Unsupervised Blind Image Quality Assessment

Rajeev Bhatt Ambati

April 4, 2018

Abstract

Image quality assessment (IQA) has been a long standing problem in computer vision largely because of the intrinsic hardness to model natural scene statistics. In the recent times, the success of Convolutional Neural Networks (CNN) for computer vision tasks has driven many to use them to extract good representation of images. In such an attempt, most of the methods proposed in the IQA literature also train an end-to-end model using a CNN that will be regressed over the subjective ratings. We should also note that unsupervised learning is the next frontier in machine learning that is worth solving as the annotations are very expensive. In this project, I would like to explore three ways of doing it: Using SVM-loss on extracted features, Capturing the aesthetic image manifold similar to a Variational Autoencoder (VAE) and using the hierarchical abstraction nature of a CNN.

Contents

1 Introduction **2**
1.1 Related Work . . . 2
1.2 Large-Scale Database for Aesthetic Visual Analysis (AVA) [1] . . . 3
1.3 Tampere Image Database 2013 (TID 2013) [2] . . . 3
1.4 Idea . . . 3

2 Architectures **3**
2.1 Pre-trained CNN features . . . 4
2.2 Mapping encoded representations . . . 5

3 Evaluation **6**

4 Results **6**
4.1 Pre-trained CNN features. . . . 6

5 Conclusion **9**

1 Introduction

There are several computer vision tasks in which we often compare the outputs of a machine learning model with the ground truth. The examples include, semantic segmentation, super resolution, image denoising, image compression and several other robotics applications. The usage could be to use a quality assessment metric as a loss function or simply for evaluating a machine learning algorithm. Although many methods for the above tasks complain on the ineffectiveness of SSIM [3] in comparing images, a modified version of SSIM thats suitable to the problem is used. This is simply because a computational graph usually need a differentiable and computationally effective loss function. A deeplearning model can easily do this because it will just be a backward propagation through the network. Most recently, Many methods in the IQA literature use CNNs to extract good representations of images and typically fine-tune the network by regressing over the quality ratings. Since annotated data is expensive, its worth posing the IQA as an unsupervised problem and explore the ways of solving it.

1.1 Related Work

Recent success [4] show that applying machine learning models on high-level features extracted from a CNN can achieve state-of-the-art performance. This could be one of the motivation to use replace the hand-crafted features used in [5] with the learned features of a CNN. In [4], input patches of size 32 x 32 are extracted from an image and fed to a shallow network consisting of a convolutional layer and two fully connected layers. But from [6], [7], we can infer that individual patches may capture distortion but aesthetics depend on the whole image. For *e.g.* a distorted week textured patch may have a low distortion score but, it will have a high aesthetic score than a distorted highly textured patch.

In [6] a Siamese CNN network is trained to produce 0 or 1 for low and high quality images respectively. Most often input images fed to a deep learning model are resized. But on doing so, it alters image composition and results in loss of fine grained details which are critical for aesthetic assessment. [8] uses adaptive spatial pooling for allowing multiple input sizes and a CNN is trained with a binary crossentropy loss. So, [7] proposes a multi-scale approach in which five patches of size 224 x 224 x 3 are extracted from the input image and fed to 5 VGG networks [9] of shared weights. This is very expensive to be used as a loss function.

It is observed in [10] that merely using a regression loss performs poorly on images with close quality scores. To tackle this problem, they have used a ranking loss between prediction of two images fed to a Siamese AlexNet [11]. They also tried to extract separate attribute and aesthetic from a network that shares the same low-level features . These are finally weighted with corresponding scores to obtain a final quality score. [12] argue that quality score is a subjective problem and therefore regression over mean quality score is not a good strategy. They instead obtain a histogram of ratings for different users from a CNN and compute a weighted score over 1 - 10 ratings. This is the state-of-the-art in Blind Image Quality Assessment (BIQA).

In the literature of Image Quality Assessment, the following two datasets are used for evaluation:

1.2 Large-Scale Database for Aesthetic Visual Analysis (AVA) [1]

The AVA dataset contains about 255,000 images covering a wide variety of subjects on 1,447 challenges. Each image is associated with a single challenge and has three types of annotations from which aesthetic annotations are used for the problem. This aesthetic score is scored by an average of 200 people in response to photographic contests. The image ratings are in between 1 and 10 in which the high score implies a good quality image.

