}, \tag{5}$$

where H and W are the height and width of an input patch to a model, respectively. NIMA with the pre-trained Inception-v2 network has low performance, large model size, and high complexity. Although simple DL methods (e.g., WaDIQaM and BIECON) use smaller networks with lower FLOPs, their performance is still inferior to GreenBIQA. Finally, advanced DL methods (e.g., DBCNN and HyperIQA) outperform GreenBIQA in SROCC and PLCC performance. However, their model sizes are much larger and their computational complexities are much higher. The numbers of FLOPs of DBCNN and HyperIQA are 485 and 376 multiples of that of GreenBIQA, respectively.

It is important to emphasize that GreenBIQA, as a non-DL-based method, will benefit less from GPU than DL-based methods at this time due to the lack of hardware-software integration for non-DL-based methods. On the other hand, the extremely low complexity in the computation of GrenBIQA, as shown in Table 5, suggests its potential in GPU-supported environments, with further advancements in third-party libraries and coding optimizations.

4.4.4 Memory/Latency Tradeoff

There is a tradeoff between memory usage and latency in the image quality inference stage. That is, latency can be reduced when given more computing resources. To observe the tradeoff, we control the memory usage using different test image numbers in each run (i.e. the batch size). Figure 12 shows the latency (in linear scale along the vertical axis) and memory usage (in log scale along the horizontal axis) of GreenBIQA and three advanced DL methods, where we set the batch size equal to 1, 4, 16, and 64 in four experiments. We see from the figure that the latency of GreenBIQA is much smaller than NIMA, DBCNN, and HyperIQA under the same memory size (say, 10^3MB). Along this line, the memory requirement of GreenBIQA is much lower than that of NIMA, DBCNN, and HyperIQA at the same level of latency. Again, the memory/latency tradeoff curve of GreenBIQA can be further improved through code optimization.

4.5 Ablation Study

To understand the impact of individual components on the overall performance of GreenBIQA, we conduct an ablation study in Table 6, where S-features, SC-

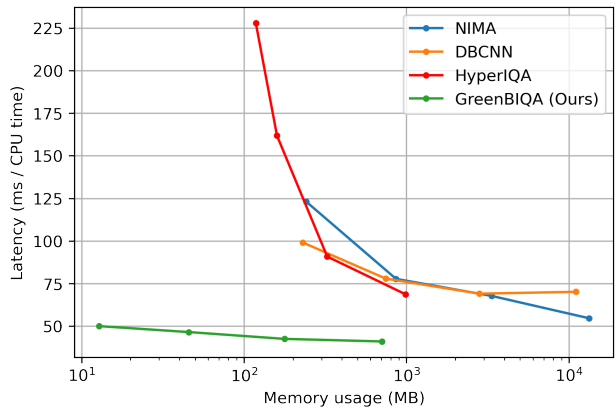


Figure 12: Tradeoff between memory usage and latency for four BIQA methods: 1) NIMA, 2) DBCNN, 3) HyperIQA, and 4) GreenBIQA.

Table 6: Ablation Study for GreenBIQA.

	CSIQ		LIVE-C		KADID-1K		KonIQ-10k	
Components	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC
S-features	0.925	0.936	0.774	0.778	0.847	0.848	0.822	0.838
S-features + SC-features	-	-	0.782	0.783	-	-	0.835	0.850
S-features + Dist-predict	0.952	0.959	0.786	0.788	0.886	0.893	0.839	0.856
S-features + SC-features + Dist-predict	-	-	0.801	0.809	-	-	0.858	0.870

features, and Dist-predict denotes spatial features, spatio-color features, and distortion-specific prediction, respectively. We first examine the effectiveness of the spatial features and then add spatio-color features in the first two rows. Both SROCC and PLCC improve on the two authentic-distortion datasets. Similarly, adding distortion-specific prediction to S-features can improve SROCC and PLCC for all datasets in the third row. Finally, we use all the components in the fourth row and see that SROCC and PLCC can be further improved to reach the highest value. Note that we do not report the performance of joint spatial and spatio-color features for synthetic datasets since spatial features are powerful enough. The distortion-specific prediction benefits the performance significantly on synthetic datasets by leveraging the distortion label.

