DeepRN: A CONTENT PRESERVING DEEP ARCHITECTURE FOR BLIND IMAGE QUALITY ASSESSMENT

Domonkos Varga, Dietmar Saupe, Tamás Szirányi

Budapest University of Technology
Budapest, Hungary
varga.domonkos7@upcmail.hu

University of Konstanz
Konstanz, Germany
dietmar.saupe@uni-konstanz.de

MTA SZTAKI
Budapest, Hungary
sziranyi.tamas@sztaki.mta.hu

ABSTRACT

This paper presents a blind image quality assessment (BIQA) method based on deep learning with convolutional neural networks (CNN). Our method is trained on full and arbitrarily sized images rather than small image patches or resized input images as usually done in CNNs for image classification and quality assessment. The resolution independence is achieved by pyramid pooling. This work is the first that applies a fine-tuned residual deep learning network (ResNet-101) to BIQA. The training is carried out on a new and very large, labeled dataset of 10,073 images (*KonIQ-10k*) that contains quality rating histograms besides the mean opinion scores (MOS). In contrast to previous methods we do not train to approximate the MOS directly, but rather use the distributions of scores. Experiments were carried out on three benchmark image quality databases. The results showed clear improvements of the accuracy of the estimated MOS values, compared to current state-of-the-art algorithms. We also report on the quality of the estimation of the score distributions.

Index Terms— Blind image quality assessment, deep learning, CNN, spatial pyramid pooling

1. INTRODUCTION

Blind image quality assessment (BIQA) maps images, regarded as distorted views of real-world scenes, to corresponding quality scores on a numerical scale. An algorithm for image quality assessment is called "blind" if it has no other input besides the distorted image. Since in most cases we are not able to access the undistorted versions of these digital images, BIQA has become an intensively studied field.

Proposed algorithms are judged on their performance for benchmark image quality databases that are labeled by subjective quality scores obtained from human "expert" viewers. In these databases contributors usually publish images with their Mean Opinion Scores (MOS), representing overall image quality. In most cases, the MOS is given as a decimal number ranging from 1.0 to 5.0 where 1 means the lowest quality and 5 the highest. Evaluation of algorithms is based on the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Ordered Correlation Coefficient (SROCC).

BIQA algorithms can be classified into two groups. The first group assumes the presence of a specific noise or artifact such as blur, blocking, or graininess, while the second group is distortion generic. We present a novel distortion generic BIQA algorithm. It is based on a Convolutional Neural Network (CNN) architecture that predicts the quality score distribution instead of the MOS. The quality perception of humans is inherently subjective and it is very common in public databases that images with approximately same MOS values have different quality score distributions. These distributions contain information about how much consensus or divergence exists among the individual ratings (see Figure 1), which can be learned by a system. Predicting the distributions may facilitate a relevant application for streaming content providers that may wish to guarantee a desired percentage of satisfied users at the client side rather than a certain MOS.

Recently proposed BIQA deep learning algorithms use a fine-tuned pretrained network such as VGG16 [2] that takes fixed-sized input images. However, resizing input images to the same size is not a suitable solution as it can mask some image artifacts [3], and taking only image patches of the same size as input relies on the weak assumption that image quality does not vary locally in an image. To this end we introduce an algorithm inspired by [4] which accepts input images of arbitrary resolution. Furthermore, to our best knowledge this is the first work that applies ResNet-101 for BIQA.

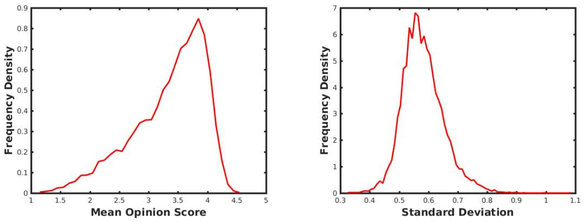

Fig. 1: The figure shows the distribution of MOS (left) in percent of the quality scale from 1 (worst) to 5 (best) and of the standard deviations (right) in KonIQ-10k [1].

