No-reference Image Quality Assessment with Deep Convolutional Neural Networks

Yuming Li, Lai-Man Po, Litong Feng, Fang Yuan
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, China

***Abstract*—The state-of-the-art general-purpose no-reference image or video quality assessment (NR-I/VQA) algorithms usually rely on elaborated hand-crafted features which capture the Natural Scene Statistics (NSS) properties. However, designing these features is usually not an easy problem. In this paper, we describe a novel general-purpose NR-IQA framework which is based on deep Convolutional Neural Networks (CNN). Directly taking a raw image as input and outputting the image quality score, this new framework integrates the feature learning and regression into one optimization process, which provides an end-to-end solution to the NR-IQA problem and frees us from designing hand-crafted features. This approach achieves excellent performance on the LIVE dataset and is very competitive with other state-of-the-art NR-IQA algorithms.**

Keywords-no-reference image quality assessment; convolutional neural networks; network in network

I. INTRODUCTION

Nowadays, with the rapid development of multimedia and network technology, images and videos are much easier to be generated and transmitted by many different devices, and shared by many social media, such as Facebook, Twitter, YouTube, and Instagram. Since a large number of video contents are produced every day for entertainment or education of human viewers, it is of prime importance to guarantee that the perceived visual quality of these videos is still maintained at an acceptable level at the end-user after the production and distribution chain. To achieve this goal, effective image and video quality assessment algorithms are needed and have recently attracted considerable research attention.

Visual quality measurement is a vital yet complex work in many image and video processing applications. IQA can be completed using two types of methods, which are subjective and objective IQA methods. Subjective IQA methods rely on the opinions of a large number of viewers, which makes them expensive to implement and impractical in real applications. Although subjective IQA methods are cumbersome in real applications, they are usually adopted to design a subjective score for each image or video in IQA database, such as the mean-opinion-score (MOS) in each IQA database. Objective IQA methods refer to designing algorithms to automatically predict the visual quality of an image or a video which is consistent with human perception. According to the dependency of reference images or videos, objective IQA methods are usually divided into three types: full-reference (FR), reduced-reference (RR) and no-reference (NR).

FR-IQA and RR-IQA metrics assume that the whole reference signal or partial information of the signal is available, and do a comparison between reference signal and tested signal. Since information about original signal can be used as reference, state-of-the-art FR-IQA methods can achieve a high correlation with human perception. Some state-of-the-art FR-IQA algorithms include IFC [1], VIF [2] and FSIM [3].

NR-I/VQA metrics exploit only the tested signal and have no need of any information about reference signal. Because of this advantage, NR-IQA algorithms have much wider applicability and received a great deal of attention. Previous researchers have attempted to develop distortion-specific NR-IQA algorithms. These algorithms calibrate some specific distortions, such as JPEG [4], JPEG2000 [5], H.264/AVC [6]. Although these methods work well for the specific distortions, it is not easy for them to be generalized to other new distortion types. Thus, these approaches are inferior to the state-of-the-art approaches. Nowadays, many researchers have paid much effort to investigate NSS based general-purpose NR-IQA algorithms. Some successful examples of such kind of NR-IQA approaches include DIIVINE [7], BLIINDS-II [8] and BRISQUE [9]. Compared with the NSS based NR-IQA approach, nowadays, training-based NR-IQA is a new trend. With the development of feature learning methods, training-based NR-IQA approaches learn discriminative features directly from raw image patches without using hand-crafted features. These methods deal with small image patches (such as 32 × 32) and the whole image quality score is the average score of small patches. The representative works about this type of NR-IQA work include CORNIA [10] and CNN NR-IQA [11]. CORNIA aims at training image representation kernels directly from raw image pixels using unsupervised feature learning and CNN NR-IQA integrates feature learning and regression into one optimization process using traditional Convolutional Neural Networks.

Recently, deep learning has gained researchers' attention and achieved great success on various computer vision tasks. Specifically, recent studies have shown that deep CNN significantly improves the performance on various vision tasks, such as object detection, image classification, and segmentation. These accomplishments are attributed to the ability of deep CNN to learn the rich mid-level image representations. Besides, one of CNN's advantages is that it can take raw images as input and incorporate feature learning into the training process. With a deep structure, the CNN can effectively learn complicated mappings while requiring minimal domain knowledge. In this way, there is no need to put so much energy into designing elaborate hand-crafted features. Inspiring from the advancement in deep learning, we raise a question that can we take the advantage of deep CNN to achieve NR-IQA? Instead of using the shearlet transform to extract features in our previous NR-I/VQA works, such as SHANIA [12], SESANIA [13] and SACONVA [14], can we generate image quality score directly from the deep CNN just using the raw images? To address these questions, we propose a deep CNN model that can learn image

978-1-5090-4165-7/16/$31.00 ©2016 IEEE

representations and image quality scores at the same time.

