https://doi.org/10.2352/ISSN.2470-1173.2018.09.IRIACV-202

No-Reference Utility Estimation with a Convolutional Neural Network

Edward T. Scott; Northeastern University; Boston, Massachussetts, USA
Sheila S. Hemami; Draper; Cambridge, MA, USA

Abstract

Traditional quality estimators evaluate an image's resemblance to a reference image. However, quality estimators are not well suited to the similar but somewhat different task of utility estimation, where an image is judged instead by how useful it would be in terms of extracting information about image content. While full-reference image utility metrics have been developed which outperform quality estimators for the utility-prediction task, assuming the existence of a high-quality reference image is not always realistic. The Oxford Visual Geometry Group's (VGG) deep convolutional neural network (CNN) [1], designed for object recognition, is modified and adapted to the task of utility estimation. This network achieves no-reference utility estimation performance near the full-reference state of the art, with a Pearson correlation of 0.946 with subjective utility scores of the CU-Nantes database and root mean square error of 12.3. Other no-reference techniques adapted from the quality domain yield inferior performance. The CNN also generalizes better to distortion types outside of the training set, and is easily updated to include new types of distortion. Early stages of the network apply transformations similar to those of previously developed full-reference utility estimation algorithms.

Introduction

The concept of image utility was developed by Rouse et al. [2], and is different from the related concept of image quality. The goal of image quality estimation is to accurately estimate an image's perceptual similarity to a reference image. However, humans are adept at looking “through” visible distortions to extract information about image content. An image's *utility* refers to the success of the information extraction process, and effectively measures the potential of an image to communicate visual information. Additionally, images of low quality can still be quite useful to an observer, supporting the independence of quality and utility. For example, firefighters may use thermal imagery to assess risk and devise a plan of action before entering a burning building, and surveillance video can be of low quality yet still communicate necessary information to law enforcement [3, 4].

Image utility estimation has not been as active an area of research as image quality estimation, and the differences between human observers' perception of utility and quality are still being explored. Rouse et al. developed the CU-Nantes image utility database, consisting of a number of reference images and distorted versions of those references along with associated subjective utility and quality ratings, to aid utility estimation research [2]. Comparing utility and quality scores of the same set of images, it was observed that image pairs may be of the same quality and different utility, or vice versa, indicating a weak relationship between quality and utility, and supporting the development of algorithms designed to estimate utility. The first purpose-built utility estimator, Natural Image Contour Evaluation (NICE), was developed using this database [5]. Measuring the disruption to image contours, NICE estimates perceived image utility better than many quality estimation algorithms.

To overcome discrepancies between the multi-scale integrative nature of the human visual system (HVS) and the single scale structure of NICE, Multi-Scale Difference of Gaussian Utility (MS-DGU) was developed [6]. MS-DGU attempts to emulate aspects of the HVS's multi-scale processing, and produces utility estimates more strongly correlated with subjective ratings than NICE, with approximately 25% lower root mean square error (RMSE). However, both NICE and MS-DGU are *full-reference* algorithms, meaning in order to estimate an image's utility, an undistorted reference version of that image is required. This requirement is limiting, and in many applications could preclude an algorithm's use.

There has been a great deal of effort in recent years to develop accurate *no-reference* quality estimators, which require no additional information to estimate the quality of an input image. This paper represents the first step of a similar effort to develop no-reference utility estimators. A deep convolutional neural network (CNN) architecture originally developed by the Visual Geometry Group (VGG) at Oxford [1] is adapted to estimate image utility. Its performance is compared to several no-reference quality estimators applied to the utility estimation problem. The CNN outperforms other no-reference estimators, providing no-reference utility estimates with accuracy between that of NICE and MS-DGU.

The paper is organized as follows: previous work in both utility and quality estimation is reviewed, and the architecture of the neural network developed and applied in this work presented. The utility estimation performance of the network and other relevant algorithms is then evaluated, followed by concluding comments.

Related Work

This section briefly summarizes previously developed utility estimation and no-reference quality estimation algorithms.