1.3 Tampere Image Database 2013 (TID 2013) [2]

TID2013 is originally intended for full-reference image visual quality assessment metrics. It allows to estimate how a given metric corresponds to mean human perception. The TID2013 contains 25 reference images with 5 levels of distortion for 24 distortion types. For a total of 971 observers, two randomly chosen images are shown at a time from which the subject has to chose the best quality image. Doing this way, the good quality image gets 1 points and the other gets 0 points. A Mean Opinion Score (MOS) of an image is calculated by taking the average of all points it obtained.

1.4 Idea

Most of the approaches to BIQA problem as discussed above uses a single image thats fed to a CNN which is regressed over the subjective score. At training time, the model will have no information about an aesthetically appealing image or a bad quality image for that matter. This information is provided merely by a reward signal which is the loss function that is used to regress over scores. I argue that a more efficient way is to give the network a prior of a good or bad quality image since the quality estimation is relative. Even in the TID2013 experiments, the subject is given a prior of a bad quality image by showing two images out of which a good quality image is chosen.

Consider for *e.g.* three images I_1, I_2, I_3 arranged in increasing order of quality. Ideally if all the combinations of two images are shown to a user the images would score 0, 0.5, 1 respectively. This doesn't depend on the how bad or how good one image is to another. This reinforces earlier argument to give a prior of a good/bad quality image to the model. On capturing the extent of degradation, a more efficient way is to capture an aesthetic manifold and rate images according to its distance from this manifold. This solves two problems: one is unsupervised learning objective and also rating of the extent of degradation. I would like to explore two ways of doing it in this project.

2 Architectures

The idea of capturing the aesthetic manifold can be done in the following two ways:

2.1 Pre-trained CNN features

Let $f(I, \theta)$ be the encoded representation from the higher layer of a CNN for an input image I, where θ are the parameters of the CNN. For 4 images I_1, I_2, I_3, I_4 in the decreasing order of quality, a quality score metric $Q(I)$ should satisfy

$$Q(I_1) > Q(I_2) > Q(I_3) > Q(I_4) \tag{1}$$

Since this is an unsupervised learning setting, we can either set the quality function $Q(I)$ to be simply the L2-norm of the encoded representation of the image and scale it to 10 or directly calculate a score from the network as follows:

$$Q(I) = \|f(I, \theta)_{512\times1}\|_2 \times \frac{10}{512} \tag{2}$$

$$Q(I) = f(I, \theta) \tag{3}$$

Ideally, we would want our quality score $Q(I)$ to not follow equation 1 but to satisfy is by a margin say ϵ.

$$Q(I_1) > Q(I_2) + \epsilon > Q(I_3) + 2\epsilon > Q(I_4) + 3\epsilon \tag{4}$$

I have explored the following two loss functions that satisfy these equations:

$$\begin{aligned} L = \max(0, Q(I_2) + \epsilon - Q(I_1)) + \\ \max(0, Q(I_3) + 2\epsilon - Q(I_1)) + \\ \max(0, Q(I_4) + 3\epsilon - Q(I_1)) \end{aligned} \tag{5}$$

$$\begin{aligned} L = |10.0 - Q(I_{ref})| + \max(0, Q(I_1) + \epsilon - Q(I_{ref})) + \\ \max(0, Q(I_2) + 2\epsilon - Q(I_{ref})) + \\ \max(0, Q(I_3) + 3\epsilon - Q(I_{ref})) \end{aligned} \tag{6}$$

Note its similarity to loss function in max-margin classifiers like Support Vector Machine (SVM). A similar loss function called triplet loss is used in [13] and achieved good performance in face recognition. The complete architecture used for training is shown in Fig. 1. A clean image namely I_{clean} and 3 other images I_1, I_2, I_3 with different levels of distortion are passed through a pre-trained AlexNet [11] to obtain 1 × 4096 encodings. On top of the pre-trained AlexNet, the obtained encodings are fed to a two layer feed forward neural network consisting of 1024 and 512 units respectively. The output is a single neuron predicting the quality score. All the added layers have ReLU (rectified linear unit) nonlinearity.

$$\begin{aligned} x_1 &= hW_1 + b_1 \\ z_1 &= \max(0, x_1) \\ x_2 &= z_1 W_2 + b_2 \\ \text{output} &= \sigma(x_2) \end{aligned} \tag{7}$$