Le Kang et al. are the pioneers to apply CNN to general-purpose NR-IQA [11]. Although they proposed a very meaningful framework and achieved excellent experimental results, there are still some limitations. For example, the CNN they used is the traditional CNN which is somewhat out of date. Besides, the CNN only contains one convolution layer which is too shallow compared with the state-of-the-art deep CNN. In addition, in order to obtain enough labeled training data, they train the network on 32 × 32 patches taken from large images. For training, they assign each patch a quality score as its source image's ground truth score (MOS) because they assume that the training images in the experiments have homogeneous distortions. However, the MOS for each image is obtained by human opinion which is based on the perception of the whole image. The 32 × 32 patch is too small and the given label may not be accurate.

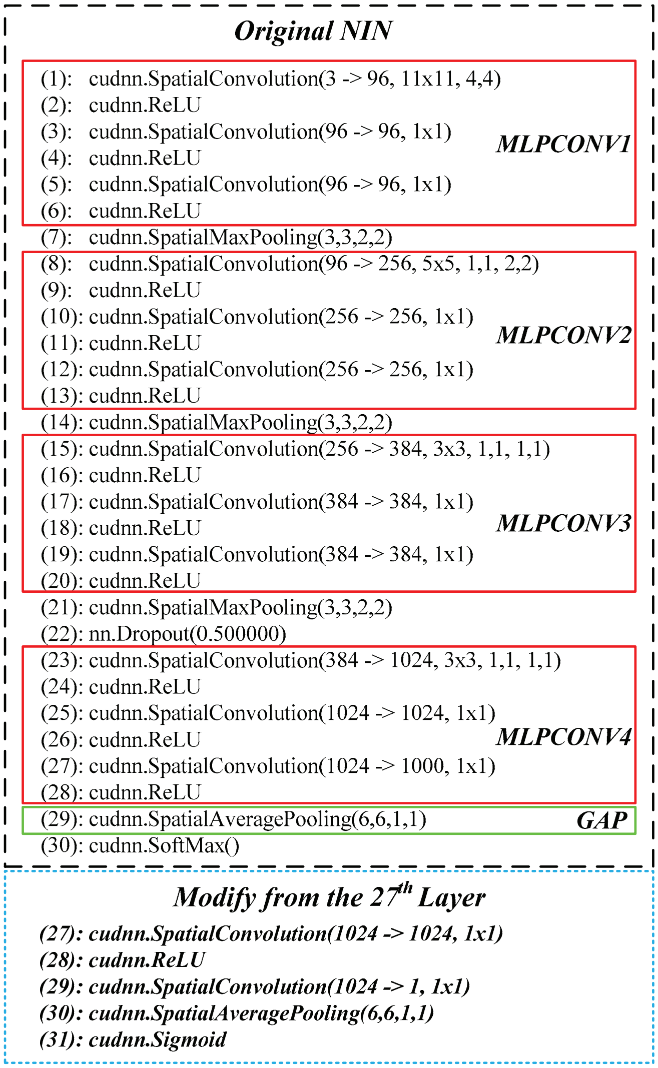
In this paper, we propose to use deep CNN and some state-of-the-art training techniques to deal with the NR-IQA problem and also made a prospect to the future research about NR-IQA. The remainder of this paper is organized as follows. Section II introduces the detailed structure about the deep CNN we used. In section III, experimental results and the analysis of this framework are presented. Finally, conclusion is given in section IV.