Utility Estimation

Two purpose-built utility estimation algorithms have been previously proposed. Natural Image Contour Evaluation (NICE) was first proposed in 2009 by Rouse et al. [5]. NICE is based upon the hypothesis that image utility is a function of observers' ability to recognize objects, and that this ability is directly related to the degradation of image contours. As such, NICE operates by

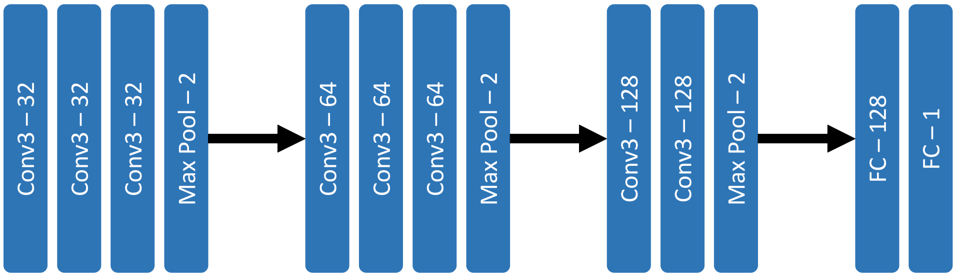


Figure 1: Network Architecture. Layer names follow the convention of [1]: for example, a Conv3-32 layer consists of 32 3x3 convolution kernels for each of its inputs, and an FC-128 layer contains 128 fully connected neurons.

comparing contours between reference and test images at a single scale. Contours are identified by the dilated output of an edge detector, and predicted utility is a function of the Hamming distance between those test and reference contours.

In contrast, Multi-Scale Difference of Gaussian Utility (MS-DGU) operates across multiple scales, and is based on the hypothesis that disruption to coarse image structures impairs the ability of the human visual system to build object representations [6]. It decomposes test and reference images by passing them through a difference of Gaussian (DoG) bandpass filter bank, and compares the location of extrema in the decompositions. The use of DoG filters as opposed to a traditional edge detector results in a focus on corners and blobs, as compared to the edge-based approach of NICE. MS-DGU outperforms all other estimators for both utility and quality when evaluated on the CU-Nantes dataset.

No-Reference Image Quality Estimation

While both utility estimators described above require undistorted reference images, many algorithms for no-reference image quality estimation (NR-IQA) have been proposed. These typically consist of a feature extraction operation, followed by the application of a learned model to relate those features to quality labels. Most algorithms employ perceptually based features, designed to capture natural scene statistics (NSS) properties of images. A representative sampling of NR-IQA algorithms was chosen for application to the CU-Nantes utility database.

The blind/referenceless image spatial quality evaluator (BRISQUE) applies support vector regression (SVR) to luminance coefficients of mean-subtracted and divisively normalized images [7]. The Natural Image Quality Estimator extends BRISQUE, modeling the distribution of BRISQUE coefficients in undistorted natural images and predicting quality by evaluating the degree to which distorted images deviate from the model [8]. NIQE is less sensitive to types of distortion not present in the training set than BRISQUE.

Liu et al. utilize statistics of gradient features, then feed those statistics into an Adaboosting neural network with two hidden layers [9]. This approach yields the highest demonstrated performance among algorithms based on “hand-crafted” features when evaluated on the LIVE database.

Recently, techniques have been proposed which learn both relevant features and models using convolutional neural networks (CNNs). Kang et al. proposed the first such method for NR-IQA, employing a relatively simple network with one 7×7×50 convolutional layer, two pooling layers, and two fully connected layers [10]. This approach matches the performance of those based on explicitly designed perceptual features for NR-IQA.

