– Supplemental Document –

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

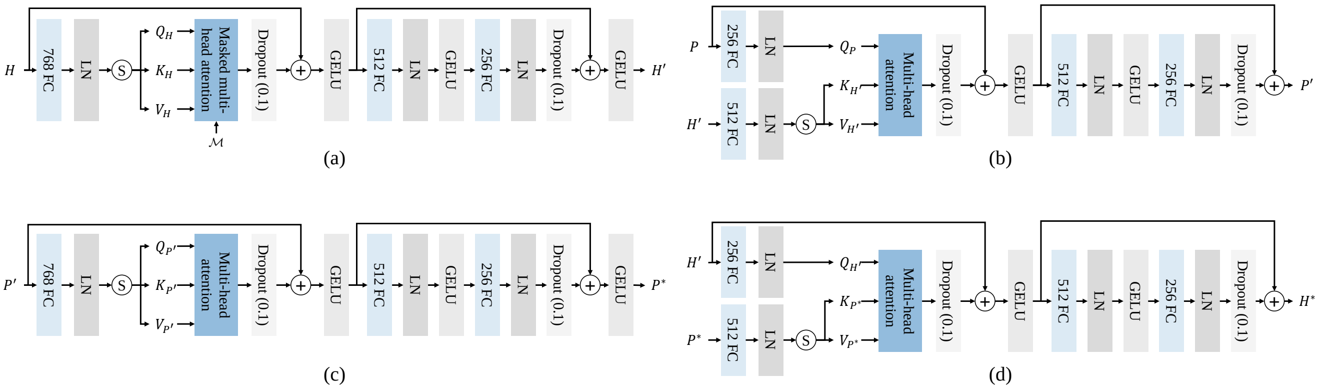
Chang-Su Kim
Korea University
changsukim@korea.ac.kr

1. Implementation Details

We set the number of channels of each feature vector to $C = 256$. Also, the number M of score pivots and the number N of input images are fixed to be 101 and 18, respectively, except that $M = 6$ and $N = 6$ for the BID dataset [3], whose scores range from 0 to 5. We resize the short side of an image to 384 while maintaining the aspect ratio. For the FLIVE dataset [18], we halve an image both horizontally and vertically. We train the proposed QCN for 100 epochs. We set the learning rate to 5×10^{-5} initially and decrease it using the cosine annealing learning rate scheduler. Specifically, we first linearly increase the learning rate from 0 to 5×10^{-5} for 5 epochs. Then, we decrease it using the scheduler for 95 additional epochs. We do not do data augmentation during training.

2. Network Architecture

The structure of each component in CT is detailed in Figure S-1. In Figure S-1(a), (b), (c), and (d), the FSU, FPCU, PSU, and PFCU modules are presented, respectively. Note that 'c FC' denotes the fully connected layer with c output channels and 'LN' means the layer normalization [1]. In FSU, masked attention operations in (5) in the main paper are implemented by masked multi-head attention layers with 8 heads. Also, in FPCU, PSU, and PFCU, attention operations in (6), (8), and (9) in the main paper are implemented by multi-head attention layers with 8 heads.



(a) (b) (c) (d)

Figure S-1. Detailed structure of each component in CT: (a) FSU, (b) FPCU, (c) PSU, and (d) PFCU. Here, Ⓢ denotes the channel split of a feature vector.

3. More Experimental Results

3.1. More Results on KonIQ10K

In the main paper, to compare the performance on KonIQ10K [10], we randomly split the entire dataset into train and test sets with a ratio of 8:2. On the other hand, in Table S-1, we adopt the fixed train and test split from [10]. We see that, in this split setting as well, the proposed QCN outperforms the conventional algorithms.

Table S-1. Comparison results on the fixed split for KonIQ10K.

