Learning to Distort Images Using Generative Adversarial Networks

Li-Heng Chen, Christos G. Bampis, Zhi Li, and Alan C. Bovik, *Fellow, IEEE*

***Abstract*—Modeling image and video distortions is an important, but difficult problem of great consequence to numerous and diverse image processing and computer vision applications. While many statistical models have been proposed to synthesize different types of image noise, real-world distortions are far more difficult to emulate. Toward advancing progress on this interesting problem, we consider distortion generation as an image-to-image transformation problem, and solve it via a data-driven approach. Specifically, we use a conditional generative adversarial network (cGAN) which we train to learn four kinds of realistic distortions. We experimentally demonstrate that the learned model can produce the perceptual characteristics of several types of distortion.**

***Index Terms*—Convolutional neural networks (CNNs), distortion model, generative adversarial networks (GANs), perceptual image quality.**

I. Introduction

THERE has long been a great interest in removing unpleasant artifacts from image or video signals. However, the "reverse" problem of generating artifacts on pristine source pictures has been studied hardly at all. Being able to model distortions of image/video is a crucial research problem, owing to the boom of social media and streaming video content and the great diversity of complex and commingled distortions that afflict them. While some distortions may be easily synthesized, such as compression artifacts (blocking, ringing, etc.), the broad spectrum of real-world distortions are complex, hard to model, and generally hard to synthesize. Moreover, multiple distortions may coexist and combine [1], creating more difficult synthesis scenarios.

Blur arising from camera or object motion, defocus or low light commonly arises in modern digital cameras is notoriously hard to model and synthesize. Being able to synthesize realistic examplars of complex picture distortions would be a boon to algorithms developers seeking to recognize, evaluate, and remediate distortions. There are also a wide variety of noise processes that arise in imaging which interact with other distortions, such as compression, creating composite, hard-to-describe distortions. Banding is a severe compression distortion of great concern to the streaming video industry [2]. Although banding is produced by compression algorithms, it is difficult to predict where it will occur, and in what configuration, making it hard to synthesize.

The question thus arises whether data-driven methods of distorted picture synthesis might be feasible, using deep generative neural networks. For example, generative networks have been trained to reconstruct high-quality pictures from degraded pictures [3]–[6]. An important barrier, however, is obtaining enough before-and-after distortion picture samples, especially of degradations not produced by an algorithm (like compression).

Motivated by the significant potential of generative models to create pictures that are realistic but not genuine, we investigated the potential capacity of GANs to learn a pristine-to-distorted mapping function, given adequate, representative training data. Hence, we built four dedicated training datasets containing images impaired by single distortions, on which we trained GAN models to emulate distortions of real pictures.

Related Work: Little attention has been paid to studying the generation of picture distortions using deep networks. Of course, there has long been an interest in modeling and synthesizing noise and blur pictures as part of testing de-noising and de-blurring algorithms [8]–[13]. Modeling picture blur has long been treated as a linear kernel estimation problem [14], [15]. However, blur is often nonlinear, space-variant, and can arise in many ways, making it hard to model.

Several learning-based noise modeling approaches have been proposed, such as Noise Flow [16], which applies a flow-based generative model to maximize the likelihood of image noise. An early GAN model [17] generated noise to train a denoiser, but did not utilize any information from pristine images. The Grouped Residual Dense Network (GRDN) denoiser [18] is trained on images generated by a GAN generator with conditional side information. All these deep noise models are designed to model specific camera noises. Deep neural networks (DNNs) have also been used to estimate blur convolutional kernels [19], [20].

Our goal is to better understand the capabilities of GANs to generate broader classes of distortions. Learning accurate models that generate the gamut of possible distortions is an ambitious goal, given the diversity of impairments and the complex ways they depend on and interact with picture content (and other distortions), and how they affect appearance. We make no claims to have solved this large problem, which will require large datasets of all kinds of real distortions. In fact, we only attempt to generate distortions easily synthesized by other means, to probe the capabilities of GAN models to generate complex distortions.

Manuscript received September 22, 2020; revised November 20, 2020; accepted November 22, 2020. Date of publication November 26, 2020; date of current version December 21, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mylene Q. Farias. *(Corresponding author: Li-Heng Chen.)*

Li-Heng Chen and Alan C. Bovik are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: lhchen@utexas.edu; bovik@ece.utexas.edu).

Christos G. Bampis and Zhi Li are with Netflix Inc., Los Gatos, CA 95032 USA (e-mail: christosb@netflix.com; zli@netflix.com).