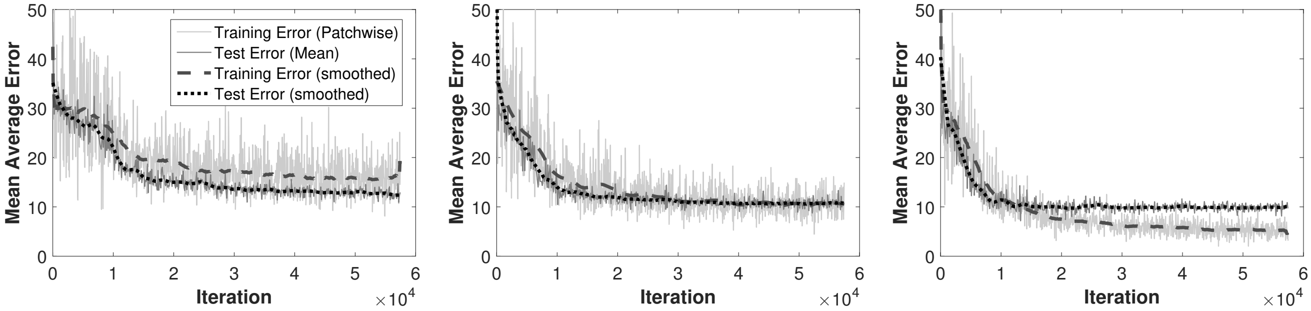
Bosse et al. proposed an adaptation of the VGG deep neural network originally designed for the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [11, 1, 12]. The application of a deeper, more sophisticated network to the problem results in performance comparable to the best full-reference estimators on the LIVE dataset. A modified and simplified extension of this method is proposed below for no-reference utility estimation.

Utility Estimation with a Convolutional Neural Network

Deep convolutional neural networks (CNNs) have proven to be highly discriminative and accurate tools in classification applications such as image and action recognition, with most of the best-known models developed for the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [12]. This section presents a network architecture derived from the VGG model presented by Simonyan et al. for ILSVRC 2014 [1] and an adaptation presented by Bosse et al. for quality estimation [11]. The VGG approach is to build very deep networks using only very small convolution kernels, approximating larger kernel extents by stacking 3×3 convolution layers. This technique increases network depth and nonlinearity and allows similar convolutional coverage to a shallower network with larger kernels while having fewer parameters. The VGG network employs 3×3 convolution kernels and 2×2 pooling operations.

Bosse et al. applied a modified 12-layer VGG network structure to the problem of no-reference quality estimation [11] with excellent results. The method involved extensive data augmentation to overcome limitations of small datasets such as LIVE and CU-Nantes. One of the key differences between the quality or utility estimation task and the object recognition task is that the ILSVRC dataset contains approximately 500,000 images, while the CU-Nantes utility dataset (described in the next section) contains only 235. Due to the small number of images available for training and testing, data augmentation is required. The small extent of the convolution kernels in the VGG network topology makes it straightforward to train the network on small image patches instead of full images. By training the network on many small patches of the original database images and changing those patches for each round of training, the likelihood the network will learn to recognize features specific to the training images is significantly reduced. This method also has the advantage of generating many more training samples per epoch than related techniques such as random shifts, where an epoch is defined as the number of iterations required for each image in the training set to be passed through the network one time.

Each grayscale image is represented as a collection of *N*



(a) Patch Size = 16×16 **(b)** Patch Size = 32×32 **(c)** Patch Size = 64×64

Figure 2: Effect of varying patch size on patchwise training error and test error. A patch size of 32 is found to be optimal, consistent with the observations of [10] and [11].

32×32 pixel randomly sampled patches, where $N = 32$, with new patches sampled each time an image is passed to the network. Each patch is labeled with the parent image's associated subjective utility score l. During training, a patch-wise Mean Average Error (MAE) objective is minimized over the patches taken from each parent image, while during testing the error is calculated for each image by taking the mean predicted patch utility:

$$E_{patchwise} = \frac{1}{N} * \sum_{p=1}^{N} |y_p - l_p| \qquad E_{test} = |(\frac{1}{N} * \sum_{p=1}^{N} y_p) - l| \quad (1)$$

where y_p and l_p represent the predicted and ground truth utility of patch p, respectively, and $l_p = l$. To predict the utility of an unlabeled image, the predicted utility of N patches randomly sampled from the image are averaged. While more complex weighting schemes were pursued in [11], simple patch-wise averaging was found to yield the best performance for globally uniform distortions, and tests on the CU-Nantes database confirmed this observation. The network was trained using the Adam solver, with a base learning rate $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$ [13]. While [11] uses color image patches, the CU-Nantes utility database contains grayscale images only. Tests on the LIVE image quality database reveal that the impact of color channel information is minimal, with the omission of color resulting in performance loss of less than 1%.

