Blind Image Quality Assessment Based on Geometric Order Learning

Nyeong-Ho Shin
Korea University
nhshin@mcl.korea.ac.kr

Seon-Ho Lee
Korea University
seonholee@mcl.korea.ac.kr

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

Abstract

A novel approach to blind image quality assessment, called quality comparison network (QCN), is proposed in this paper, which sorts the feature vectors of input images according to their quality scores in an embedding space. QCN employs comparison transformers (CTs) and score pivots, which act as the centroids of feature vectors of similar-quality images. Each CT updates the score pivots and the feature vectors of input images based on their ordered correlation. To this end, we adopt four loss functions. Then, we estimate the quality score of a test image by searching the nearest score pivot to its feature vector in the embedding space. Extensive experiments show that the proposed QCN algorithm yields excellent image quality assessment performances on various datasets. Furthermore, QCN achieves great performances in cross-dataset evaluation, demonstrating its superb generalization capability. The source codes are available at https://github.com/nhshin-mcl8/QCN.

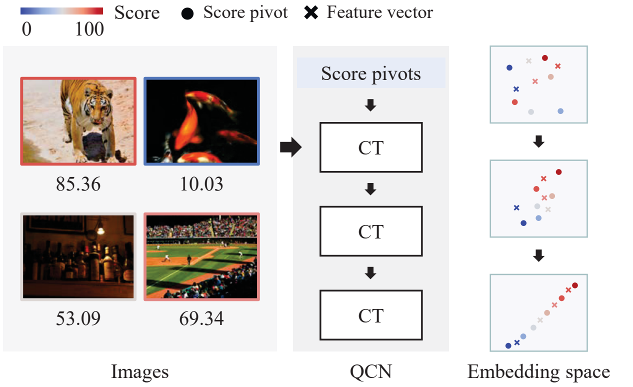


Figure 1. Illustration of the proposed QCN algorithm. The color of an image boundary represents the quality score of the image. The CT modules sequentially updates the feature vectors of multiple images. To guide this update process, we use score pivots, which act as the centroids of the feature vectors of similar-quality images. As the update goes on in the CT modules, the images and the score pivots are sorted according to their quality scores in the embedding space. We estimate the quality score of a test image by finding the nearest score pivot to its feature vector.

1. Introduction

Image quality assessment (IQA) aims to estimate the human perceptual quality of an image. It can be divided into two categories: full-reference IQA and blind IQA (BIQA). In full-reference IQA, we estimate the quality of an image by comparing it with its pristine version, referred to as the reference image. On the other hand, in BIQA, we do not use the reference image. In real-world problems, generally, references are unavailable. Hence, the demand for BIQA has increased in various applications, including image restoration [17], compression [16], and super-resolution [21].

Recently, many deep learning techniques have been developed for BIQA, achieving promising performances. Some of them focus on the structural aspect of a deep network for regressing the quality score of an image [11, 30, 37], while others explore the data aspect of deep learning and attempt to pre-train networks using a large amount of data specialized for BIQA [22, 27, 39]. These techniques [11, 22, 27, 30, 37, 39], however, do not explicitly use score relations between images, such as ordering relationship and score difference. Such relations can provide useful cues for the score estimation. Thus, Golestaneh *et al*. [7] and Zhang *et al*. [38] exploit the relative rank information between images to train a network, but they use the score relations in the training phase only.

A novel approach to BIQA is proposed in this paper, which assesses image quality reliably by exploiting both ordering relationships and score differences between images. To this end, we construct an embedding space, in which the direction and distance between the embedded vectors of two images represent the ordering relationship and score difference between the two images, respectively. The basic concept of this geometric representation learning, called geometric order learning (GOL) [14], has been proposed recently for rank estimation tasks, including facial age estimation and historical image classification. However, GOL may provide poor results for BIQA since it is designed for discrete rank estimation. In contrast, in BIQA, we should estimate the continuous quality score of an image.

In this paper, we first develop the GOL-based algorithm for BIQA, called quality comparison network (QCN), which arranges input images according to their quality scores in the embedding space, as illustrated in Figure 1. The proposed QCN employs multiple comparison transformers (CTs) and score pivots, which act as the centroids of feature vectors of similar-quality images. In each CT, the score pivots and the feature vectors of input images are updated according to their ordered correlation. To train CTs to achieve this goal, we adopt four loss functions. Then, given a test image, we estimate its quality score by finding the nearest score pivot to its feature vector in the embedding space. Extensive experiments show that the proposed QCN provides excellent performances on various datasets.

This work has the following major contributions:

- We propose the first BIQA algorithm based on geometric order learning, called QCN.
- We develop the CT networks to construct an effective embedding space, in which the feature vectors of images are sorted according to their quality scores.
- QCN achieves excellent performances on various BIQA datasets, including BID [2], CLIVE [5], KonIQ10K [9], SPAQ [4], and FLIVE [34]. Furthermore, QCN provides superb performances in cross-dataset evaluation, demonstrating its good generalization capability.

2. Related Work

2.1. Blind Image Quality Assessment

Recently, with the success of deep learning in diverse vision tasks, various deep-learning-based BIQA techniques have been proposed. Both network design and network pre-training have been researched for BIQA.

Network design: Several network structures have been developed for reliable BIQA. Zhang *et al*. [37] employed two different encoders to extract image distortion types and image contents, respectively. Su *et al*. [30] designed the local distortion aware module to identify local distortions in an image. Ke *et al*. [11] developed a transformer-based encoder to extract the distortion and content features by preserving the aspect ratio and composition of an image. However, these algorithms do not explicitly employ the ordering relations and score differences between images. Hence, for better BIQA, Golestaneh *et al*. [7] and Zhang *et al*. [38] employed the ranking loss to train a network, but these algorithms exploit the ranking relations in the training only, not in the test.