Algorithm	SRCC	PCC
DeepRN [17]	0.867	0.880
DeepBIQ [2]	0.907	0.911
KonCept512 [10]	0.921	$\underline{0.937}$
MUSIQ [11]	$\underline{0.924}$	$\underline{0.937}$
Proposed QCN	**0.931**	**0.942**

3.2. Sensitivity to Auxiliary Images

To estimate the quality score of a test image, QCN takes it together with $N-1$ auxiliary images, which are selected from the training set. Specifically, we first split the entire score range uniformly to $N-1$ intervals, and randomly select one image from each interval. QCN yields slightly different results due to this randomness; it is not very sensitive to the selection of the auxiliary images.

Table S-2 summarizes 10 evaluation results on each of the BID [3], CLIVE [5], KonIQ10K [10], SPAQ [4], and FLIVE [18] datasets. Note that the deviations are negligible.

Table S-2. Multiple evaluation results on BID, CLIVE, KonIQ10K, SPAQ, and FLIVE. The means and standard deviations of SRCC and PCC are reported.

	BID	CLIVE	KonIQ10K	SPAQ	FLIVE
SRCC	0.89342 ± 0.00069	0.87791 ± 0.00101	0.93393 ± 0.00002	0.92280 ± 0.00001	0.64358 ± 0.00002
PCC	0.88926 ± 0.00072	0.89341 ± 0.00113	0.94513 ± 0.00004	0.92761 ± 0.00004	0.74147 ± 0.00023

3.3. Sensitivity to Score Pivot Initialization

Table S-3 lists the results of four different score pivot parameter initialization schemes on the KonIQ10K dataset. Initializing all score pivot parameters to zero results in ineffective network training. Hence, the performance degrades significantly. Meanwhile, the other three schemes yield comparable results. Since the truncated normal method achieves the best results, we adopt it as the default option.

Table S-3. Comparison of the performances of four different initialization schemes on the KonIQ10K dataset.

	Zeros	Truncated normal	Kaiming normal [8]	Xavier normal [6]
SRCC	0.515	0.934	0.933	0.931
PCC	0.510	0.945	0.945	0.943

3.4. Storage Costs

Table S-4 lists memory requirements for the feature vectors of auxiliary images for the score estimation. The memory requirements are negligible, only 8.83KB for both datasets.

Table S-4. Memory requirements for the feature vectors of auxiliary images for the KonIQ10K and SPAQ datasets.

	KonIQ10K	SPAQ
Memory requirement	8.83KB	8.83KB

3.5. Performance According to M

Table S-5 compares the results according to the number M of score pivots on the KonIQ10K dataset. The best results are achieved at $M = 101$, which is used as the default option.

Table S-5. Comparison of the performances according to M on the KonIQ10K dataset.

M	26	51	101	201	401
SRCC	0.931	0.933	0.934	0.933	0.930
PCC	0.943	0.944	0.945	0.944	0.939

3.6. Performance According to Encoder Backbone

We adopted ResNet50 [9] as an encoder. Table S-6 lists the results using ResNet50 and three different encoders on KonIQ10K. Even with these different encoders, QCN outperforms the state-of-the-arts.

Table S-6. Comparison of the performances according to encoder structure on the KonIQ10K dataset.

Encoder	ResNet50 [9]	ResNet101 [9]	ConvNext-B [13]	Swin-S [12]
SRCC	0.934	0.938	0.944	0.945
PCC	0.945	0.948	0.954	0.956

3.7. Model Complexity

Table S-7 compares the complexity of the proposed QCN with those of conventional algorithms. We see that QCN requires a similar number of parameters to the conventional algorithms. However, QCN provides excellent performances on various IQA datasets, as listed in Table 1 in the main paper.

Table S-7. Comparison of model complexities.

	HyperIQA [15]	MUSIQ [11]	TReS [7]	Re-IQA [14]	Proposed QCN
Parameters (M)	27	27	152	47	30

3.8. Embedding Space Visualization

Figure S-2 visualizes how feature vectors and score pivots for CLIVE and SPAQ are aligned through the three CTs. The t-SNE [16] is used for the visualization. In both datasets, they are gradually arranged and separated according to their scores, as the update goes on.