The network architecture is shown in Figure 1. This VGG-based network consists of two stacks of three 3×3 convolution layers, with each stack followed by a 2×2 max pooling layer, then one stack of two 3×3 convolutional layers followed by another 2×2 max pooling layer. The network depth was chosen based on the finding that very low frequency image features are not important to the recognition of image utility [2, 6]. The inclusion of two max pooling operations allows for the recognition of image content over a range of spatial frequencies similar to that of MS-DGU, and experimental results confirmed that the addition of a fourth or fifth convolution stack did not significantly improve performance.

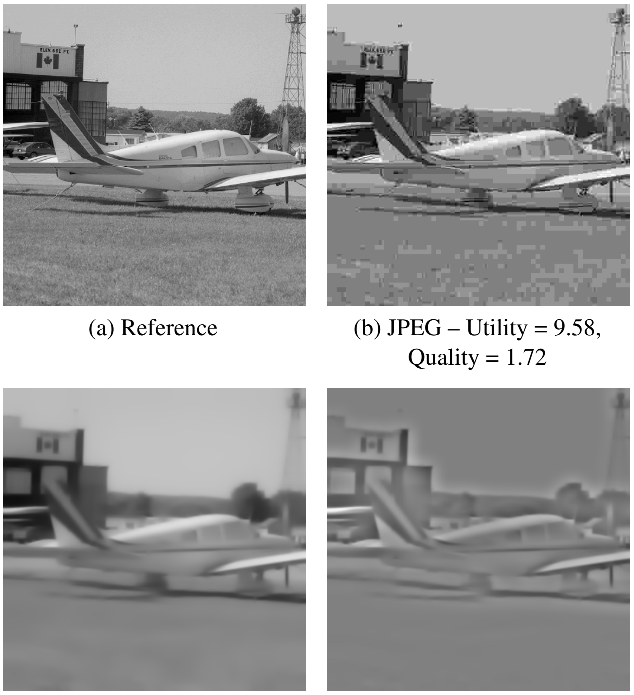
The top of the network consists of one fully connected layer with 128 neurons, sized to match the number of filters of the last convolution layer, followed by a 1-D fully connected layer, with the 1-D output representing the predicted utility of an input image patch. All convolutional and fully connected layers are activated through rectified linear unit (ReLu) activation functions, and dropout is applied to the first fully connected layer with a ratio of 0.5 to help prevent overfitting. Compared to the architecture of Bosse et al., this network omits two convolution stacks consisting of two layers each with 256 and 512 filters, respectively. As a result, the network has approximately one tenth the number of parameters. At the same time, this network has a larger filter extent in the first two convolution stacks with the inclusion of an extra layer in each stack, allowing for greater sensitivity to lower frequency content in shallow network layers.

Figure 2 shows the result of varying patch size. Test error decreases when increasing the patch size from 16×16 to 32×32. With a patch size of 32×32, training and test error converge to a similar point. With a patch size of 64×64, test error converges to approximately the same level as with 32×32 patches, while training error is lower, indicating overfitting. Additionally, reducing N to 16 results in significantly higher error (not shown). Experimental results suggest that a patch size of 32×32 is optimal, with at least $N = 32$.

Performance Evaluation

The CNN described above was implemented using the Caffe framework [14] on a machine with an Intel Core i7 processor and NVIDIA GTX 970 graphics card. Training was conducted for 1000 epochs. A batch size of four images was used, corresponding to 128 patches when $N = 32$. Batch sizes of anywhere from two to eight work equally well. At the completion of each epoch, a new set of N randomly sampled patches was generated for each image. On this system, each epoch takes approximately 1 second.