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

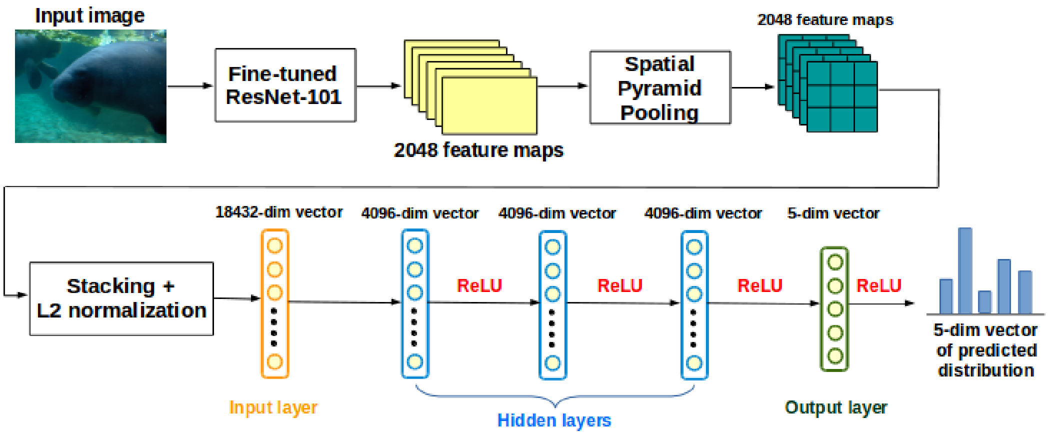

Fig. 2: Architecture of the proposed system.

In our experiments on three image quality benchmark databases we found that our method outperforms the state-of-the-art in terms of correlation with the ground truth, and can predict not only the MOS but also the rating distributions.

2. RELATED WORK

In [5], the authors addressed BIQA for images corrupted with blur. In [6] and [7], the researchers dealt with JPEG distorted images. Several other distortion specific algorithms were considered, such as blockiness [8] or ringing effects [9].

Many distortion generic models derive image quality from Natural Scene Statistic (NSS) [10] models that indicate the statistical “naturalness” of non-distorted images. Based on this theory, [11] proposed the DIIVINE index which is a 2-stage framework involving distortion identification followed by distortion-specific quality assessment. Similarly, [12] applied scene statistics but locally normalized luminance coefficients quantified possible losses of “naturalness” in the image. In this model distortion-specific features were not computed and considered. C-DIIVINE [13] is an extension of DIIVINE [11] based on the complex Gaussian scale mixture model. On the other hand, BLIINDS-II [14] relies on a Bayesian inference model to predict MOS given certain extracted features which are derived using an NSS model of the image’s Discrete Cosine Transform (DCT) coefficients.

Several deep learning approaches for BIQA were already proposed. In [15], a CNN was introduced which contains one convolutional layer incorporating 50 feature maps with max and min pooling, two fully-connected layers, and one output node. The whole structure was trained and evaluated on the LIVE database [16]. Similarly, [17] trained a CNN from scratch. A CNN with input size 32×32 was trained for unprocessed RGB image patches and the MOS of the whole image was obtained by averaging the MOS values of N randomly selected patches. The architecture of [18] is very similar to [15] but at the end of the network they utilize a softmax classification layer instead of a simple output node. [19] introduced an algorithm based on Deep Belief Nets to identify a suitable feature representation that was used to train a regressor for quality prediction.

In [3] a fine-tuned VGG16 was used as a feature extractor. Their best proposal was called DeepBIQ which predicts the image quality by average-pooling the MOSs predicted on multiple image patches. The score of each patch was determined by training a Support Vector Regression (SVR). This algorithm was evaluated on the LIVE In the Wild Image Quality Challenge Database [20] and on the LIVE database.

3. METHODS

Figure 2 presents our proposed architecture. Overall, the system takes as input a color image of arbitrary size and predicts a distribution of quality scores, which are on an integer scale from 1 to N. Therefore, for the training, a set of images and corresponding empirical score distributions is required. For this purpose, a new large-scale, authentic, and diverse image quality assessment database, *KonIQ-10k*, was recently introduced [1]. It contains 10,073 images of different content, each with 120 crowdsourced quality ratings on a 5-point scale, see Figure 1.

Our system relies on the ResNet-101 network [21], which surpassed the VGG16 [2] in image classification [22], object detection [23], and semantic segmentation [24]. VGG16 was used in BIQA as a feature extractor or as the initial network of transfer learning [3]. The main disadvantage of using such CNNs is that they accept only fixed size input images. In [3], the authors introduced a sliding-window based approach. Instead, we adopt the approach of [4] and insert an adaptive Spatial Pyramid Pooling (SPP) layer after the last convolutional layer of the ResNet-101 network. To our best knowledge this is the first work that utilizes ResNet-101 for BIQA and applies SPP [4] in order to accept arbitrarily sized images.