For accurate quality score estimation of images based on the ordering relations and score differences, we propose a novel network architecture for constructing an embedding space, in which such relations are well preserved.

Network pre-training: On the other hand, pre-training schemes have been developed for BIQA. Madhusudana *et al*. [22] learned to cluster images based on their distortion degrees via self-supervised learning. Also, to extract the content and distortion information, Saha *et al*. [27] proposed a self-supervised learning algorithm that pre-trains the content-aware encoder and the quality-aware encoder. Zhao *et al*. [39] applied various types of distortions to images to adopt contrastive learning in the BIQA task.

These pre-training schemes, however, demand significant training time and computational power. On the contrary, without such pre-training, the proposed QCN provides competent BIQA performances.

2.2. Order Learning

Order learning aims to predict the rank of an object by comparing it with multiple references with known ranks, and its techniques have been developed mostly for facial age estimation [12–14, 18, 28]. Lim *et al*. [18] first proposed the notion of order learning. For more reliable comparisons, Lee and Kim [12] performed the order-identity decomposition and selected references with similar identity features. Shin *et al*. [28] developed a regression approach to order learning. In practice, ordering relationships may be known for a limited amount of training data, so Lee *et al*. [13] developed a weakly-supervised training scheme for order learning. However, these algorithms should conduct multiple comparisons with many references of different ranks, since they consider only relative priorities between objects. To overcome this issue, Lee *et al*. [14] proposed GOL, which exploits metric relations, as well as order relations, among objects. GOL predicts the rank of an object via a simple k-NN search.

In this work, we adopt the concept of GOL but propose a novel transformer-based network for estimating the continuous quality score of an image. Note that the original GOL was designed for estimating discrete ranks by employing a convolutional neural network.

3. Proposed Algorithm

3.1. Problem Definition

Given an input image x, the objective of BIQA is to estimate its quality score $\theta(x)$. Note that an order and score differences provide useful information for quality score estimation; they convey complementary information. Suppose that there are three images x, y, and z, and their quality scores are 5, 10, and 20, respectively; $\theta(x) = 5$, $\theta(y) = 10$, and $\theta(z) = 20$. Since $\theta(x) < \theta(y) < \theta(z)$, the order indicates that x and z should be located at opposite sides with respect to y in a well-designed embedding space. On the other hand, since $|\theta(x) - \theta(y)| < |\theta(y) - \theta(z)|$, the score differences indicate that x should be closer from y than z is

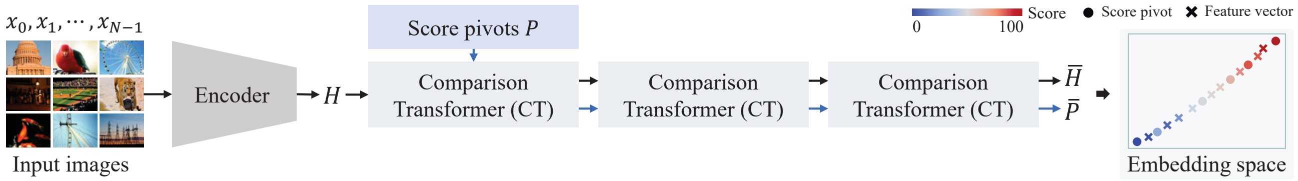


Figure 2. An overview of the proposed QCN algorithm: Given N images, the encoder extracts their feature vectors (or tokens) and forms the token matrix H. Then, the three CTs sequentially update the feature vectors, together with learnable score pivots in P, and yield the updated token matrix $\bar{H}$ and the updated pivot matrix $\bar{P}$. As a result, the updated feature vectors and score pivots are arranged according to their quality scores in the embedding space.

from y in the embedding space.

To exploit such order and score differences for BIQA, we construct an embedding space, in which the ordering relationships and the score differences are reflected by the directions and the distances between feature vectors, respectively. In other words, we attempt to sort the images in the embedding space according to their quality scores.

3.2. QCN

We develop QCN to construct an embedding space, where the feature vectors of N images, $x_0, x_1, \ldots, x_{N-1}$, from a training set $\mathcal{X}$ are arranged according to their quality scores. QCN is composed of an encoder and three CTs, as shown in Figure 2.

We adopt ResNet50 [8] as the encoder backbone. The encoder transforms the N images into feature vectors $h_{x_0}, h_{x_1}, \ldots, h_{x_{N-1}} \in \mathbb{R}^C$. Then, we form the token matrix

$$H = [h_{x_0}, h_{x_1}, \ldots, h_{x_{N-1}}]^t \in \mathbb{R}^{N\times C}. \tag{1}$$

Next, each CT updates the feature vectors (or tokens) in H to sort them according to their quality scores. To guide this update process, we introduce M score pivots $p_0, p_1, ..., p_{M-1}$, which are learnable parameters functioning as the centroids of feature vectors of similar-quality images. Thus, the first CT takes the learnable pivot matrix

$$P = [p_0, p_1, ..., p_{M-1}]^t \in \mathbb{R}^{M\times C} \tag{2}$$

as input. For example, if the score range is $[0, 100]$ and $M = 11$, these eleven pivots play the role of the representative images of scores $0, 10, \ldots, 100$ in the case of uniform quantization. Through the three CTs, the N images and the M pivots are arranged in the embedding space, as illustrated in Figure 2. The final CT yields the updated token matrix $\bar{H}$ and the updated pivot matrix $\bar{P}$.