Digital Object Identifier 10.1109/LSP.2020.3040656

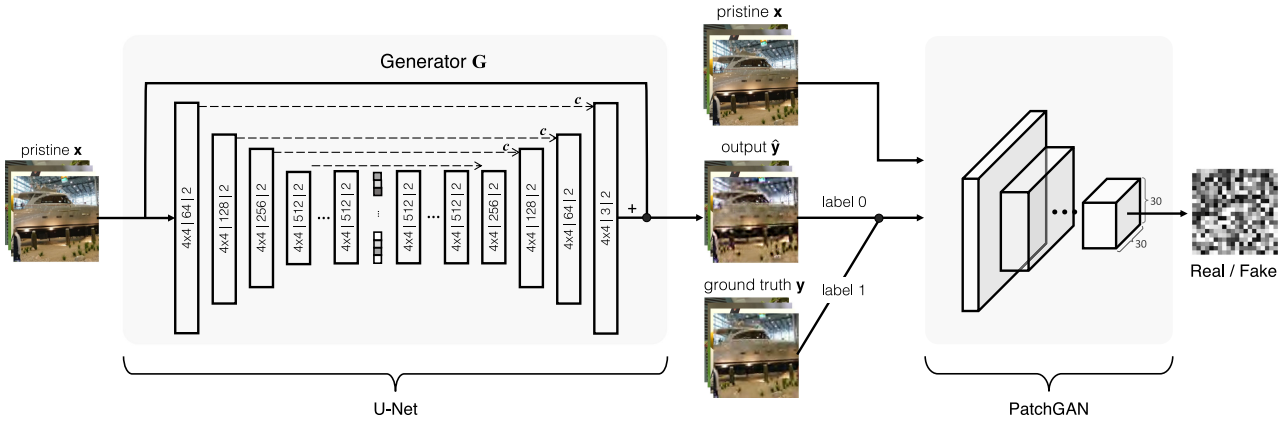

Fig. 1. Schematic diagram of how the proposed GAN-based distortion generator is trained. For each block of the generator, the convolutional parameters are denoted as: kernel size (height × width) | number of filters | stride. The operation **c** denotes concatenation. The discriminator is identical to PatchGAN [7].

Specifically, we train the GAN on large datasets of synthetically created JPEG, JPEG2000, Gaussian white noise, and pink noise. The successes of our attempts are interesting enough to motivate future attempts to learn distortions not easily simulated by other means. Successes on this spectrum of problems could impact future picture correction, quality assessment, and compression algorithms.

The letter is arranged as follows. We present the details of our GAN-based distortion synthesizer in Section II. Section III provides experimental results and analysis. Finally, Section IV concludes the letter.

II. Learning a Pristine-to-Distorted Picture Function

As illustrated in Fig. 1, the idea is to use a GAN [21] to corrupt a pristine picture. This involves optimizing both a generative network G, and a discriminative network D. The generator G produces distorted pictures, while the discriminator D decides whether the input was produced by G, or is a picture afflicted by true distortion. The pristine "conditioning" picture is input to both the generator and the discriminator.

A. Network Architecture

We used a 14-layer U-Net similar to [22] as the generator. The output of each block in the encoder is concatenated with the input of a corresponding block in the decoder. On the encoder side, each convolutional layer (stride $= 2$) has 4×4 filter kernels followed by Batch Normalization (BN) and Leaky ReLU (LR) activation. Each block on the decoder side is composed of upsampling, with symmetric parameters, followed by BN and LR. Different from the original structure in [22], an additional skip connection is used at the input and output of the generator, allowing the network to learn residuals (the distortions) rather than the distorted pictures.

Since the perception of picture distortions is content-dependent, because of, e.g., masking effects, its perceived visibility and severity varies with a picture [23]. Thus, aiming to learn local distortion structures, we reward or penalize generated distortions on local patches, instead of on entire pictures. We deployed the PatchGAN [7] architecture as the discriminator. PatchGAN classifies each patch as real or fake. The contractive discriminator yields a 30×30 output receptive field of the patch classes.

B. Loss Functions

Denote a distorted batch of corresponding pictures from ground truth and the generator by $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively. Let $\mathbf{x}$ be the batch of corresponding pristine pictures. When training the GAN, we used the hinge version of the adversarial loss [24], [25], which empirically performed better than the standard GAN losses in our tasks. The generator and discriminator networks are trained alternately as follows.

Updating the generator: The goal of the generator is to "fool" the discriminator by producing realistic distortions. To minimize the classification error, the discriminator output D drives the optimization of G:

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\mathbf{x},\hat{\mathbf{y}}}\left[D(\mathbf{x},\hat{\mathbf{y}})\right] = -\mathbb{E}_{\mathbf{x},\hat{\mathbf{y}}}\left[D(\mathbf{x},G(\mathbf{x}))\right]. \quad (1)$$

We also introduce a regularization term on the learned residuals. The underlying assumption is that common distortions are generally sparse in some space. Hence, we use the ℓ_l norm to preserve sparsity and also improve training stability

$$\mathcal{L}_{\text{sparse}} = \mathbb{E}_{\mathbf{x},\hat{\mathbf{y}}}\left[\|G(\mathbf{x}) - \mathbf{x}\|_1\right]. \quad (2)$$

Finally, the model parameters of D are fixed, while training G using the weighted loss function

$$\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \lambda \mathcal{L}_{\text{sparse}}. \quad (3)$$

By back-propagating through the forward model, the loss derivative is used to drive G. We used $\lambda = 0.1$ as the weighting factor in all of the experiments.