4.6 Weak Supervision

We train BIQA models using different percentages of the KonIQ-10K training dataset (e.g., from 1% to 90%) as shown in Figure 13 and show the PLCC performance against the full test dataset. For a fair comparison, we only compare GreenBIQA with WaDIQaM, which is a simple DL method. Note that we do

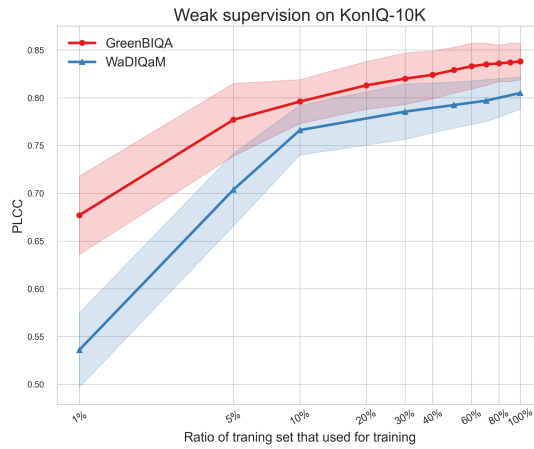


Figure 13: The PLCC performance curves of GreenBIQA and WaDIQaM are plotted as functions of the percentages of the full training dataset of KonIQ-10K, where the solid line and the banded structure indicate the mean value and the range of mean plus/minus one standard deviation, respectively.

not choose advanced DL methods with pre-trained networks for performance benchmarking since pre-trained networks have been trained by other larger datasets. We show the mean and the plus/minus one standard deviation. We see that GreenBIQA performs robustly under the weak supervision setting. Even if it is only trained on 1% of training samples, GreenBIQA can achieve a PLCC value higher than 0.67. Conversely, WaDIQaM does not perform well when the percentage goes low since a small number of samples is not sufficient in the training of a large neural network.

4.7 Active Learning

To further investigate the potential of GreenBIQA, we implement an active learning scheme [32, 34] below.

1. Keep the initial training set as 10% of the full training dataset and obtain an initial model denoted by M_1.

2. Predict the performance of remaining samples in the training dataset using M_i, $i = 1, 2, \cdots, 8$. Compute the standard derivation of predicted scores of all sub-images associated with the same image, which indicates prediction uncertainty.

3. Select a set of images that have the highest standard derivations in Step 2, where its size is 10% of the full training dataset. Merge them into the current training image set; namely, their ground truth labels are leveraged to train Model M_{i+1}.

We repeat the above process in sequence to obtain models $M_1, \cdots, M_9$. Model M_{10} is the same as the one that uses all training samples. We compare the PLCC performance of GreenBIQA with active learning and with random sampling in Figure 14. We see that the active learning strategy can improve the performance of the random selection scheme in the range from 20% to 70% of full training samples.

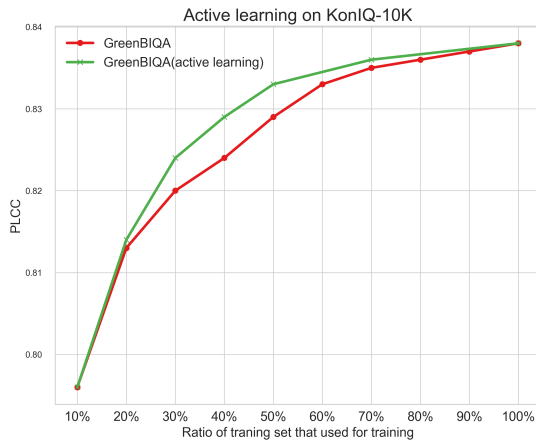


Figure 14: Comparison of the PLCC performance of GreenBIQA using active learning (in green) and random selection (in red) on the KonIQ-10k dataset.