3.3. CT

As in Figures 3, each CT comprises four modules: feature self-update (FSU), feature-pivot cross-update (FPCU), pivot self-update (PSU), and pivot-feature cross-update (PFCU) modules.

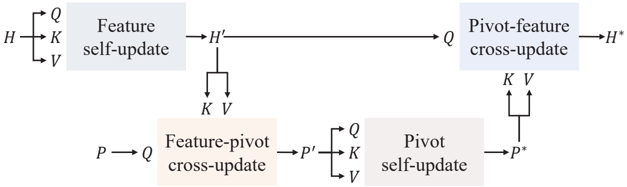


Figure 3. A block diagram of a CT.

FSU module: To construct an embedding space in which the score relations between input images are reflected, we should analyze the correlation among the images. To this end, we apply the masked self-attention [33] to H.

First, we obtain query Q_H, key K_H, value V_H by

$$Q_H = HU_q^t, \quad K_H = HU_k^t, \quad V_H = HU_v^t, \tag{3}$$

using projection matrices $U_q^t, U_k^t, U_v^t \in \mathbb{R}^{C\times C}$. Then, we analyze the correlation via

$$A_{\mathcal{M}} = \text{softmax}\left(Q_H K_H^t + \mathcal{M}\right) \tag{4}$$

where $\mathcal{M} \in \mathbb{R}^{N\times N}$ is a mask whose (i, j)th element is 0 if $i \neq j$, and $-\infty$ if $i = j$. By employing $\mathcal{M}$, each feature vector is compared with all the others, excluding itself. Notice that $Q_H K_H^t$ in (4) computes the correlation between images. Then, as in Figure 4(a), the FSU module updates each feature vector based on the correlation by

$$\begin{aligned} H' &= \text{MaskedAttention}(Q_H, K_H, V_H, \mathcal{M}) \\ &= \phi(A_{\mathcal{M}} V_H + H), \end{aligned} \tag{5}$$

where ϕ is a feedforward network.

FPCU module: In BIQA, even images with similar scores may have significantly different distortions. To cope with this problem, we use score pivots to guide the grouping of images with similar scores in the embedding space. The FPCU module updates the pivot matrix P by considering H' from the FSU module.

First, we obtain Q_P from P and $K_{H'}, V_{H'}$ from H'. Then, we perform the cross-attention to yield

$$P' = \text{Attention}(Q_P, K_{H'}, V_{H'}) = \phi(AV_{H'} + P) \tag{6}$$

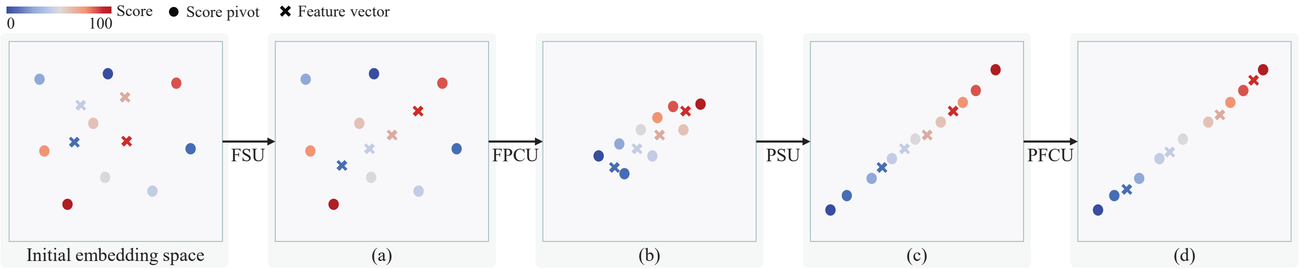


Figure 4. Illustration of the transition of feature vectors and score pivots in each CT. The feature vectors and score pivots are aligned according to their scores, as they pass through the four modules of FSU, FPCU, PSU, and PFCU.

where

$$A = \mathrm{softmax}\left(Q_P K_{H'}^t\right) \tag{7}$$

is the cross-correlation matrix. In (6), the score pivots are updated using the feature vectors, as shown in Figure 4(b).

PSU module: The score pivots should be also ordered. Thus, we first obtain $Q_{P'}$, $K_{P'}$, and $V_{P'}$ from P'. Then, we perform the self-attention on the pivots in P' and yield the updated pivot matrix

$$P^* = \mathrm{Attention}(Q_{P'}, K_{P'}, V_{P'}) \tag{8}$$

as illustrated in Figure 4(c).

PFCU module: Finally, we update H' based on P^*. We obtain $Q_{H'}$ from H' and K_{P^*}, V_{P^*} from P^*. Then, we apply the cross-attention to the updated token matrix

$$H^* = \mathrm{Attention}(Q_{H'}, K_{P^*}, V_{P^*}) \tag{9}$$

as in Figure 4(d).

3.4. Loss Functions

To arrange input images according to their ordering relationships and score differences, we train QCN with the order loss and the metric loss. Also, for accurate score estimation, we employ the center loss and the mean absolute error (MAE) loss. Note that we compute these four losses on $\bar{H}$ and $\bar{P}$, which are the output of the last CT in Figure 2.

Order loss: We design the order loss to arrange the score pivots according to their order. Let us define the direction vector $v(r, s)$ from point r to point s in the embedding space as

$$v(r, s) = \frac{s - r}{\|s - r\|}. \tag{10}$$

Then, we define the order loss as

$$L_{\mathrm{order}} = \textstyle\sum_{m=1}^{M-2} v(\bar{p}_m, \bar{p}_{m-1})^t v(\bar{p}_m, \bar{p}_{m+1}). \tag{11}$$

To minimize this term, the angle between $v(\bar{p}_m, \bar{p}_{m-1})$ and $v(\bar{p}_m, \bar{p}_{m+1})$ should be maximized as shown in Figure 5(a), which means that the three consecutive pivots $\bar{p}_{m-1}$, $\bar{p}_m$, and $\bar{p}_{m+1}$ should be aligned on a line.