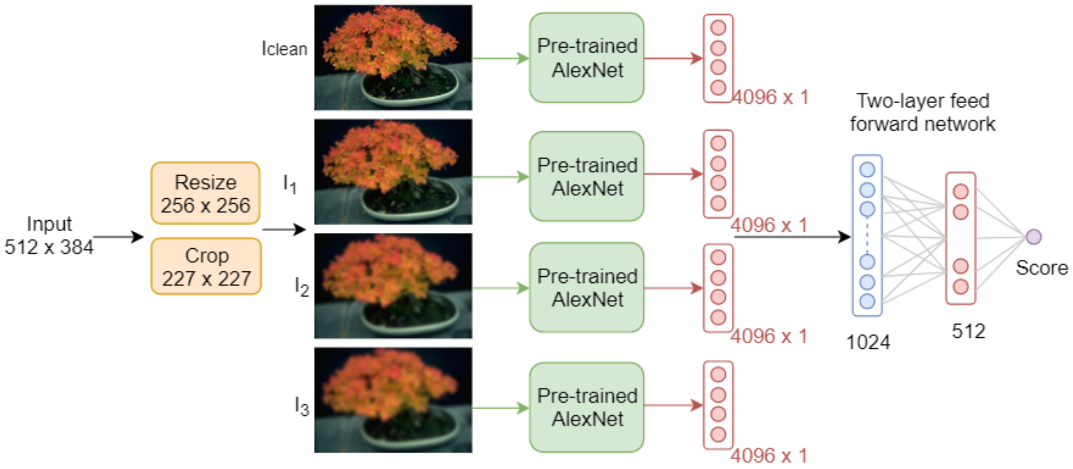


Figure 1: 4 Images of different quality are fed to a pre-trained CNN to obtain $h \times 1$ encodings. This network is fine-tuned with an SVM loss function defined on these encodings thats given in eq. 5.

$$\begin{aligned} x_1 &= hW_1 + b_1 \\ z_1 &= \max(0, x_1) \\ x_2 &= z_1 W_2 + b_2 \\ z_2 &= \max(0, x_2) \\ Q &= z_2 W_3 + b_3 \end{aligned} \tag{8}$$

The above equations concretely describe the functionality of the layers used in the architecture. The eqs. 7, 8 correspond to the losses described in eqs. 5, 6 respectively. Where, $h \in \mathbb{R}^{1\times 4096}$ is the encoding obtained from the pre-trained AlexNet, $W_1 \in \mathbb{R}^{4096\times 1024}, b_1 \in \mathbb{R}^{1\times 1024}, W_2 \in \mathbb{R}^{1024\times 512}, b_2 \in \mathbb{R}^{1\times 512}, W_3 \in \mathbb{R}^{512\times 1}, b_2 \in \mathbb{R}^{1x1}$ are the weights and biases of the first, second and output layer respectively. For the loss function described in eq. 5, $f(I, \theta)$ is the output of the hidden unit with 512 units and its norm is considered as the quality score. Whereas, for the loss function shown in eq. 6, $f(I, \theta)$ is directly the output of the network Q that is constrained to lie in between 0 and 10. The performance using both the loss functions and also a comparison is discussed in the results section.

2.2 Mapping encoded representations

The second idea of capturing the aesthetic manifold is inspired from variational autoencoder (VAE) which is the mapping of latent vector to a Gaussian distribution. The prior of good and bad quality image can be introduced into the network by mapping the encoded representation of an image to lie on a straight line joining the representations of good and bad quality image. A block diagram of this method is shown in Fig. 2. Consider an image I for which the corresponding good and bad quality reference images are I_{good} and I_{bad} respectively.

Let the functions $f_{enc}(I, \theta_{enc}), f_{dec}(h, \theta_{dec}), f_q(I, \theta_q)$ represent an encoder which encodes the image I to a latent space, decoder which reconstructs the image I and a quality estimator whose value is between 0 and 1. Where, $\theta_{enc}, \theta_{dec}, \theta_q$ are the parameters of the encoder, decoder and quality network respectively. Let h_{good}, h_{bad}, h_I be the encoded representations of I_{good}, I_{bad}, I respectively. Since we are introducing an architectural prior of good and bad reference images by forcing h_I to lie on the straight line joining h_{good} and h_{bad}, it can be expressed as follows:

$$h_{good} = f_{enc}(I_{good}, \theta_{enc}) \tag{9}$$

$$h_{bad} = f_{enc}(I_{bad}, \theta_{enc}) \tag{10}$$

$$h_I = Qh_{good} + (1 - Q)h_{bad} \tag{11}$$

Where $Q = f_q(I, \theta_q), 0 \leq Q \leq 1$, the quality score of image I. Note that for $Q = 1, h_I = h_{good}$ signifies a good quality image, for $Q = 0, h_I = h_{bad}$ signifies a bad quality image and for the rest $0 < Q < 1$ implies a quality score of how good and bad the image is. The loss function therefore will look like this:

$$I_{recon} = f_{dec}(h_I, \theta_{dec}) \tag{12}$$

$$L = \|I - I_{recon}\|_2 \tag{13}$$

Since h_I is not directly obtained from the encoder and instead constructed from h_{good} and h_{bad}, the quality network should capture the Q value accurately in order for the decoder to reconstruct I. For the testing phase, we only need the quality network $f_q(I, \theta_q)$. The networks $f_{enc}(I, \theta_{enc}), f_{dec}(h, \theta_{dec}), f_q(I, \theta_q)$ will ideally be a CNN, the best architectures of them have to explored as per training requirements. At test time a forward pass through the Quality estimator network will give the quality score of an image.

3 Evaluation

In the classic machine learning approaches to BIQA, the proposed methods were evaluated by taking a correlation between predicted quality scores and ground truth scores. Since the scales of predicted and ground truth score maybe different, Spearman's Rank-Order Correlation Coefficient (SRCC) is used. Along with SRCC, Linear Correlation Coefficient (LCC) is also reported. In the recent deeplearning approaches to BIQA, very few methods have reported SRCC on AVA dataset and instead a classification accuracy for two classes low quality and high quality is reported.

4 Results

4.1 Pre-trained CNN features.

The architecture described above is trained and tested on TID 2013 dataset. The whole data is randomly shuffled and first 80% data is used for training and validation while the