5 Conclusion and Future Work

A novel and lightweight blind image quality assessment method, called GreenBIQA, was presented in this paper. Its performance is quantified using PLCC and SROCC on two synthetic-distortion datasets and two authentic-distortion datasets. GreenBIQA demonstrates superior performance compared to all conventional (non-DL-based) BIQA methods, as well as basic deep learning-based (DL-based) BIQA methods, across all four datasets. As compared to SOTA methods (average performance of five DL-based methods with pre-trained models shown in Table 2), GreenBIQA achieves 1.02× performance in synthetic datasets and 0.93× performance in authentic datasets with 133× smaller

model size. It can predict accurate visual quality for images in real-time, i.e., 31 images per second, using only CPUs. These characteristics position GreenBIQA as a highly suitable choice for BIQA tasks, particularly in the context of resource-constrained mobile and edge devices.

There are several research topics worth further investigation. First, we illustrate the performance of GreenBIQA through the assessment of three exemplary images, each paired with a GreenBIQA predicted MOS value and corresponding ground truth (refer to Figure 15). The left two images demonstrate accurate prediction scores, while the remaining one exhibits a suboptimal prediction. Specifically, GreenBIQA tends to underestimate the MOS value for images with blurred backgrounds, as exemplified by the longhorn deer image. To further improve the performance of GreenBIQA, there is a need to develop a cost-effective mechanism for identifying attention regions and/or a method to assess prediction confidence effectively. Second, there is a critical need to extend the capabilities of GreenBIQA to encompass lightweight yet high-performance blind video quality assessment. Achieving this involves the incorporation of temporal information, which is essential for assessing video quality accurately.

MOS(G) = 3.63	MOS(G) = 3.72	MOS(G) = 4.21
MOS(P) = 3.65	MOS(P) = 3.81	MOS(P) = 3.75

Figure 15: Comparison of the ground truth MOS and GreenBIQA-predicted MOS values of three exemplary images, which are denoted as MOS(G) and MOS(P), respectively.

References

[1] M. Awad and R. Khanna, "Support vector regression", in *Efficient learning machines*, Springer, 2015, 67–80.

[2] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment", *IEEE Transactions on image processing*, 27(1), 2017, 206–19.

[3] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector", in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[4] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, 785–94.

[5] Y. Chen and C.-C. J. Kuo, "Pixelhop: A successive subspace learning (SSL) method for object recognition", *Journal of Visual Communication and Image Representation*, 2020, 102749.

[6] Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, "Pixelhop++: A small successive-subspace-learning-based (SSL-based) model for image classification", in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3294–8.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, 248–55.

[8] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality", *IEEE Transactions on Image Processing*, 25(1), 2015, 372–87.

[9] S. A. Golestaneh, S. Dadsetan, and K. M. Kitani, "No-reference image quality assessment via transformers, relative ranking, and self-consistency", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, 1220–30.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks", *Communications of the ACM*, 63(11), 2020, 139–44.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770–8.

[12] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe, "KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment", *IEEE Transactions on Image Processing*, 29, 2020, 4041–56.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint arXiv:1704.04861*, 2017.

[14] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, 1733–40.

[15] J. Kim and S. Lee, "Fully deep blind image quality predictor", *IEEE Journal of selected topics in signal processing*, 11(1), 2016, 206–20.

[16] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model", *Journal of Visual Communication and Image Representation*, 41, 2016, 406–13.

[17] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook", *Journal of Visual Communication and Image Representation*, 90, 2023, 103685.

[18] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design", *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.

[19] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy", *Journal of electronic imaging*, 19(1), 2010, 011006.

[20] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "TGHop: an explainable, efficient, and lightweight method for texture generation", *APSIPA Transactions on Signal and Information Processing*, 10, 2021, e17.

[21] H. Lin, V. Hosu, and D. Saupe, "KADID-10k: A large-scale artificially distorted IQA database", in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2019, 1–3.

[22] K.-Y. Lin and G. Wang, "Hallucinated-IQA: No-reference image quality assessment via adversarial learning", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 732–41.

[23] T.-J. Liu, W. Lin, and C.-C. J. Kuo, "Image quality assessment using multi-method fusion", *IEEE Transactions on image processing*, 22(5), 2012, 1793–807.

[24] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-to-end blind image quality assessment using deep neural networks", *IEEE Transactions on Image Processing*, 27(3), 2017, 1202–13.