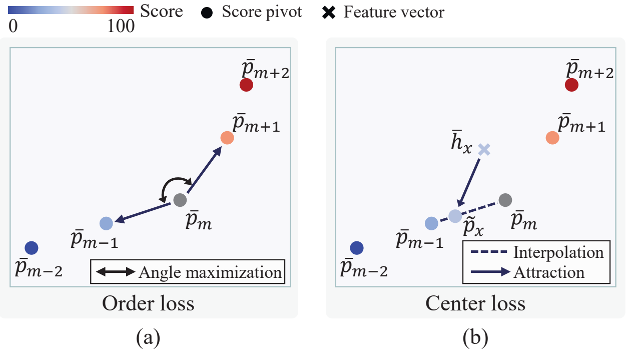


Figure 5. Computation of (a) the order loss and (b) the center loss.

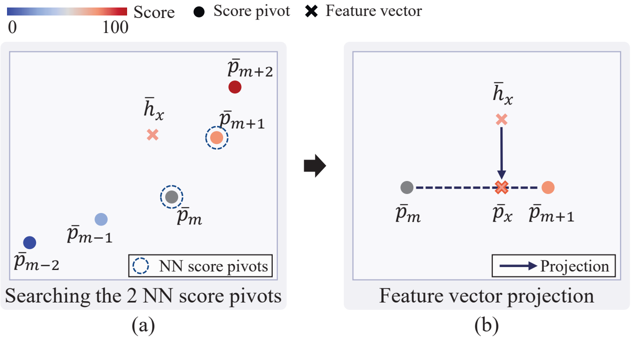
Metric loss: Note that the order loss in (11) considers each triple of consecutive score pivots, thus it attempts to arrange the score pivots locally. To consider the global relationship of all pivots as well, we adopt the metric constraint in [14] as the metric loss.

Center loss: In the embedding space, the feature vector $\bar{h}_x$ of an image should be near its corresponding score pivot. However, since we use a finite number of score pivots to represent the continuous score range, there may be no score pivot exactly matching $\bar{h}_x$. Hence, we first obtain a linearly interpolated score pivot

$$\tilde{p}_x = \frac{(\theta(\bar{p}_{m+1}) - \theta(x))\bar{p}_m + (\theta(x) - \theta(\bar{p}_m))\bar{p}_{m+1}}{\theta(\bar{p}_{m+1}) - \theta(\bar{p}_m)} \tag{12}$$

where $\theta(\bar{p}_m) \leq \theta(x) \leq \theta(\bar{p}_{m+1})$. Note that $\tilde{p}_x$ is an internally dividing point between the two nearest pivots in terms of quality scores, as in Figure 5(b). Then, we attempt to minimize the distance $\|\bar{h}_x - \tilde{p}_x\|$. These distances are computed for all feature vectors in $\bar{H}$, and the center loss is defined as

$$L_{\mathrm{center}} = \textstyle\sum_{n=0}^{N-1} \|\bar{h}_{x_n} - \tilde{p}_{x_n}\|. \tag{13}$$



Searching the 2 NN score pivots

(a)

Feature vector projection

(b)

Figure 6. To estimate the score of image x, we first find the two nearest neighbor (NN) score pivots $\bar{p}_m$ and $\bar{p}_{m+1}$ in (a), and then project the feature vector $\bar{h}_x$ onto the line from $\bar{p}_m$ to $\bar{p}_{m+1}$ in (b).

MAE loss: We estimate the score of a test instance using the rule in Section 3.5. To minimize the difference between the estimate score $\hat{\theta}(x_n)$ and the ground-truth score $\theta(x_n)$ for each x_n, we adopt the smooth MAE loss L_{mae} in [6].

Finally, we minimize the overall loss function

$$L = L_{\text{order}} + L_{\text{metric}} + L_{\text{center}} + L_{\text{mae}} \tag{14}$$

to optimize the network parameters in QCN and learn the score pivots in P.

3.5. Score Estimation

Given an unseen test image x, we estimate its quality score by applying it together with $N-1$ auxiliary images into QCN. We select the $N-1$ auxiliary images from the training set $\mathcal{X}$. More specifically, we first divide the entire score range uniformly into the $N-1$ intervals. Then, we randomly select an image from each interval. It is shown in the supplemental document that the score estimation performance is not sensitive to this random selection.

Then, QCN yields the feature vector $\bar{h}_x$ of the test image and the score pivots in $\bar{P}$, which are aligned in the embedding space. Note that the feature vectors of the auxiliary images are not employed in the score estimation. Then, we find the adjacent pair of score pivots $\bar{p}_m$ and $\bar{p}_{m+1}$ minimizing the sum $\|\bar{h}_x - \bar{p}_m\| + \|\bar{h}_x - \bar{p}_{m+1}\|$, as illustrated in Figure 6(a). In other words, we search the two nearest neighbor (NN) pivots of $\bar{h}_x$ in terms of Euclidean distances. Then, we project $\bar{h}_x$ onto the line from $\bar{p}_m$ to $\bar{p}_{m+1}$, as in Figure 6(b). Hence, the projected point is given by