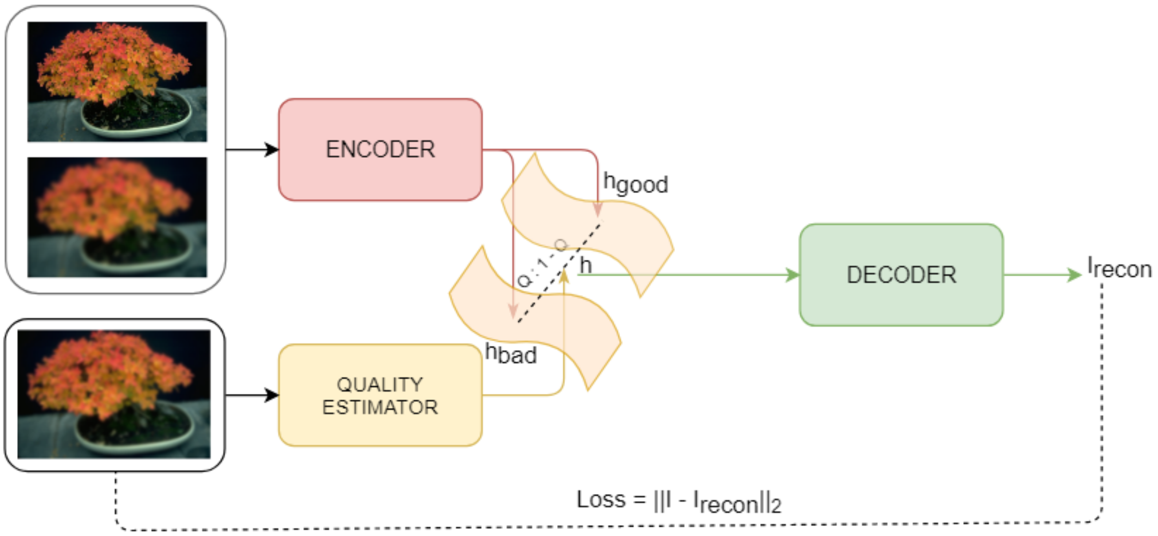


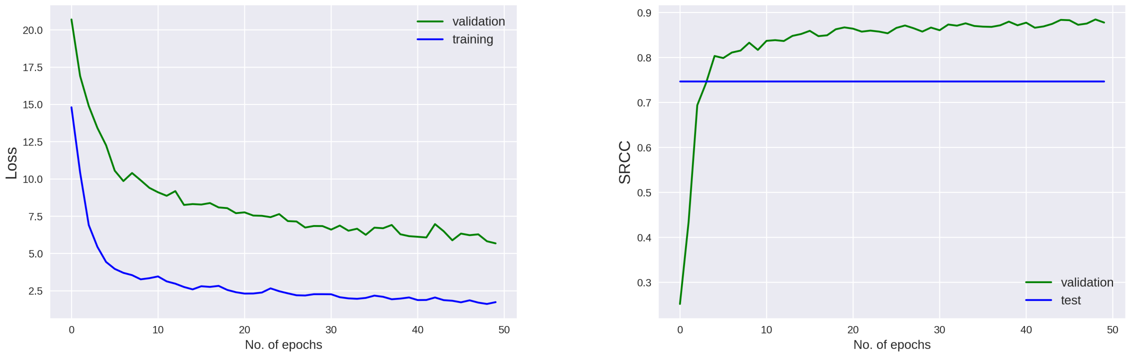
Figure 2: Good, bad image encodings are obtained from the encoder. Using these encodings and quality Q, encoding of the image I is interpolated. This architectural prior forces the quality estimator to learn a score Q that will help the decoder in reconstructing the image I.

last 20% is used for testing. Let $I_{d1}, I_{d2}, I_{d3}, I_{d4}, I_{d5}$ be the 5 levels of distorted images for a reference image I_{ref}. The first 3 distorted images I_{d1}, I_{d2}, I_{d3} are used for training and I_{d1}, I_{d4}, I_{d5} are used for validation. Since validation set has new images, we can also check the generalization performance of the model. All the 5 levels of distortion are considered for testing. Overall, this leads to 480 examples each for training set, validation set and 120 examples for test set as shown below:

$$\begin{aligned} X_{train} &= [I^i_{ref}, I^i_{d1}, I^i_{d2}, I^i_{d3}] \\ X_{val} &= [I^i_{ref}, I^i_{d1}, I^i_{d4}, I^i_{d5}] \\ X_{test} &= [I^j_{ref}; I^j_{d1}; I^j_{d2}; I^j_{d3}; I^j_{d4}; I^j_{d5}] \end{aligned}$$

Where $i = 1, 2, ..., 480$ and $j = 481, 481, ..., 600$. With the score function defined as in eq. 2, the scores obtained were small in magnitude and not very interpretable. This is because the sigmoid function squashes the values between 0 and 1 non-uniformly irrespective of its input. Instead, the quality score thats directly calculated by adding a fully connected layer is very interpretable.

As a preprocessing step, the 512 × 384 input image is resized to 256 × 256 and a 227 × 227 crop is extracted out of it. This is just to preserve the composition of the image, this procedure is followed in the deeplearning literature and almost always shown good performance. A random horizontal flipping is also used to prevent the model from overfitting as the size of dataset if very small. Fig 3 shows the graduate decrease in loss function and the increase in SRCC score as the training proceeds.



(a) (b)

Figure 3: Performance of the model: First plot shows the change of training, validation loss with the epochs. Second plot shows the SRCC (Spearman's rank correlation coefficient) on the validation set throughout the training. The SRCC on the test set is evaluated only after the training is completed.