[25] Z. Mei, Y.-C. Wang, X. He, and C.-C. J. Kuo, "GreenBIQA: A Lightweight Blind Image Quality Assessment Method", in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2022, 1–6.

[26] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain", *IEEE Transactions on image processing*, 21(12), 2012, 4695–708.

[27] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a completely blind image quality analyzer", *IEEE Signal processing letters*, 20(3), 2012, 209–12.

[28] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices", *IEEE Signal processing letters*, 17(5), 2010, 513–6.

[29] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality", *IEEE transactions on Image Processing*, 20(12), 2011, 3350–64.

[30] F.-Z. Ou, Y.-G. Wang, and G. Zhu, "A novel blind image quality assessment method based on refined natural scene statistics", in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, 1004–8.

[31] A. Rehman and Z. Wang, "Reduced-reference image quality assessment by structural similarity estimation", *IEEE transactions on image processing*, 21(8), 2012, 3378–89.

[32] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C. J. Kuo, "Low-resolution face recognition in resource-constrained environments", *Pattern Recognition Letters*, 149, 2021, 193–9.

[33] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the DCT domain", *IEEE transactions on Image Processing*, 21(8), 2012, 3339–52.

[34] B. Settles, "Active learning literature survey", 2009.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[36] S. Su, Q. Yan, Y. Zhu, C. Zhang, X. Ge, J. Sun, and Y. Zhang, "Blindly assess image quality in the wild guided by a self-adaptive hyper network", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 3667–76.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 2818–26.

[38] H. Talebi and P. Milanfar, "NIMA: Neural image assessment", *IEEE transactions on image processing*, 27(8), 2018, 3998–4011.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE transactions on image processing*, 13(4), 2004, 600–12.

[40] A. B. Watson *et al.*, "Image compression using the discrete cosine transform", *Mathematica journal*, 4(1), 1994, 81.

[41] J. Xu, P. Ye, Q. Li, H. Du, Y. Liu, and D. Doermann, "Blind image quality assessment based on high order statistics aggregation", *IEEE Transactions on Image Processing*, 25(9), 2016, 4444–57.

[42] W. Xue, X. Mou, L. Zhang, A. C. Bovik, and X. Feng, "Blind image quality assessment using joint statistics of gradient magnitude and Laplacian features", *IEEE Transactions on Image Processing*, 23(11), 2014, 4850–62.

[43] Y. Yang, W. Wang, H. Fu, C.-C. J. Kuo, *et al.*, "On supervised feature selection from high dimensional feature spaces", *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[44] P. Ye and D. Doermann, "No-reference image quality assessment using visual codebooks", *IEEE Transactions on Image Processing*, 21(7), 2012, 3129–38.

[45] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment", in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, 1098–105.

[46] H. Zeng, L. Zhang, and A. C. Bovik, "A probabilistic quality representation approach to deep blind image quality prediction", *arXiv preprint arXiv:1708.08190*, 2017.

[47] H. Zeng, L. Zhang, and A. C. Bovik, "Blind image quality assessment with a probabilistic quality representation", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, 609–13.

[48] K. Zhang, B. Wang, W. Wang, F. Sohrab, M. Gabbouj, and C.-C. J. Kuo, "Anomalyhop: an ssl-based image anomaly localization method", in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2021, 1–5.

[49] L. Zhang, Z. Gu, X. Liu, H. Li, and J. Lu, "Training quality-aware filters for no-reference image quality assessment", *IEEE MultiMedia*, 21(4), 2014, 67–75.

[50] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator", *IEEE Transactions on Image Processing*, 24(8), 2015, 2579–91.

[51] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment", *IEEE transactions on Image Processing*, 20(8), 2011, 2378–86.

[52] M. Zhang, Y. Wang, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop++: A Lightweight Learning Model on Point Sets for 3D Classification", *arXiv preprint arXiv:2002.03281*, 2020.

[53] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop: An Explainable Machine Learning Method for Point Cloud Classification", *IEEE Transactions on Multimedia*, 2020.

[54] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network", *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1), 2018, 36–47.

[55] K. Zhao, K. Yuan, M. Sun, M. Li, and X. Wen, "Quality-aware pre-trained models for blind image quality assessment", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 22302–13.