First of all, we fine-tune the ResNet-101 network. For this task we use the common practice. Namely, we truncate the last softmax layer of ResNet-101 and we replace it with a new softmax that is relevant to our task. Consequently, we define five categories: A (very high image quality, MOS from 80% to 100% on the quality scale), B, C, D, E (very low image quality, MOS from 0% to 20%), and ResNet-101's last layer is replaced accordingly with a 5-softmax.

The weights of the first 81 layers are frozen and the initial learning rate is set ten times smaller than the one used in [21]. The batch size is set to 128 and the momentum is adjusted to 0.9. The learning rate is set to 0.01 and divided by 10 when the validation error stops improving. We regularize the training by L_2 decay and the multiplier is $5 \cdot 10^{-4}$.

After the fine-tuning of ResNet-101, the convolutional layers are applied to an RGB input image. This way $C = 2,048$ feature maps are obtained. A CNN chiefly contains two elements: convolutional layers followed by fully-connected layers. As pointed out in [4] convolutional layers do not need a fixed size image for input and are able to produce feature maps of any sizes. However, the fully-connected layers require fixed length vectors. Therefore, we insert an SPP layer after the last convolutional layer. SPP is adaptive because every cell in the grid is resized depending on the size of the original image. In our architecture, we applied an $n \times n$ grid where $n = 3$. After the pooling layer, C matrices of dimension n^2 are stacked into a Cn^2-dimensional vector and L_2-normalized in order to normalize feature maps across images.

The SPP layer results in a $18,432$-dimensional feature vector for every image. After the SPP a Deep Neural Network (DNN) with three hidden layers is inserted. This DNN consists solely of fully-connected layers, so the input of a neuron in a layer is just the weighted sum of the neurons' outputs in the preceeding layer. The input layer has $18,432$ nodes and we set the number of neurons in the hidden layers to $4,096$. This learning problem is phrased as a regression task where the last regression layer predicts each discrete rating distribution bin independently.

Our training data consists of tuples $(\mathbf{s}_i, \mathbf{q}_i), i = 1, \ldots, M$, where M stands for the number of training samples, $\mathbf{s}_i$ is a $18,432$-dimensional feature vector and $\mathbf{q}_i$ is the corresponding empirical probability distribution of quality scores, so $\sum_{j=1}^{N} q_{i,j} = 1$. The universal approximation theorem states that a DNN is able to represent arbitrarily complex functions [25]. Here, we learn a function $\mathcal{F}(\Theta, \mathbf{s})$ that maps the feature vector $\mathbf{s}$ from the input layer to a rating distribution $\mathbf{q}$. The task is to determine parameters Θ using the classical error back-propagation algorithm.

In our network, we used Rectified Linear Units as activation functions to speed up the convergence of the training process [26]. Because fully connected layers are prone to overfitting we applied drop-out to avoid it [27]. At each training stage, individual nodes are dropped out with probability p so that a reduced network is obtained; incoming and outgoing edges are also removed. Only the reduced network is trained on the data in that stage. The removed nodes are then reinserted into the network with their original weights. In our experiments, we have used $p = 0.5$.

Those BIQA algorithms, based on CNN, that use regression to predict MOS, usually apply L1 loss [15, 17], Euclidean loss [28], or just extract features using VGG16 and feed them into an SVR [3]. In statistics, Huber loss is used in regression [29] because it is more robust to outliers in data than L1 loss or Euclidean loss. Huber loss for a scalar prediction error x is defined piecewise by

$$h_\delta(x) = \begin{cases} \frac{1}{2}x^2 & |x| \le \delta, \\ \delta \cdot (|x| - \frac{\delta}{2}) & \text{otherwise}, \end{cases} \tag{1}$$

where $\delta > 0$ controls the degree of influence given to larger prediction errors. After cross-validation, we chose $\delta = \frac{1}{9}$. Finally, we let the Huber loss for a predicted distribution $\mathbf{p} = (p_1, \ldots, p_N)$ be $L_\delta(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{N} h_\delta(p_i - q_i)$, where $\mathbf{q} = (q_1, \ldots, q_N)$ is the ground truth for the distribution of ratings.