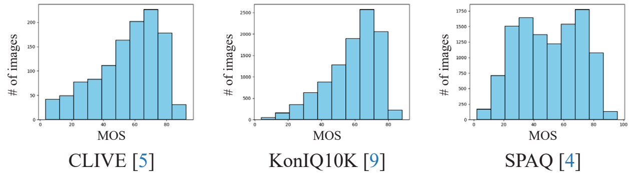
$$\bar{p}_x = \bar{p}_m + \alpha(\bar{p}_{m+1} - \bar{p}_m), \tag{15}$$

where

$$\alpha = \frac{(\bar{h}_x - \bar{p}_m)^t(\bar{p}_{m+1} - \bar{p}_m)}{\|\bar{p}_{m+1} - \bar{p}_m\|^2}. \tag{16}$$

Then, the score of x is estimated by

$$\hat{\theta}(x) = \theta(\bar{p}_m) + \alpha\big(\theta(\bar{p}_{m+1}) - \theta(\bar{p}_m)\big). \tag{17}$$



CLIVE [5]   KonIQ10K [9]   SPAQ [4]

Figure 7. MOS histograms of three BIQA datasets.

4. Experimental Results

4.1. Implementation

Training details: We initialize the encoder using ResNet50 pre-trained on ILSVRC2012 [3]. We use the AdamW optimizer [20] with a batch size of 54 and a weight decay of 5×10^{-4}. We set the learning rate to 5×10^{-5} initially and decrease it using the cosine annealing learning rate scheduler. By default, the number M of score pivots and the number N of input images are set to be 101 and 18, respectively. In IQA, changing the aspect ratio and composition of an image may impact the image quality. Hence, as done in [11], we preserve the aspect ratio of an image during both training and testing. Specifically, we resize the short side of an image to 384 while maintaining the aspect ratio. For evaluation, we estimate the quality score of an image and its horizontally flipped version. Then, we average the prediction scores of the two images. More details are available in the supplemental document.

Non-uniform score pivot generation: We quantize the entire score range to M reconstruction levels, which are assigned to the M pivots as the scores. In general, the distribution of quality scores in a BIQA dataset is not uniform, as shown in Figure 7. Hence, to minimize quantization errors, we adopt the Lloyd-Max algorithm [19], instead of uniform quantization. The impacts of this non-uniform quantization will be analyzed in Section 4.4.

4.2. Datasets and Evaluation Protocol

We use five IQA datasets to assess the performance of the proposed QCN.

- BID [2]: It provides 586 images with blur artifacts, *e.g.*, due to out-of-focus, complex motion, and simple motion.
- CLIVE [5]: It contains 1,162 images in diverse categories taken from different cameras.
- KonIQ10K [9]: It consists of 10,073 images selected from YFCC-100M [31] to cover various types of distortions.
- SPAQ [4]: It provides 11,125 photos taken with 66 smartphones.
- FLIVE [34]: It is one of the largest BIQA datasets, which contains about 40K images and 120K patches. As done in [22, 34, 39], we only use the images, not the patches, for both training and testing.

Table 1. Comparison of BIQA results on the BID, CLIVE, KonIQ10k, SPAQ, and FLIVE datasets. Pre-training algorithms are marked with the asterisk *. The best results are boldfaced, and the second-best are underlined.

	BID		CLIVE		KonIQ10k		SPAQ		FLIVE	
Algorithm	SRCC	PCC	SRCC	PCC	SRCC	PCC	SRCC	PCC	SRCC	PCC
NIQE [25]	0.477	0.471	0.454	0.468	0.526	0.475	0.697	0.685	0.105	0.141
ILNIQE [36]	0.495	0.454	0.453	0.511	0.503	0.496	0.719	0.654	0.219	0.255
BRISQUE [24]	0.574	0.540	0.601	0.621	0.715	0.702	0.802	0.806	0.320	0.356
BMPRI [23]	0.515	0.458	0.487	0.523	0.658	0.655	0.750	0.754	0.274	0.315
CNNIQA [10]	0.616	0.614	0.627	0.601	0.685	0.684	0.796	0.799	0.306	0.285
WaDIQaM-NR [1]	0.653	0.636	0.692	0.730	0.729	0.754	0.840	0.845	0.435	0.430
PQR [35]	0.775	0.794	0.857	0.882	0.880	0.884	-	-	-	-
SFA [15]	0.820	0.825	0.804	0.821	0.888	0.897	0.906	0.907	0.542	0.626
DB-CNN [37]	0.845	0.859	0.844	0.862	0.878	0.887	0.910	0.913	0.554	0.652
HyperIQA [30]	0.869	0.878	0.859	0.882	0.906	0.917	0.916	0.919	0.535	0.623
PaQ-2-PiQ [34]	-	-	0.840	0.850	0.870	0.880	-	-	0.571	0.623
UNIQUE [38]	0.858	0.873	0.854	0.890	0.896	0.901	-	-	-	-
MUSIQ [11]	-	-	-	-	0.916	0.928	0.917	0.921	**0.646**	0.739
TReS [7]	-	-	0.846	0.877	0.915	0.928	-	-	0.554	0.625
CONRTIQUE* [22]	-	-	0.845	0.857	0.894	0.906	0.914	0.919	0.580	0.641
Re-IQA* [27]	-	-	0.840	0.854	0.914	0.923	0.918	0.925	0.645	0.733
QPT* [39]	0.888	**0.911**	**0.895**	**0.914**	0.927	0.941	**0.925**	**0.928**	0.610	0.677
Proposed QCN	**0.892**	0.890	0.875	0.893	**0.934**	**0.945**	0.923	**0.928**	0.644	**0.741**

Table 2. Cross-dataset evaluation results in SRCC. The first and second rows specify the training and test datasets, respectively.