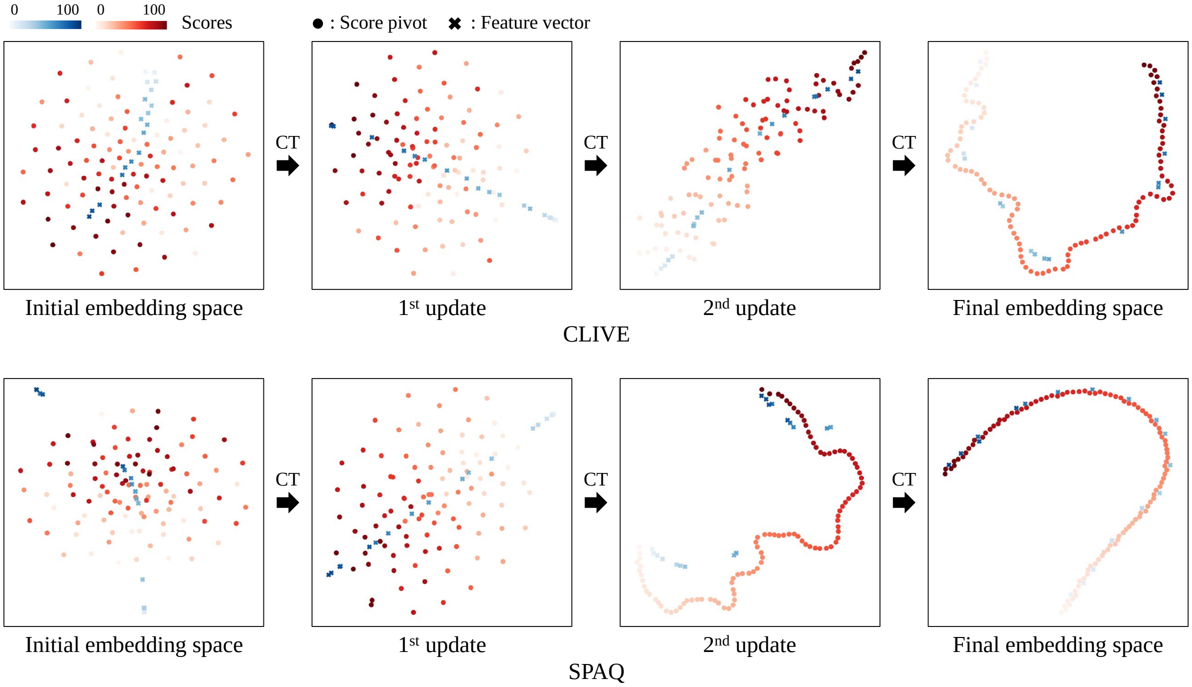


Figure S-2. t-SNE visualization [16] of feature vectors and score pivots in each CT for the CLIVE and SPAQ datasets. We depict the scores of the score pivots and the feature vectors in red and blue shades, respectively.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *arXiv preprint arXiv:1607.06450*, 2016. 1

[2] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. On the use of deep learning for blind image quality assessment. *Signal, Image and Video Processing*, 12:355–362, 2018. 2

[3] 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. 1, 2

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

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

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 2

[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. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 2

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

[10] 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

[11] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *ICCV*, 2021. 2, 3

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 3

[14] Avinab Saha, Sandeep Mishra, and Alan C. Bovik. Re-IQA: Unsupervised learning for image quality assessment in the wild. In *CVPR*, 2023. 3

[15] 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. 3

[16] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11):2579–2605, 2008. 4

[17] Domonkos Varga, Dietmar Saupe, and Tamás Szirányi. DeepRN: A content preserving deep architecture for blind image quality assessment. In *ICME*, 2018. 2

[18] 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. 1, 2