Note that while the ground truth distribution $\mathbf{q}$ is a vector of non-negative values, normalized to unit length, the predicted distribution $\mathbf{p}$ as a result of the neural network computation need not be normalized. The MOS of a given image can be estimated from this non-normalized, predicted distribution as

$$MOS(\mathbf{p}) = \sum_{j=1}^{N} j \cdot \frac{p_j}{p_1 + \cdots + p_N}, \tag{2}$$

where the ratings are in the range $1, \ldots, N$ (in *KonIQ-10k*, $N = 5$).

We implemented the described model and loss function using the Caffe library [30]. In the training process we used stochastic gradient descent with momentum. A batch size of 128, a learning rate of 10^{-3}, momentum of 0.9, and a weight decay of $4 \cdot 10^{-4}$ were applied during training.

4. EXPERIMENTAL RESULTS

In this section we evaluate the design choices and we compare our method to state-of-the-art algorithms. For training our algorithm, we randomly selected $8,073$ out of the total of $10,073$ RGB images contained in the *KonIQ-10k* database and their corresponding distributions of quality ratings. On these training images we carried out data augmentation, generating 12 transformed images for each original one by applying 5 random rotations around the center between ±5°. Furthermore, we translated the image horizontally or vertically (5 times) by a fraction of up to 20% of the image width, respectively height. Finally, random reflections were applied horizontally or vertically.

Rotating and shifting images produces blank image regions at the borders which we removed by cropping. Thereby, we increased the number of training images from $8,073$ to $13 \cdot 8,073 = 104,949$. Training on this augmented dataset of

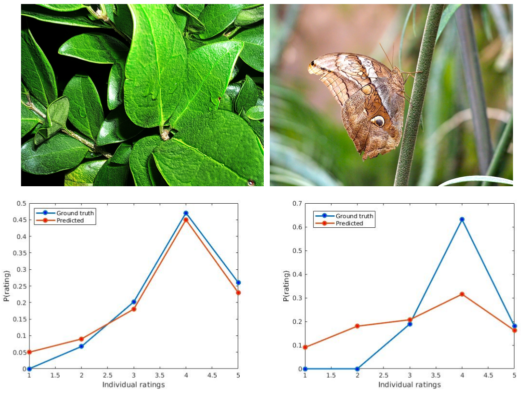

Fig. 3: Sample images from *KonIQ-10k* with plots of their ground-truth score distributions (blue line) and predicted score distributions (red line). Our model performs well on most images (left) but may suffer when the image quality varies locally (right) or the score distribution is very skewed.

Table 1: Comparison to state-of-the-art methods trained and tested on *KonIQ-10k*.

Results on KonIQ-10k		
Method	**PLCC**	**SROCC**
BosICIP	0.67	0.65
CNN	0.67	0.63
DeepBIQ	0.92	0.90
DeepRN-m-Eucl	0.89	0.88
DeepRN-Eucl	0.92	0.91
DeepRN-Huber	**0.95**	**0.92**

Table 2: Comparison to state-of-the-art measured on the *LIVE Database* [16] and the *LIVE In the Wild Image Database* [20]. All correlation values in this table, except for versions of DeepRN, are from the literature as listed.

	LIVE		**LIVE In the Wild**	
Method	**PLCC**	**SROCC**	**PLCC**	**SROCC**
BLIINDS-II [14]	0.92	0.91	0.45	0.40
S3 index [31]	0.85	0.87	0.32	0.31
C-DIIVINE [13]	0.95	0.94	0.66	0.63
FRIQUEE [20]	0.95	0.93	0.71	0.68
BosICIP [17]	0.97	0.96	0.70	0.70
CNN [15]	0.95	0.95	0.73	0.71
DeepBIQ [3]	**0.98**	0.97	0.91	0.89
DeepRN-m-Eucl	0.97	0.96	0.89	0.87
DeepRN-Eucl	**0.98**	0.97	0.91	0.90
DeepRN-Huber	**0.98**	**0.98**	**0.93**	**0.91**

~100k images with an NVIDIA GeForce Titan Xp graphics board took about two weeks.

We used three datasets for testing: (1) the 2000 remaining images in *KonIQ-10k* (not used for training), (2) the *LIVE Database Release 2* [16] consisting of 779 original and distorted images (with compression artifacts, blurring, added white noise and bit errors in the JPEG2000 stream), and (3) the *LIVE In the Wild Image Quality Challenge Database* [20] containing 1162 images with MOS and variance values.