	BID		CLIVE		KonIQ10k	
Algorithm	CLIVE	KonIQ10K	BID	KonIQ10K	BID	CLIVE
DBCNN [37]	0.725	0.724	0.762	0.754	0.816	0.755
PQR [35]	0.680	0.636	0.714	0.757	0.755	0.770
HyperIQA [30]	0.770	0.688	0.756	0.772	0.819	0.785
TReS [7]	-	-	-	0.733	-	0.786
CONTRIQUE* [22]	-	-	-	0.676	-	0.731
Re-IQA* [27]	-	-	-	0.769	-	0.791
QPT* [39]	-	-	0.845	0.749	0.825	0.821
Proposed QCN	**0.800**	**0.730**	**0.886**	**0.784**	**0.847**	**0.840**

We adopt the Spearman's rank correlation coefficient (SRCC) [29] and Pearson's correlation coefficient (PCC) [26] metrics. SRCC and PCC measure how well a network sorts images according to their ranks and scores, respectively.

For the BID, CLIVE, KonIQ10K, and SPAQ datasets, we randomly split each dataset into train and test sets with a ratio of 8:2. Then, we repeat the training and testing for 10 different splits and report the median SRCC and PCC scores, as done in [11, 27, 30, 39]. For FLIVE, we employ the same evaluation protocol as in [22, 34, 39] — 30K images for training and 1.8K images for testing.

4.3. Comparative Assessment

Table 1 compares the performances of the proposed QCN with those of conventional algorithms on the five IQA datasets. Note that the pre-training algorithms [22, 27, 39] are listed separately in the middle section of the table.

Comparison with network design techniques: The proposed QCN is one of the network design techniques. We see that, in Table 1, QCN outperforms all conventional network design techniques in 9 out of 10 tests.

Compared with MUSIQ [11], which is the state-of-the-art in the network design approach, QCN improves the SRCC and PCC performances by 1.97% and 1.83%, respectively, on KonIQ10K. Also, on FLIVE, which is a challenging dataset with various types of distortions, QCN yields comparable and better results than MUSIQ. It is worth pointing out that, while we use only 30K training images for FLIVE as in [22, 34, 39], MUSIQ exploits 90K training patches additionally to boost their performances on FLIVE.

Comparison with network pre-training techniques: Even without pre-training the network, the proposed QCN provides competent results to the pre-training techniques.

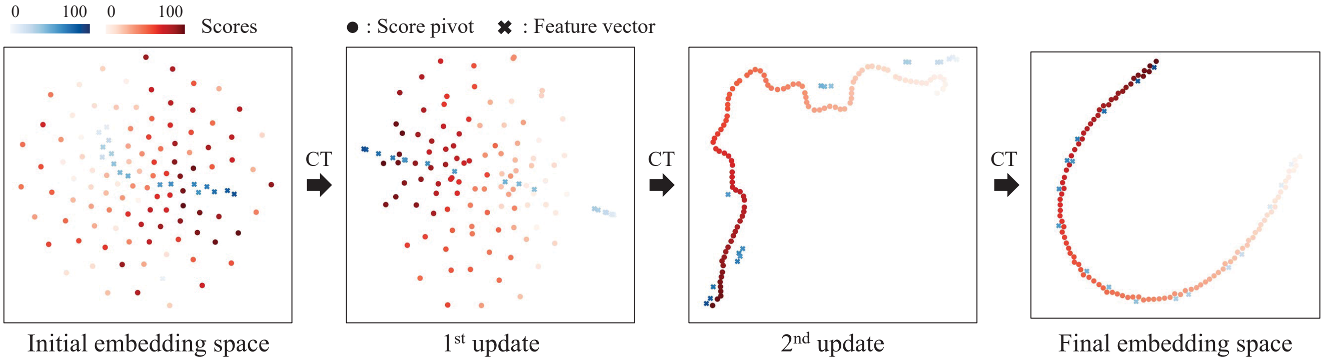


Figure 8. t-SNE visualization [32] of feature vectors and score pivots for the KonIQ10K dataset in each CT. We depict the scores of the score pivots and the feature vectors in red and blue shades, respectively.

Table 3. Ablation studies for the FSU and PSU modules in QCN on the KonIQ10K dataset.

Method	FPCU / PFCU	FSU	PSU	SRCC	PCC
I	✓			0.858	0.849
II	✓	✓		0.916	0.881
III	✓		✓	0.930	0.939
IV	✓	✓	✓	0.934	0.945

Table 4. Ablation studies for the loss functions in (14) on the KonIQ10K dataset.

Method	L_{mae}	L_{center}	L_{metric}	L_{order}	SRCC	PCC
I	✓				0.855	0.866
II	✓	✓			0.860	0.877
III	✓	✓	✓		0.929	0.939
IV	✓	✓	✓	✓	0.934	0.945

In Table 1, it provides better results than those techniques in 5 out of 10 tests.

Compared with the state-of-the-art QPT [39], QCN improves the results by 0.76% in SRCC and 0.43% in PCC on KonIQ10K. Note that the pre-training is beneficial, especially for small datasets. However, even on BID, which contains only about 470 training images, QCN yields comparable results to QPT.

Cross-dataset evaluation: Table 2 compares cross-dataset evaluation results. Even without pre-training, QCN performs the best in all 6 tests. Also, in the challenging combination of training on CLIVE (1,162 images) and testing on KonIQ10K (10,073 images), QCN outperforms the second-best HyperIQA [30] by 1.55%. This indicates that QCN has better generalization capability than the other algorithms, including the pre-training techniques.