Estimator performance is evaluated using the CU-Nantes utility database, created by Rouse et al. [2]. Paired comparison experiments were conducted to collect subjective utility and scores for a variety of images. The database consists of 9 grayscale reference images (scenes) and 235 distorted versions of those references. All images are 512×512 pixels. Five types of distortion are represented: JPEG Compression, Blocking (DCT DC coefficient quantization), JPEG2000 compression using Dynamic Contrast Quantization [15], texture smoothing (TS, soft thresholding of Haar wavelet coefficients), and texture smoothing with high-pass filtering (TS+HPF). Some of these images are so distorted that they are below the human recognition threshold; in other words, their content is unrecognizable. A utility score of zero corresponds to the *recognition threshold*, below which an image is not useful as a substitute for the reference scene. An



(a) Reference

(b) JPEG – Utility = 9.58, Quality = 1.72

(c) TS – Utility = 12.3, Quality = 1.96

(d) TS+HPF – Utility = 8.35, Quality = 1.56

Figure 3: [6] Reference image *airplane* and low-utility representations. Subjective utility on interval [0,100] (from not useful to a perfect substitute). Subjective quality scores on interval [1,5] (higher is better).

image with a utility score of 100 is visually identical to the reference, and scores above 100 represent an image which is more useful than the reference.

Fig. 3 depicts a reference image distorted by several processes, such that the resulting images are of low utility but above the recognition threshold.

Following the convention of [2, 6], unrecognizable images, with utility less than -15, are not included when testing the performance of utility estimation algorithms. In this case, they also are not included in the training set, reducing the number of images to 163. The database is split into training and test sets using a leave one out methodology. The nine scenes are split nine times. In each split, one scene is left out for testing, and a network is trained on the other eight. A different scene is left out in each split, so each image in the dataset is tested once. The predicted utility scores and ground truth labels from each split are then concatenated to calculate overall network performance.

Table 1: Performance of various estimators as utility estimators. Full-reference estimators above horizontal divider, no-reference estimators below. VGG-CNN refers to the VGG-based network presented in this paper. Shown: Spearman's ρ, Kendall's τ, Pearson's r, root mean square error (RMSE), outlier ratio (OR). † Quality estimators, but used to estimate utility. * Full-reference estimators.