Figure 3 presents some sample images from the test set of *KonIQ-10k* with their ground-truth score distributions (blue lines) and predicted score distributions (red lines). As shown in Figure 1, there are comparatively few images in *KonIQ-10k* with very high or very low quality ratings. We believe that as a result, the performance of our trained system on such images with very skewed rating distributions is not as good as on images with more common distributions of ratings.

To show the usefulness of the applied Huber loss, we implemented our approach also with Euclidean loss function (*DeepRN-Eucl* in Table 1 and 3).

Furthermore, we trained a model that differs from the main one in Figure 2 in one aspect. Namely, this model predicts only one real number corresponding to the MOS. This implies that the last layer contains only one neuron instead of five (*DeepRN-m-Eucl* in Table 1 and 2).

To compare the quality prediction performance on *KonIQ-10k* with that of current state-of-the-art algorithms, we reimplemented three systems, BosICIP [17], CNN [15], and DeepBIQ [3], and also trained them on the selected set of training images from *KonIQ-10k*. Table 1 presents the results. It can be seen that *DeepRN-Huber* significantly outperformed the other state-of-the-art methods. BosICIP and CNN are basically convolutional neural network architectures trained from scratch so they do not utilize pre-trained nets. DeepBIQ is based on VGG16 and examines about 30 patches from an image to predict the MOS of an image. In contrast, our method preserves the entire content of the input image and relies on ResNet-101. Table 1 shows that already with the Euclidean loss function our method performed better than the others. The Huber loss function even further reduced the gap to a perfect PLCC of 1.0 by more than 30%. Predicting only MOS, our results are comparable to those of DeepBIQ.

Application of the statistical two-sample t-test confirmed that the PLCC/SROCC values for *DeepRN-Huber* are significantly larger than the PLCC/SROCC values for the other deep learning methods (BosICIP, CNN, DeepBIQ) on the *KonIQ-10k* database at a confidence level of 95%. Furthermore, the correlation with ground truth for *DeepRN-Huber* is also significantly stronger than for *DeepRN-m-Eucl* and *DeepRN-Eucl* at a confidence level of 95%. Namely, we re-sampled the test set of *KonIQ-10k* ($2,000$ images) into 20 subsets containing 100 images. To these subsets we determined first the PLCC and then the SROCC values with respect to the ground-truth and the results of a given BIQA method. This process resulted in two 20-dimensional vectors for a given BIQA method, one vector for PLCC coefficients and the other for SROCC coefficients. Then the vector of *DeepRN-Huber*

Table 3: Accuracy of distribution prediction. We report the symmetric Kullback-Leibler divergence (KL), Kolmogorov-Smirnov distance (KS), Earth Mover's Distance (EMD), Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE) of the predicted distributions relative to the the ground-truth, averaged over all images. The PLCC and SROCC correlation coefficients between the standard deviations of the distributions of predictions and ground-truth are also listed.

Accuracy of distribution prediction for *KonIQ-10k*							
	average					**std dev**	
Method	**KL↓**	**KS↓**	**EMD↓**	**MAE↓**	**RMSE↓**	**PLCC↑**	**SROCC↑**
DeepRN-Eucl	0.130	0.175	0.172	0.064	0.081	0.75	0.73
DeepRN-Huber	0.124	0.146	0.169	0.053	0.069	0.80	0.77

containing the PLCC coefficients were compared to those of another method using the two-sample t-test. Subsequently, we did same for the vectors containing the SROCC values.

Table 2 presents the comparison to 7 other state-of-the-art algorithms measured on the *LIVE Database* and the *LIVE In the Wild Image Quality Challenge Database*. In the first case, the performances in PLCC ranged from 0.92 to 0.98 (except for the S3 Index with only 0.85). Our method performed at the top level, with a PLCC of 0.98.

For the *LIVE In the Wild database*, DeepBIQ gave the best performance of the current algorithms while ours performed still slightly better even though it was trained on a different database, i.e., *KonIQ-10k*. Our method also outperformed the other two deep learning algorithms, BosICIP and CNN, as well as all other algorithms, based on natural scene statistics. Our method predicts also the entire distributions of ratings for a stimulus rather than just the MOS. In Table 3 we report on the accuracy of these predictions of distributions.