4.4. Analysis

Efficacy of FSU and PSU modules: We conduct ablation studies to analyze the efficacy of the FSU and PSU modules in a CT in Figure 3. In Table 3, we compare ablated methods on the KonIQ10K dataset. Method I uses the FPCU and PFCU modules only. In II and III, FSU and PSU are additionally used, respectively.

Compared with the full QCN in IV, method I degrades the results severely. By employing FSU and PSU, II and III perform better than method I, but the gaps with IV are still large. Both FSU and PSU modules are essential for reliable feature vector arrangement.

Loss functions: Table 4 compares ablated methods for the loss terms in (14). Method I employs only L_{mae}. In II, III, and IV, we additionally use L_{center}, L_{metric}, and L_{order} in that order.

From I and II, we see that L_{center} improves the results, by encouraging the feature vector of an input to be located near its corresponding score pivots. However, compared with the proposed QCN in IV, methods I and II yield inferior results, for we cannot sort feature vectors meaningfully without L_{order} and L_{metric}. By employing L_{metric}, III outperforms II significantly. Also, by comparing IV with III, we see that L_{order} further improves the results.

Embedding space visualization: Figure 8 visualizes how feature vectors and score pivots for KonIQ10K are aligned through the three CTs. The t-SNE method [32] is used for the visualization. Note that they are gradually arranged and separated according to their scores, as the update goes on.

Performance according to N: Table 5 compares the results according to the number N of input images on KonIQ10K. Without auxiliary images ($N = 1$), the performance degrades severely because we cannot exploit the score relations between images. As N increases, the performance gets better but saturates around the default $N = 18$.

Performance according to T: Table 6 compares the results according to the number T of CTs on KonIQ10K. The best

Table 5. Comparison of the performances according to the number N of input images on the KonIQ10K dataset.

N	1	6	12	18	24
SRCC	0.909	0.928	0.931	0.935	0.934
PCC	0.924	0.940	0.942	0.945	0.944

Table 6. Comparison of the performances according to the number T of CTs on the KonIQ10K dataset.

T	1	2	3	4	5
SRCC	0.928	0.930	0.934	0.933	0.933
PCC	0.939	0.941	0.945	0.943	0.943

Table 7. Comparison of the SRCC and PCC scores on KonIQ10K and SPAQ according to the score pivot generation schemes.

	KonIQ10K		SPAQ	
	SRCC	PCC	SRCC	PCC
Uniform	0.929	0.941	0.914	0.919
Non-uniform	0.934	0.945	0.923	0.928

Table 8. Comparison of QCN with GOL [14] on the KonIQ10K and SPAQ datasets.

	KonIQ10K		SPAQ	
	SRCC	PCC	SRCC	PCC
GOL [14]	0.918	0.909	0.908	0.907
Proposed QCN	0.934	0.945	0.923	0.928

results are achieved at the default $T = 3$.

Non-uniform score pivot generation: We use the Lloyd-Max algorithm to quantize the scores of pivots non-uniformly. Table 7 compares this scheme with the uniform quantization on the KonIQ10K and SPAQ datasets. We see that the non-uniform quantization yields better results than the uniform quantization on both datasets, so it is used as the default mode.

Comparison with geometric order learning: Table 8 compares the proposed QCN with the GOL algorithm [14] on the KonIQ10K and SPAQ datasets. Since GOL is designed for discrete rank estimation, it may fail to yield accurate score predictions. Therefore, QCN performs better than GOL in BIQA.

Testing time: To estimate the quality score of a test image, the proposed QCN exploits auxiliary images, which are selected from a training set. For efficiency, we extract the features of all auxiliary images in advance. Hence, during the test, the feature extraction of the auxiliary images is not required. We measure the testing time on KonIQ10K using an RTX 3090 GPU. QCN takes only 0.033s to test an image on average: 0.006s for the feature extraction, 0.001s for the auxiliary image selection, and 0.026s for the score estimation. Hence, QCN is feasible for many use cases.

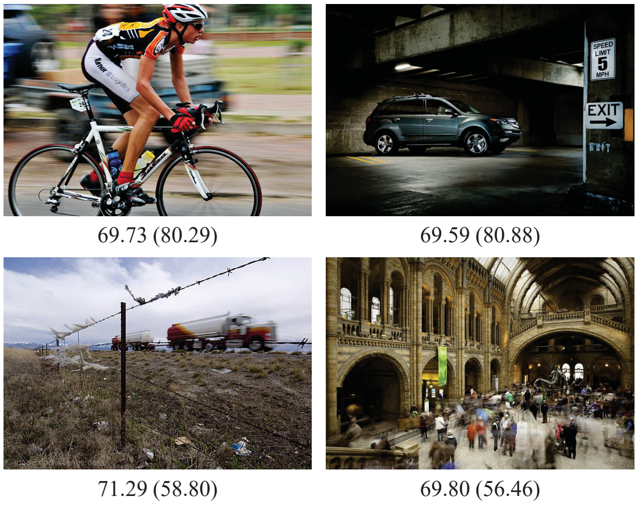

69.73 (80.29)
69.59 (80.88)
71.29 (58.80)
69.80 (56.46)

Figure 9. Failure cases of the proposed QCN algorithm. For each image, the predicted score is reported with the ground truth within the parentheses.

Failure cases: Figure 9 shows some failure cases of the proposed QCN. The first row shows images underrated by QCN. In these cases, QCN may estimate the scores by focusing on the blurred or poorly illuminated background, while the annotators may rate the quality scores by focusing on the standing out composition of the foreground objects. On the other hand, the second row shows images overrated by QCN. In these cases, QCN and the annotators may determine the qualities based on clear background and blurred foreground, respectively.