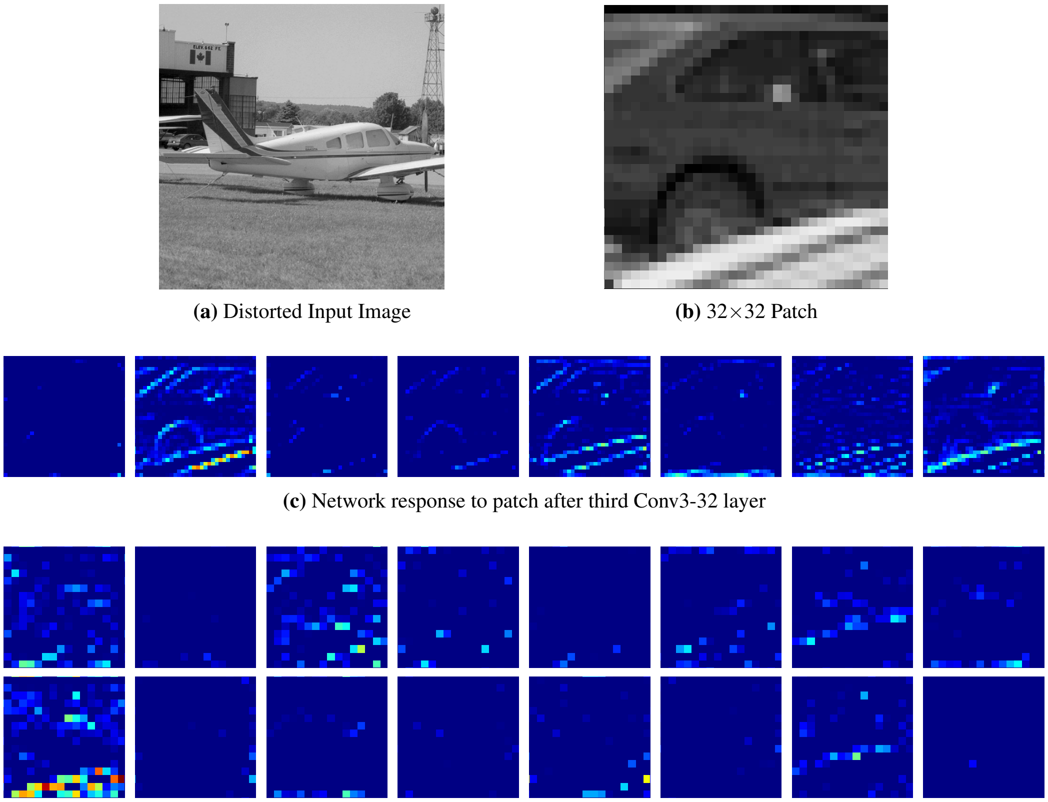
Utility Estimator	ρ	τ	r	RMSE	OR
PSNR †*	.520	.422	.414	34.1	.859
VIF †* [16]	.959	.821	.943	12.4	.583
NICE* [2]	.937	.785	.935	13.3	.460
MS-DGU* [6]	.966	.838	.967	9.5	.436
NIQE [8]	.928	.752	.892	16.95	.638
OG-IQA [9]	.901	.712	.905	16.0	.650
Kang-CNN [10]	.934	.766	.928	15.46	.632
BRISQUE [7]	.931	.766	.934	13.5	.558
VGG-CNN	.942	.779	.946	12.3	.549

The results of tests on the CU-Nantes database are shown in Table 1 for several full-reference and no-reference estimators, where algorithms above the horizontal line are full-reference and those below are no-reference. Three correlation metrics are reported: Pearson linear correlation and Kendall and Spearman rank correlation. It has been shown that an affine transformation is sufficient to map objective utility estimates to the range of subjective values contained in the CU-Nantes database, and all three correlation measures are applicable [2, 6]. Also reported are two accuracy statistics: RMSE and outlier ratio (OR). The outlier ratio represents the proportion of estimates which are outside two standard deviations of the mean opinion score. A higher OR indicates lower reliability of an estimator as compared to human observers. Estimators not already described include PSNR, presented purely due to its ubiquity, and Visual Information Fidelity, which was designed as a quality estimator, but was found to outperform other quality estimators when applied to the task of utility estimation [16, 2]. The network described in this paper is shown in bold as VGG-CNN, and matches or outperforms VIF and NICE, both well-performing full-reference estimators. Though it cannot match the accuracy of MS-DGU, it predicts utility better than all other no-reference techniques tested, and is potentially useful for many more applications than a full-reference algorithm. While it takes time to train a neural network, once the network is trained the speed of utility prediction for a given image is comparable to other methods.

Additional testing was done to measure the performance degradation when each type of distortion from CU-Nantes was excluded from the training set. These results are shown for BRISQUE and the VGG-CNN in Table 2, with MS-DGU broken down by distortion type for comparison (note that MS-DGU is not a trained estimator). The nine train and test splits were split a second time by distortion type. The score for each distortion represents the concatenated predictions of networks which were trained on data not including that distortion type. The overall results (in the rightmost column) are generated by concatenating the predictions for all splits. While overall performance is certainly worse when the VGG-CNN is tested with distortions not present

Table 2: Pearson correlation of BRISQUE and VGG-CNN predictions with CU-Nantes ground truth scores when tested on distortion types not included during training. Also shown are results for the same VGG-CNN model after additional fine-tuning (FT) including each distortion type, and MS-DGU's performance by distortion type. J2K = JPEG 2000, TS = Texture Smoothing, and TS+HPF = Texture Smooothing + High Pass Filter.