II. METHODOGY

Fig. 1 shows the proposed NR-IQA framework which is based on deep CNN. Our method includes three main steps. The first step is the supervised pre-training on the large-scale ImageNet dataset [15]. The second step is the network modification. The third step is fine-tuning the new network for NR-IQA purpose. In the first step, we construct an original deep CNN model. This model is a Network in Network (NIN) model trained on ImageNet dataset which contains more than 1.2 million images categorized into 1000 object classes. Through this pre-training step, we obtain relatively good initial weights which is much better than randomly initialized weights. In the second step, we modify this original model and make it suitable for NR-IQA. We retain the pre-trained NIN from the 1st layer to the 26th layer. Five new layers are concatenated following the 26th layer which is shown in the blue box in Fig. 1. In this way, the modified new network can directly outputs image quality scores. Now, in the new network, layer 1 to layer 26 already have good initial weights. Only layer 27 and layer 29 are randomly initialized and their parameters can be easily tuned by fine-tuning process. In the third step, only a small number of labeled data can make this new network work for NR-IQA purpose. Detailed information about this deep CNN is illustrated in the following sub-sections.

A. Network Architecture

The proposed network consists of 31 layers. Given a color image, we first sample 224 × 224 image patches from the original image, and then perform a global contract normalization in each channel by subtracting the mean image of ImageNet database for each patch. These patches are the input of this network. We use this deep CNN to estimate the quality score for each patch and average the patch scores to obtain a quality estimation for the whole image. Some state-of-the-art structures and training techniques are adopted in this deep CNN such as Multi-Layer Perceptron (MLP) convolution layer, Global Average Pooling (GAP) layer and Dropout. Instead of traditional sigmoid or tanh neurons, the activation function we used in this deep CNN is Rectified Linear Units (ReLU). [15] demonstrated that in a deep CNN that ReLUs enable the network to train several times faster compared to using tanh units. The last layer is a sigmoid function with a one dimensional output that provides the image quality score.



```
(1):   cudnn.SpatialConvolution(3 -> 96, 11x11, 4,4)
(2):   cudnn.ReLU
(3):   cudnn.SpatialConvolution(96 -> 96, 1x1)
(4):   cudnn.ReLU
(5):   cudnn.SpatialConvolution(96 -> 96, 1x1)
(6):   cudnn.ReLU
(7):   cudnn.SpatialMaxPooling(3,3,2,2)
(8):   cudnn.SpatialConvolution(96 -> 256, 5x5, 1,1, 2,2)
(9):   cudnn.ReLU
(10): cudnn.SpatialConvolution(256 -> 256, 1x1)
(11): cudnn.ReLU
(12): cudnn.SpatialConvolution(256 -> 256, 1x1)
(13): cudnn.ReLU
(14): cudnn.SpatialMaxPooling(3,3,2,2)
(15): cudnn.SpatialConvolution(256 -> 384, 3x3, 1,1, 1,1)
(16): cudnn.ReLU
(17): cudnn.SpatialConvolution(384 -> 384, 1x1)
(18): cudnn.ReLU
(19): cudnn.SpatialConvolution(384 -> 384, 1x1)
(20): cudnn.ReLU
(21): cudnn.SpatialMaxPooling(3,3,2,2)
(22): nn.Dropout(0.500000)
(23): cudnn.SpatialConvolution(384 -> 1024, 3x3, 1,1, 1,1)
(24): cudnn.ReLU
(25): cudnn.SpatialConvolution(1024 -> 1024, 1x1)
(26): cudnn.ReLU
(27): cudnn.SpatialConvolution(1024 -> 1000, 1x1)
(28): cudnn.ReLU
(29): cudnn.SpatialAveragePooling(6,6,1,1)
(30): cudnn.SoftMax()
```

```
(27): cudnn.SpatialConvolution(1024 -> 1024, 1x1)
(28): cudnn.ReLU
(29): cudnn.SpatialConvolution(1024 -> 1, 1x1)
(30): cudnn.SpatialAveragePooling(6,6,1,1)
(31): cudnn.Sigmoid
```

Figure 1. The proposed deep CNN based NR-IQA framework.

B. Network In Network

The convolution filter in CNN is a Generalized Linear Model (GLM) for the underlying data patch. In [16], Lin et al. argue that the level of abstraction is low with GLM, which means the learned features by traditional CNN filters is variant to the variants of the same concept. Therefore, a more potent nonlinear function approximator should be used to replace the GLM in order to enhance the abstraction ability of the local model. To achieve this goal, Lin et al. propose to replace the GLM with a "micro network" structure which is a general nonlinear function approximator and instantiate it using MLP.