Updating the discriminator: To learn to distinguish between real distorted pictures and generated data, D is updated by minimizing the hinge discriminator loss function:

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\min(0, 1 + D(\mathbf{x},\mathbf{y}))\right] + \mathbb{E}_{\mathbf{x},\hat{\mathbf{y}}}\left[\min(0, 1 - D(\mathbf{x},\hat{\mathbf{y}}))\right]. \quad (4)$$

Similarly, the model parameters of G are fixed while training the discriminator.

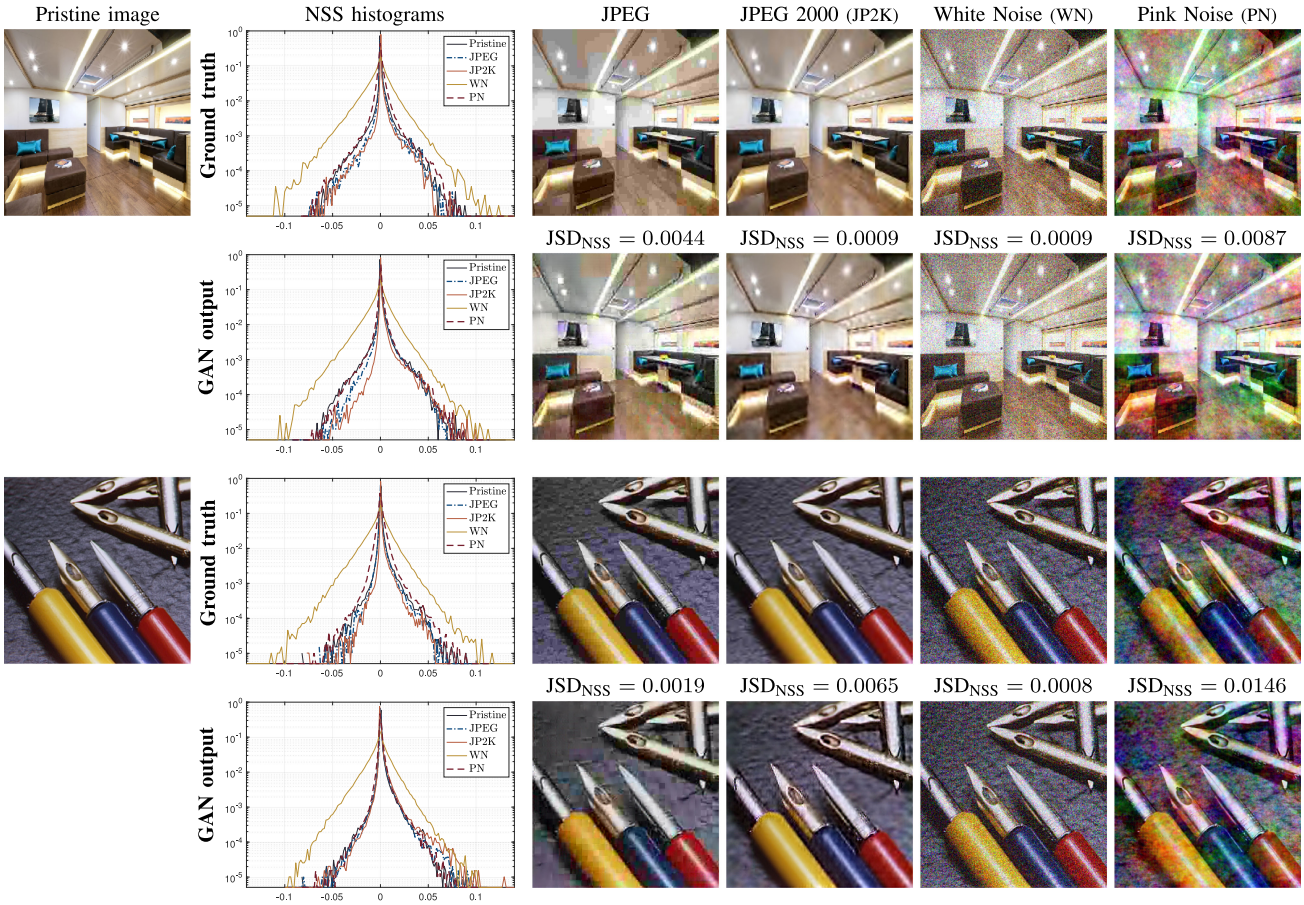

Fig. 2. Example qualitative and quantitative result comparisons on four distortions. The NSS histograms show the empirical probability distributions of products of adjacent MSCN coefficients. The Jensen Shannon divergence (JSD$_{NSS}$) is used to measure the similarity between two NSS distributions of each distortion.