5. CONCLUSION

In this paper, we introduced a novel framework for BIQA based on the ResNet-101 network and Huber's loss function. One innovation of this work is to predict distributions of quality ratings instead of mean opinion scores directly. Thereby, we involved information about consensus, respectively divergence, among individual ratings in the training process, which turned out to be of advantage. Moreover, predicting such distributions may be important for applications and deserves more attention in the research community. We have also shown that in this context Huber loss [29] is a better choice than Euclidean loss because outlier opinions are likely among individual ratings. We evaluated the design choices and compared our method to state-of-the-art algorithms on three benchmark image quality databases. We have used the *LIVE* and the *LIVE In the Wild* databases, and also the new database.

Acknowledgments

We thank the German Research Foundation (DFG) for financial support within project A05 of SFB/Transregio 161. We are very thankful to Hanhe Lin for helping us with the usage of the KonIQ-10k database.

6. REFERENCES

[1] Hanhe Lin, Vlad Hosu, and Dietmar Saupe, "KonIQ-10k: Towards an ecologically valid and large-scale IQA database," *Preprint arXiv:1803.08489*, 2018.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv preprint arXiv:1409.1556*, 2014.

[3] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini, "On the use of deep learning for blind image quality assessment," *ArXiv preprint arXiv:1602.05531*, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[5] Alexandre Ciancio, André Luiz N Targino da Costa, Eduardo AB da Silva, Amir Said, Ramin Samadani, and Pere Obrador, "No-reference blur assessment of digital pictures based on multifeature classifiers," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 64–75, 2011.

[6] Pina Marziliano, Frederic Dufaux, Stefan Winkler, and Touradj Ebrahimi, "Perceptual blur and ringing metrics: application to JPEG2000," *Signal Processing: Image communication*, vol. 19, no. 2, pp. 163–172, 2004.

[7] Silvia Corchs, Francesca Gasparini, and Raimondo Schettini, "No-reference image quality classification for JPEG-distorted images," *Digital Signal Processing*, vol. 30, pp. 86–100, 2014.

[8] ZM Parvez Sazzad, Yoshikazu Kawayoke, and Yuukou Horita, "No-reference image quality assessment for JPEG2000 based on spatial features," *Signal Processing: Image Communication*, vol. 23, no. 4, pp. 257–268, 2008.

[9] Xiaojun Feng and Jan P Allebach, "Measurement of ringing artifacts in JPEG images," in *Proceedings of SPIE*, 2006, vol. 6076, pp. 74–83.

[10] Alan Conrad Bovik, "Automatic prediction of perceptual image and video quality," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2008–2024, 2013.

[11] Anush Krishna Moorthy and Alan Conrad Bovik, "Blind image quality assessment: from natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, 2011.

[12] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[13] Yi Zhang, Anush K Moorthy, Damon M Chandler, and Alan C Bovik, "C-DIIVINE: no-reference image quality assessment based on local magnitude and phase statistics of natural scenes," *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 725–747, 2014.

[14] Michele A Saad, Alan C Bovik, and Christophe Charrier, "Blind image quality assessment: a natural scene statistics approach in the DCT domain," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3339–3352, 2012.

[15] Le Kang, Peng Ye, Yi Li, and David Doermann, "Convolutional neural networks for no-reference image quality assessment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1733–1740.

[16] Hamid R Sheikh, Zhou Wang, Lawrence Cormack, and Alan C Bovik, "LIVE image quality assessment database release 2 (2005)," 2016.

[17] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek, "A deep neural network for image quality assessment," in *IEEE International Conference on Image Processing*. IEEE, 2016, pp. 3773–3777.

[18] Jie Fu, Hanli Wang, and Lingxuan Zuo, "Blind image quality assessment for multiply distorted images via convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 1075–1079.

[19] Deepti Ghadiyaram and Alan C Bovik, "Blind image quality assessment on real distorted images using deep belief nets," in *IEEE Global Conference on Signal and Information Processing*. IEEE, 2014, pp. 946–950.

[20] Deepti Ghadiyaram and Alan C Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2016.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1476–1481, 2017.

[24] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei, "Fully convolutional instance-aware semantic Segmentation," *ArXiv preprint arXiv:1611.07709*, 2016.

[25] Kurt Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[27] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] Yuming Li, Lai-Man Po, Litong Feng, and Fang Yuan, "No-reference image quality assessment with deep convolutional neural networks," in *IEEE International Conference on Digital Signal Processing*. IEEE, 2016, pp. 685–689.

[29] Peter J Huber et al., "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[30] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[31] Cuong T Vu, Thien D Phan, and Damon M Chandler, "S3: a spectral and apatial measure of local perceived sharpness in natural images," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 934–945, 2012.