MLP convolution layers: the mlpconv maps the input local patch to the output feature vector with a MLP consisting of multiple fully connected layers with nonlinear activation functions. The MLP is shared among all local receptive fields. The feature maps are obtained by sliding the MLP over the input in a similar manner as CNN and are then fed into the next layer. The mlpconv layers can be easily implemented using one traditional convolution layer followed by several convolution layers with 1 × 1 convolution kernel and ReLU activation function. The red box in Fig. 1 shows the MLP convolution layers we used in our experiment.

Global average pooling: the traditional fully connected layers are prone to overfitting, thus hampering the generalization ability of the overall network. In contrast, GAP itself is a structural regularizer, which natively prevents overfitting for the overall structure. The green box in Fig. 1 shows the GAP layer we used in our experiment.

The overall structure of the NIN is the stacking of multiple MLP convolution layers, on top of which lie the GAP and the sigmoid layer. The NIN we used contains four mlpconv layers. Within each mlpconv layer, there is a three-layer perceptron.

C. Learning

As discussed previously, three main steps are involved when training this deep CNN. They are pre-training using large-scale ImageNet dataset, network modification and fine-tuning using target dataset. In this section, we will mainly discuss how to conduct the fine-tuning process and testing process.

We resize the original color image into 448 × 448 and fine-tuning our network on 224 × 224 patches taken from large color images. Similar with [11], we assign each patch a normalized quality score as its source image's normalized MOS. In [11], the author takes small patches (32 × 32) as input and gets a much larger number of training samples. Since our network already has good initial weights, we can fine-tune it using a small number of training samples. For example, when conducting the experiments, we randomly select 60% of reference images and their distorted versions as the training set. For LIVE IQA database, there are about 600 training images for each training process. The stride of patch sampling is 112. Thus, each image is sampled into 9 overlapping patches. Therefore, there are about 5400 image patches for each training process. Compared with the large size of our network, the number of training data is relatively small. During the test stage, instead of sampling several image patches and run deep CNN several times. We directly use the 448 × 448 as the input and run the deep CNN only once. The output of the large image is a 8 × 8 feature map which is equal to run the deep CNN on 64 image patches. We average the 8 × 8 feature map for each image to obtain the final quality score.

III. Experiments and Related Analysis

We implemented our deep CNN and conducted the experiments using torch7, which is a scientific computing framework with wide support for machine learning algorithms [17]. Torch7 is easy to use and efficient since it is implemented using LuaJIT which is an easy and fast scripting language. In the first step of training, we use pre-trained original deep CNN model from the Caffe CNN library [18] and export it as torch7 model. LIVE IQA database is used in the experiments. LCC (Linear Correlation Coefficient) and SROCC (Spearman Rank Order Correlation Coefficient) are used as the measurements.

Parameters of the network: the parameters of the network we used is shown in Fig. 1. When training, the batch size is 2, the learning rate is 1e-4, the learning rate decay is 1e-7, the weight decay is 5e-5, the momentum is 0.9, and the max epoch for training is 300.

LIVE IQA database [19]: this IQA database contains 29 high-resolution 24-bits/pixel RGB original images distorted using five types of distortions at different distortion levels. These original images are distorted using the following distortion types: JPEG2000, JPEG, white Gaussian noise in the RGB components, Gaussian blur in the RGB components, and bit errors in JPEG2000 bit stream when transmitted over a simulated fast-fading Rayleigh channel. Besides, MOS and the standard deviation between subjective scores were computed for each image. MOS for LIVE is in the range 0 to 100. Higher MOS indicates higher image quality.

A. Performance Evaluation

As previously mentioned, our deep CNN is implemented using the torch7 framework. With torch7, we are able to easily run the algorithm on a GPU to speed up the process without much optimization. Our experiments are performed on a PC with Intel Core i7-4790 CPU and NVDIA GTX750Ti GPU. We report median LCC and SROCC obtained from 10 train-test iterations where in each iteration we randomly select 60% of reference images and their distorted versions as the training set, 20% as the validation set, and the remaining 20% as the test set. The reason why we just conduct 10 train-test iterations instead of 100 in [15] is that it takes longer time for training process. Although the number of training patch is small, it still needs relatively long time to train since the network we used is deep and complicated and the size of training patch is relatively large. Stochastic Gradient Descent (SGD) is applied to train the network and it takes about 3.5 minutes for one epoch. The max epoch for training is 300. Therefore, it takes about 17.5 hours to complete one training process. If we adopt 100 train-test iterations, we need at least 73 days to complete the experiment which is unacceptable. Compared with training process, testing process is much faster. We need only 50 ms to process one image which makes the real-time application possible.