Ut. Estimator	JPEG	J2K	TS	TS+HPF	All
MS-DGU [6]	.975	.970	.973	.962	.967
BRISQUE [7]	.820	.902	.941	.927	.821
VGG-CNN	.903	.942	.930	.895	.872
VGG-CNN (FT)	.953	.962	.943	.908	.942



(a) Distorted Input Image

(b) 32×32 Patch

(c) Network response to patch after third Conv3-32 layer

(d) Network response to patch after third Conv3-64 layer

Figure 4: Network response to one randomly sampled 32×32 patch from a JPEG distorted image. (c) and (d) show a sampling of filter responses to the patch after the third Conv3-32 and third Conv3-64 layers, respectively. The first convolution stack clearly emphasizes image contours, while the second stack appears to capture more complex relationships relating to image content around those contours.

in the training set, its performance degrades significantly less than that of BRISQUE when tested on JPEG and JPEG 2000 distortions. BRISQUE does a better job evaluating texture smoothed images, but overall the VGG-CNN experiences less performance degradation than BRISQUE. Both BRISQUE and the VGG-CNN suffer when tested on high-pass-filtered images in comparison to MS-DGU, indicating a sensitivity to unimportant low frequencies.

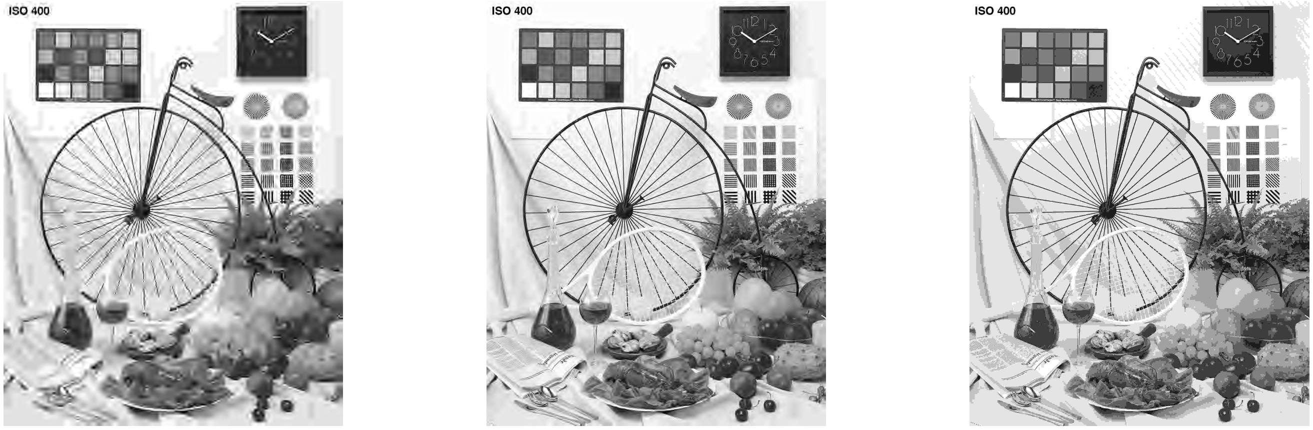
Furthermore, if a VGG-CNN model encounters previously unseen types of distortion, those distortion types can then be incorporated into the training data and the model fine-tuned based on new information, without having to retrain from scratch. Line 4 of Table 2 shows the result of retesting the models of Line 3 after 150 epochs of fine-tuning, where the network is initialized with previously learned parameters at the start of training. Performance returns nearly to the level of the network trained from scratch on all distortion types (shown in Table 1). The VGG-CNN approach is adaptable in the event system parameters change or unexpected conditions are encountered.

Examining the first two convolution stacks of the network provides some clues as to the type of features being learned. Figure 4 shows an undistorted input image, a randomly sampled patch, and a sampling of filter responses from the first and second convolution stacks. The network appears to emphasize image contours and features of the surrounding image content, consistent with early phases of previously developed full-reference utility estimators.