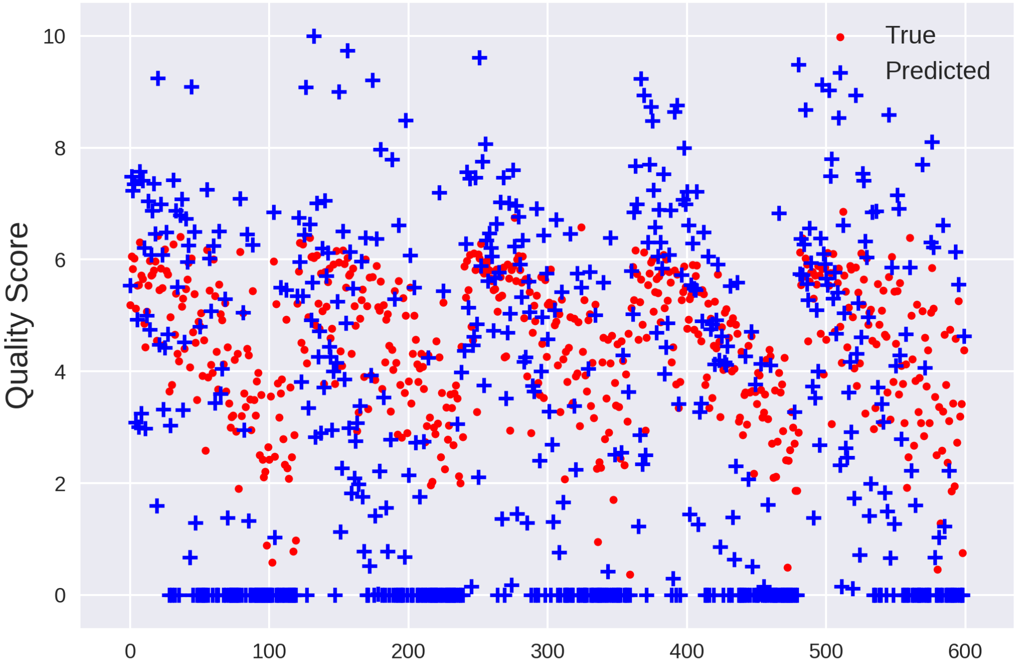


Figure 4: Scatter plot of the resulted predictions versus the ground truth labels. Points in blue correspond to the predicted scores and the ones in red correspond to the true quality scores.

Methods	SRCC	LCC
NIMA (MobileNet) [12]	0.698	0.782
NIMA (Inception-v2) [12]	0.750	0.827
Moorthy et al. [14]	0.88	0.89
NIMA (VGG 16) [12]	0.944	0.941
Bianco et al. [5]	0.96	0.96
Using pretrained embeddings	0.75	0.70

Table 1: The performance of the first architecture is compared to the existing traditional machine learning and deep learning approaches. The 2nd and 3rd columns correspond to SRCC (Spearman's rank correlation coefficient) and LCC (Linear correlation coefficient) evaluated on TID2013 dataset. This methods outperforms a variant of the NIMA thats shown in blue.

duration	work
2 weeks	Approach I
2 weeks	Approach II
2 weeks	Spear day and writeup

Table 2: Timeline for the project

5 Conclusion

The results obtained so far on TID2013 dataset are shown in Table. 1 and it is able to outperform a variant of the state-of-the-art. A scatter plot of the true scores versus the predicted scores is shown in Fig. 4. We can clearly see that for all the images, the predicted scores are in proper range with the true scores between 2 - 4 and 6 - 8. But the predictions don't exactly match the true scores. This is not a surprise as the loss function only enforces the quality score to be in a specific range. I think this problem can be tackled with the second architecture that maps the encoding to lie on the straight line joining the good image manifold and bad image manifold as the precise position matters for the decoder to reconstruct the image. I believe the performance of the first architecture is comparable to the state-of-the-art given that this follows an unsupervised method whereas all the others use quality scores. I feel that this gives enough motivation to pursue a solution for the BIQA problem in an unsupervised learning setting.

References

[1] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2408–2415. 1, 3

[2] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. Jay Kuo, "Image database tid2013," *Image Commun.*, vol. 30, no. C, pp. 57–77, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.image.2014.10.009 1, 3

[3] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004. 2

[4] L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for no-reference image quality assessment,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1733–1740. 2

[5] S. Bianco, L. Celona, P. Napoletano, and R. Schettini, “On the use of deep learning for blind image quality assessment,” *CoRR*, vol. abs/1602.05531, 2016. [Online]. Available: http://arxiv.org/abs/1602.05531 2, 9

[6] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, “Rating image aesthetics using deep learning,” *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2021–2034, Nov 2015. 2

[7] S. Ma, J. Liu, and C. W. Chen, “A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment,” *CoRR*, vol. abs/1704.00248, 2017. [Online]. Available: http://arxiv.org/abs/1704.00248 2

[8] L. Mai, H. Jin, and F. Liu, “Composition-preserving deep photo aesthetics assessment,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 497–506. 2

[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556 2

[10] S. Kong, X. Shen, Z. Lin, R. Mech, and C. C. Fowlkes, “Photo aesthetics ranking network with attributes and content adaptation,” *CoRR*, vol. abs/1606.01621, 2016. [Online]. Available: http://arxiv.org/abs/1606.01621 2

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257 2, 4

[12] H. T. Esfandarani and P. Milanfar, “NIMA: neural image assessment,” *CoRR*, vol. abs/1709.05424, 2017. [Online]. Available: http://arxiv.org/abs/1709.05424 2, 9

[13] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *CoRR*, vol. abs/1503.03832, 2015. [Online]. Available: http://arxiv.org/abs/1503.03832 4

[14] A. K. Moorthy and A. C. Bovik, “Blind image quality assessment: From natural scene statistics to perceptual quality,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, Dec 2011. 9