Table I shows the experimental results on LIVE database. In the table, CNN refers to the algorithm proposed in [11] and DeepCNN refers to the proposed algorithm. For the FR-IQA methods, 80% of the data is used for estimation parameters of a logistic function and 20% is used for testing. For the hand-crafted feature based NR-IQA methods, 80% of reference images and their distorted versions are used for training purpose and 20% as the test set. For training based methods (CNN and DeepCNN), the percentage of training images is reduced to 60%. We can see that the overall performance of our DeepCNN approaches the state-of-the-art CNN. It also outperforms CNN in some specific distortion types. In addition, compared with NR-IQA algorithms using hand-crafted features, we can clearly see the advantages of CNN and DeepCNN which utilize fewer training data and achieve better performance. To see how performance improves, we record the LCC and SROCC every ten epochs for both training patches and testing images. Fig. 2 show the results for one training process. We can see that the performance for both training patches and testing images reveal raising tendency with the increase of the epochs. In Fig. 3, we visualize the averaged outputs from the 29th layer (before GAP) using testing images with different classes. The image class is generated based on Table II and the class number is 6. Since the mean value of these feature maps is directly used as the input of the sigmoid function, we can clear see that the mean feature map of high quality images tends to be larger activations and that of low quality images tends to be smaller activations. This is explicitly enforced by GAP. In Fig. 4, we show the output of the deep CNN on a testing image (Bike) and all of its distorted versions and the image information of Fig. 4 is provided in Table III. We can also see that the output of the deep CNN is only relevant to the MOS and not sensitive to the distortion type, which further demonstrates that the proposed NR-IQA algorithm is general-purpose.

B. *Discussion*

Although the proposed deep CNN already achieves acceptable performance for NR-IQA, there is still a huge potential to improve. For example, the GPU we used is GTX750Ti which contains only 640 CUDA cores and already a little out of date. Better GPU can be used to significantly increase the training process, such as GTX980Ti (2816 CUDA cores) or GTX Titan Z (5760 CUDA cores). Besides, recently multi-GPU architecture is also a new trend to speed up the training process. In addition, with the improvement of the hardware, we can further decrease the patch sampling stride and create more training patches, and average more patch scores when predicting the whole image score. In this way, the performance can be further improved. Furthermore, since we have demonstrated that the NR-IQA problem can be successfully solved by deep CNN, the NR-IQA performance can be further advanced with the help of the rapid developing deep learning techniques.

IV. Conclusion

In this paper, we have developed a general-purpose NR-IQA algorithm based on deep CNN. Three steps are included when training this network, which are supervised pre-training on the large-scale ImageNet dataset, the network modification and fine-tuning for NR-IQA purpose. Through this algorithm, the feature learning and regression are combined as a complete optimization process, which frees us from designing elaborated hand-crafted features. This new algorithm generates image quality predictions well correlated with human perception, and achieves acceptable performance on standard IQA dataset. In addition, we also discussed how to further improve the performance of the proposed algorithm.

Acknowledgment

The work described in this paper was substantially supported by a grant from the City University of Hong Kong, Kowloon, Hong Kong with Project number of 7004058.

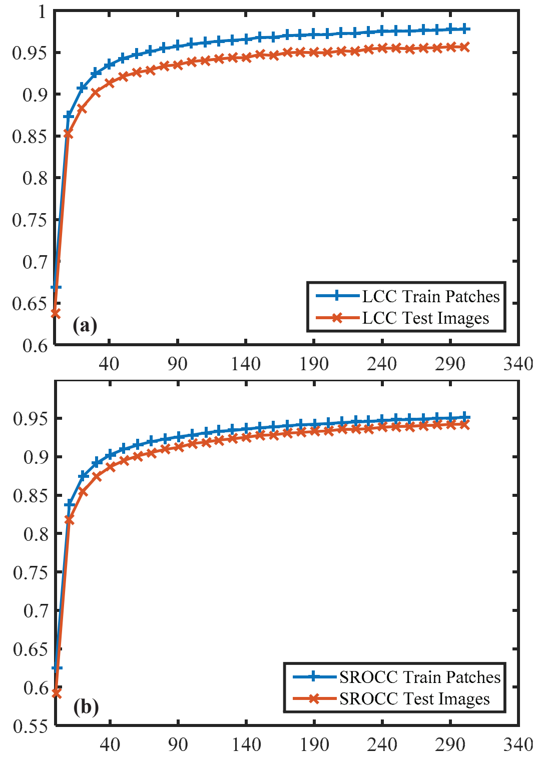


Figure 2. LCC and SROCC versus epochs for both training patches and testing images.