Finally, Figure 5 shows the result of applying the network to an image not included in the CU-Nantes database, using the same distortions as those on which the network was trained. The results shown indicate that the presented neural network approach is not scale invariant. The network's lack of scale-invariance is likely related to the contents of the CU-Nantes database; images in the database are the same size, and were viewed at a fixed distance. Possible approaches to address this issue include training the network with a more comprehensive database, which does not currently exist, or pursuing additional pre-processing steps.

Conclusions

Utility can be predicted reliably without a reference image by employing deep convolutional neural networks. The dCNN model proposed generalizes to distortion types outside the training set with less performance loss than other no-reference approaches, and the models are easily adaptable to new types of distortion with relatively little additional training. Additionally, activations of the first few network layers are consistent with early stages of specifically designed full-reference quality estimation algorithms. These attributes, combined with the lack of a requirement of a reference image, render this approach more adaptable to varying applications than other techniques.



(a) Predicted Utility = 5.3 **(b)** Predicted Utility = 19.1 **(c)** Predicted Utility = 8.9

Figure 5: Size 2560×2048 image. (a) and (b) compressed with Jpeg2000 with a target bitrate of 0.01 and 0.05, respectively. (c) compressed with JPEG with a quality factor of 3. If the images are resized to 640×512, matching the size of CU-Nantes database images, predicted utilities are 32.9, 74.8, and 85.9, respectively.

References

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

[2] David M. Rouse, Sheila S. Hemami, Romuald Pépion, and Patrick Le Callet. Estimating the usefulness of distorted natural images using an image contour degradation measure. *JOSA A*, 28(2):157–188, 2011.

[3] Terry L. Bisbee. Today's thermal imaging systems: background and applications for civilian law enforcement and military force protection. In *International Carnahan Conference on Security Technology*, pages 202–208. IEEE, 1997.

[4] Mikolaj I. Leszczuk. Determining image quality requirements for recognition tasks in generalized public safety video applications: Definitions, testing, standardization, and current trends. In *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5. IEEE, 2011.

[5] David M. Rouse, Romuald Pépion, Sheila S. Hemami, and Patrick Le Callet. Image utility assessment and a relationship with image quality assessment. In *Proceedings of the SPIE*, volume 7240, pages pp–724010, 2009.

[6] Edward T. Scott and Sheila S. Hemami. Image utility estimation using difference-of-gaussian scale space. In *2016 23rd IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016 (To be presented).

[7] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[8] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[9] Lixiong Liu, Yi Hua, Qingjie Zhao, Hua Huang, and Alan C. Bovik. Blind image quality assessment by relative gradient statistics and adaboosting neural network. *Signal Processing: Image Communication*, 40:1–15, 2016.

[10] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, 2014.

[11] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. A deep neural network for image quality assessment. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3773–3777. IEEE, 2016.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia. Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[15] Damon M. Chandler and Sheila S. Hemami. Dynamic contrast-based quantization for lossy wavelet image compression. *IEEE Transactions on Image Processing*, 14:397–410, April 2005.

[16] Hamid R. Sheikh and Alan C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15:430–444, February 2006.

Author Biography

Edward T. Scott received the B.S. and M.S. degrees from Northwestern University in 2010 and worked as an associate technical staff member at MIT Lincoln Laboratory from 2010-2014. He is currently a PhD candidate at Northeastern University in the department of Electrical and Computer Engineering. His research interests include human visual perception, image analysis, and image quality assessment.

Sheila S. Hemami received the Ph.D. degree from Stanford University (1994) and is currently with Draper in Cambridge, MA. She has held positions with Hewlett-Packard Laboratories, Cornell University, and Northeastern University. Dr. Hemami is a Fellow of the IEEE and has held various leadership positions in the IEEE. She has received numerous college and national teaching awards. Her research interests broadly concern communication of visual information, both from a signal processing perspective and from a psychophysical perspective.