5. Conclusions

We proposed a novel BIQA algorithm, called QCN, which arranges the feature vectors of images based on their quality scores in the embedding space. First, we designed the CT module to update feature vectors to sort them according to their quality scores. Second, to guide this update process, we introduced score pivots. Third, we employed the four losses to arrange the feature vectors meaningfully. Lastly, we predicted the quality score of a test image by finding the nearest score pivot to its feature vector in the embedding space. Extensive experiments on various BIQA datasets showed that QCN provides excellent performance. Furthermore, QCN demonstrated its great generalization capability in cross-dataset evaluation.

Acknowledgements

This work was supported by the NRF grants funded by the Korea government (MSIT) (No. NRF-2021R1A4A1031864 and No. NRF-2022R1A2B5B03002310).

References

[1] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE TIP*, 27:206–219, 2017. 6

[2] Alexandre Ciancio, Eduardo AB da Silva, Amir Said, Ramin Samadani, and Pere Obrador. No-reference blur assessment of digital pictures based on multifeature classifiers. *IEEE TIP*, 20:64–75, 2010. 2, 5

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[4] Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *CVPR*, 2020. 2, 5

[5] Deepti Ghadiyaram and Alan C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE TIP*, 25:372–7387, 2015. 2, 5

[6] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 5

[7] S. Alireza Golestaneh, Saba Dadsetan, and Kris M. Kitani. No-reference image quality assessment via transformers, relative ranking, and self-consistency. In *WACV*, 2022. 1, 2, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 3

[9] Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE TIP*, 29:4041–4056, 2020. 2, 5

[10] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, 2014. 6

[11] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *ICCV*, 2021. 1, 2, 5, 6

[12] Seon-Ho Lee and Chang-Su Kim. Deep repulsive clustering of ordered data based on order-identity decomposition. In *ICLR*, 2021. 2

[13] Seon-Ho Lee and Chang-Su Kim. Order learning using partially ordered data via chainization. In *ECCV*, 2022. 2

[14] Seon-Ho Lee, Nyeong-Ho Shin, and Chang-Su Kim. Geometric order learning for rank estimation. In *NeurIPS*, 2022. 1, 2, 4, 8

[15] Dingquan Li, Tingting Jiang, Weisi Lin, and Ming Jiang. Which has better visual quality: The clear blue sky or a blurry animal? *IEEE TMM*, 21:1221–1234, 2018. 6

[16] Yang Li, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Yue Wang. Quality assessment of end-to-end learned image compression: The benchmark and objective measure. In *ACM MM*, 2021. 1

[17] Haoyi Liang and Daniel S. Weller. Comparison-based image quality assessment for selecting image restoration parameters. *IEEE TIP*, 25:5118–5130, 2016. 1

[18] Kyungsun Lim, Nyeong-Ho Shin, Young-Yoon Lee, and Chang-Su Kim. Order learning and its application to age estimation. In *ICLR*, 2020. 2

[19] Stuart Lloyd. Least squares quantization in PCM. *TIT*, 28:129–137, 1982. 5

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[21] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *CVIU*, 158:1–16, 2017. 1

[22] Pavan C. Madhusudana, Neil Birkbeck, Yilin Wang, Balu Adsumilli, and Alan C. Bovik. Image quality assessment using contrastive learning. *IEEE TIP*, 31:4149–4161, 2022. 1, 2, 5, 6

[23] Xiongkuo Min, Guangtao Zhai, Ke Gu, Yutao Liu, and Xiaokang Yang. Blind image quality estimation via distortion aggravation. *IEEE TB*, 64:508–517, 2018. 6

[24] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE TIP*, 21:4695–4708, 2012. 6

[25] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Sign. Process. Letters*, 20:209–212, 2012. 6

[26] Karl Pearson. Determination of the coefficient of correlation. *Science*, 30:23–25, 1909. 6

[27] Avinab Saha, Sandeep Mishra, and Alan C. Bovik. Re-IQA: Unsupervised learning for image quality assessment in the wild. In *CVPR*, 2023. 1, 2, 6

[28] Nyeong-Ho Shin, Seon-Ho Lee, and Chang-Su Kim. Moving window regression: A novel approach to ordinal regression. In *CVPR*, 2022. 2

[29] Charles Spearman. Footrule for measuring correlation. *British Journal of Psychology*, 2:89, 1906. 6

[30] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *CVPR*, 2020. 1, 2, 6, 7

[31] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59:64–73, 2016. 5

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9 (11):2579–2605, 2008. 7

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[34] Zhenqiang Ying, Haoran Niu, Praful Gupta, Dhruv Mahajan, Deepti Ghadiyaram, and Alan Bovik. From patches to pictures (PaQ-2-PiQ): Mapping the perceptual space of picture quality. In *CVPR*, 2020. 2, 5, 6

[35] Hui Zeng, Lei Zhang, and Alan C Bovik. A probabilistic quality representation approach to deep blind image quality prediction. *arXiv preprint arXiv:1708.08190*, 2017. 6

[36] Lin Zhang, Lei Zhang, and Alan C. Bovik. A feature-enriched completely blind image quality evaluator. *IEEE TIP*, 8:2579–2591, 2012. 6

[37] Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE TCSVT*, 30:36–47, 2018. 1, 2, 6

[38] Weixia Zhang, Kede Ma, Guangtao Zhai, and Xiaokang Yang. Uncertainty-aware blind image quality assessment in the laboratory and wild. *IEEE TIP*, 30:3474–3486, 2021. 1, 2, 6

[39] Kai Zhao, Kun Yuan, Ming Sun, Mading Li, and Xing Wen. Quality-aware pre-trained models for blind image quality assessment. In *CVPR*, 2023. 1, 2, 5, 6, 7