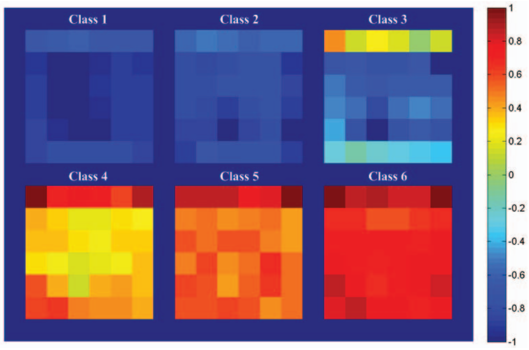


Figure 3. Averaged outputs from the 29th layer (before GAP) using testing images with different classes. The image class is generated based on Table II and the class number is 6. Since for the training patch, the output size of the 29th layer is 6 × 6, we further sampled the testing image output of this layer (13 × 13) into 64 patches and calculate the average of these patches.

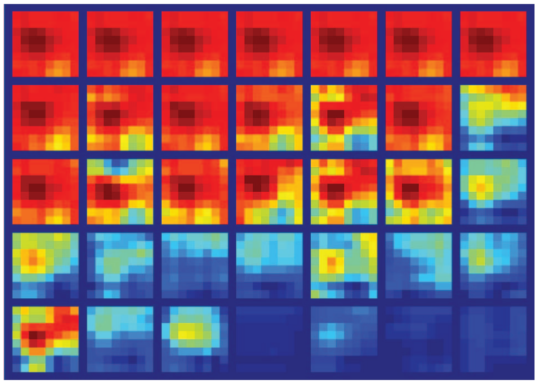


Figure 4. The output of the deep CNN on a testing image (Bike) and all of its distorted versions.

References

[1] Sheikh, Hamid R., Alan C. Bovik, and Gustavo De Veciana. "An information fidelity criterion for image quality assessment using natural scene statistics." Image Processing, IEEE Transactions on 14, no. 12 (2005): 2117-2128.

[2] Sheikh, Hamid R., and Alan C. Bovik. "A visual information fidelity approach to video quality assessment." In The First International Workshop on Video Processing and Quality Metrics for Consumer Electronics, pp. 23-25. 2005.

[3] Zhang, Lin, D. Zhang, and Xuanqin Mou. "FSIM: a feature similarity index for image quality assessment." Image Processing, IEEE Transactions on 20, no. 8 (2011): 2378-2386.

[4] Wang, Zhou, Hamid R. Sheikh, and Alan C. Bovik. "No-reference perceptual quality assessment of JPEG compressed images." In Image Processing. 2002. Proceedings. 2002 International Conference on, vol. 1, pp. I-477. IEEE, 2002.

[5] Sheikh, Hamid R., Alan C. Bovik, and Lawrence Cormack. "No-reference quality assessment using natural scene statistics: JPEG2000." Image Processing, IEEE Transactions on 14, no. 11 (2005): 1918-1927.

[6] Brandão, Tomás, and Maria Paula Queluz. "No-reference quality assessment of H. 264/AVC encoded video." Circuits and Systems for Video Technology, IEEE Transactions on 20.11 (2010): 1437-1447.

[7] Moorthy, Anush Krishna, and Alan Conrad Bovik. "Blind image quality assessment: From natural scene statistics to perceptual quality." Image Processing, IEEE Transactions on 20, no. 12 (2011): 3350-3364.

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

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

[10] Ye, Peng, Jayant Kumar, Le Kang, and David Doermann. "Unsupervised feature learning framework for no-reference image quality assessment." In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1098-1105. IEEE, 2012.

[11] Kang, L., Ye, P., Li, Y., and Doermann, D. Convolutional neural networks for no-reference image quality assessment. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1733-1740. IEEE, 2014.

[12] Li, Yuming, et al. "No-reference image quality assessment using statistical characterization in the shearlet domain." Signal Processing: Image Communication 29.7 (2014): 748-759.

[13] Li, Yuming, et al. "No-reference image quality assessment with shearlet transform and deep neural networks." Neurocomputing 154 (2015): 94-109.

[14] Y. Li, L. M. Po, X. Xu, L. Feng, F. Yuan, C. H. Cheung, K. W. Cheung, "No-reference video quality assessment with 3D shearlet transform and convolutional neural networks," IEEE Trans. Circuits Syst. Video Technol., pp.1−13, 2015.

[15] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. Vol. 1, page 4, 2012.

[16] Lin Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[17] Ronan Collobert, Clement Farabet, Koray Kavukcuoglu, and Soumith Chintala. "Torch: a scientific computing framework for LuaJIT." http://torch.ch/

[18] Berkeley Vision and Learning Center. "Caffe: deep learning framework by the BVLC." http://caffe.berkeleyvision.org/

[19] Sheikh, Hamid R., Zhou Wang, Alan C. Bovik, and L. K. Cormack. "Image and video quality assessment research at LIVE." (2003). http://live.ece.utexas.edu/research/quality/.

TABLE I. Median LCC and SROCC Correlations on the LIVE IQA Database. (*Italicized* Algorithms are NR-IQA Algorithms.)

	LCC							**SROCC**					
	JP2K	**JPEG**	**GWN**	**GB**	**FF**	**ALL**		**JP2K**	**JPEG**	**GWN**	**GB**	**FF**	**ALL**
PSNR	0.8669	0.8351	0.9516	0.8268	0.8665	0.8069		0.8395	0.8088	0.8838	0.8309	0.8348	0.8069
SSIM	0.9469	0.9097	0.9754	0.9077	0.9092	0.8002		0.9301	0.9712	0.9604	0.9445	0.9723	0.9278
VIF	0.9447	0.9692	0.9766	0.9702	***0.9754***	***0.9574***		0.9162	0.9500	0.9576	0.9709	0.9721	0.9541
FSIM	0.9034	0.7711	0.8912	0.8838	0.8316	0.7810		0.9370	0.9721	0.9574	***0.9804***	***0.9741***	0.9548
BIQI	0.8086	0.9011	0.9538	0.8293	0.7328	0.8205		0.7995	0.8914	0.9510	0.8463	0.7067	0.8195
DIIVINE	0.9220	0.9210	***0.9880***	0.9230	0.8680	0.9170		0.9319	0.9483	***0.9821***	0.9210	0.8714	0.9116
BLIINDS-II	0.9386	0.9426	0.9635	0.8994	0.8790	0.9164		0.9323	0.9331	0.9463	0.8912	0.8519	0.9124
BRISQUE	0.9229	0.9734	0.9851	0.9506	0.9030	0.9424		0.9139	0.9647	0.9786	0.9511	0.8768	0.9395
SHANIA	0.9135	0.9380	0.9731	0.9790	0.9413	0.9412		0.8611	0.8918	0.9582	0.9674	0.9169	0.9033
SESANIA	0.9537	0.9732	0.9806	0.9749	0.9195	0.9476		0.8862	0.9293	0.9309	0.9410	0.8807	0.9340
CNN	0.953	***0.981***	0.984	0.953	0.933	0.953		***0.952***	***0.977***	0.978	0.962	0.908	***0.956***
DeepCNN	***0.973***	0.955	0.981	***0.984***	0.955	0.956		0.945	0.941	0.964	0.969	0.907	0.935

TABLE II. The Relationship between Image MOS and Its Label.

MOS	**< 35**	**35-44**	**45-64**	**65-74**	**75-80**	**> 80**
Class	1	2	3	4	5	6

TABLE III. Image Information of Figure 4. (*Distortion_MOS*)

ORI_85.00	FF_85.00	GB_85.00	GWN_85.00	JPG_85.00	JPG2K_80.14	JPG2K_77.92
GWN_77.77	GB_77.76	FF_77.45	JPG2K_77.15	GWN_74.71	JPG_74.56	JPG2K_73.95
GB_70.05	JPG_69.60	GB_69.52	FF_64.41	GWN_62.01	JPG2K_60.25	JPG2K_56.25
FF_55.32	FF_53.31	JPG_50.94	GWN_50.64	GWN_45.54	JPG_44.06	JPG_43.98
GB_43.28	JPG_43.18	JPG_36.77	GWN_28.13	JPG2K_28.09	FF_25.97	JPG2K_25.72