III. Experiments

A. Implementation and Training Details

We used the TensorFlow framework (version 2.3.0) to implement the distortion generation system. The Adam solver [26] was used to optimize both networks with a batch size of 16. We set the learning rates at fixed values of $3e-4$ for the generator, and $1e-4$ for the discriminator. The pictures were randomly cropped to size 256×256.

We used a subset of 4700 pristine pictures from the Waterloo Exploration Database [27] as training data, keeping out 44 pristine pictures for validation. Each picture was distorted by four distortions: JPEG compression, JPEG2000 compression, additive white Gaussian noise, and pink noise to serve as ground truth data. While generative models have been used for noise generation, modeling compression is a challenging problem. Compression engines like JPEG are nonlinear, lossy, multi-stage processes for which no tractible models exist, but which represent complex pixel distribution mappings.

B. Quantitative and Subjective Evaluation

While several GAN image quality evaluation measures have been recently introduced, there is no consensus on their relevant efficacies [28]. These were designed to assess "good quality," whereas our goal is to assess "good distortion". The inception score [29] and the FID score [30] capture both result diversity and the distance between distributions, which are not related to our scenario. Existing full-reference (FR) picture quality models [31]–[34] measure perceptual fidelity between images, hence are not well suited for this task. Natural scene statistics (NSS) models are a more direct way to assess performance on this problem [35]. For example, bandpass, normalized picture coefficients [36], [37] analyzed under a Gaussian scale mixture model can accurately distinguish distortions [38], [39]. Given a picture I at pixel location (i, j), we use the mean subtracted contrast normalized (MSCN) coefficients

$$I_{\text{NSS}}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + 1}, \tag{5}$$

where μ and σ are Gaussian-weighted sample means and standard deviations [36], [37]. The product of diagonally adjacent coefficients captures local spatial correlations:

$$I_{\text{PMSCN}}(i,j) = I_{\text{NSS}}(i,j) I_{\text{NSS}}(i+1, j+1). \tag{6}$$

We refer the reader to [36], [37] for more details. The goal is to compare the statistics of true distortions with those generated by the GAN. To do this, we used the Jensen Shannon divergence [40] computed on the empirical distributions of (6), which we denote by JSD$_{NSS}$. A smaller JSD$_{NSS}$ value indicates that two distributions are more similar.

TABLE I
OVERALL COMPARISON OF GENERATED VS. GROUND TRUTH DISTORTIONS. THE SMALLEST MEAN JSD_{NSS} IN EACH ROW AND COLUMN ARE HIGHLIGHTED BY **BOLDFACE** AND UNDERLINE, RESPECTIVELY

GAN output \ Ground truth	JPEG	JP2K	WN	PN
JPEG	0.0107	**0.0043**	0.2191	0.0470
JP2K	0.0116	**0.0031**	0.2321	0.0541
WN	0.1847	0.2149	**0.0008**	0.0959
PN	0.0195	0.0317	0.1377	**0.0062**

Fig. 2 visually compares exemplar ground truth distorted pictures and corresponding GAN-synthesized pictures, for each distortion type. The characteristics of each distortion are effectively captured by the GAN models, including blocking and detail-crushing JPEG artifacts, and JPEG2000 edge ringing effects. Of course, the synthesized pictures are not replicas of the compressed ones. Instead, they are statistical models that perceptually resemble the ground truth distortions. The synthesized white noise picture is perceptually nearly identical to ground truth, while the synthesized pink noise result is also highly similar to ground truth, albeit with shifting color effects.

We summarize quantitative similarities computed on the validation set in Table I. Each cell presents averaged JSD_{NSS} values between the generated row distortions and the ground truth column distortions. The results the four GAN distortion models yielded MSCN distributions similar to the ground truth distortions (Table diagonal). The larger off-diagonal values were calculated from different distortion types having distinct distributions. Ideally, the ith diagonal value should be the smallest value of both the ith column and the ith row. The JSD_{NSS} values between the simulated and real noise distortions are indeed small, and somewhat larger for the compression distortions. There was one interesting anomaly, as the GAN-synthesized JPEG pictures were found to be statistically more similar to the JPEG2000 ground truth, despite their accurate appearances. This is likely a failing of the rather simplistic statistical similarity measure (6), given the nearly copacetic appearance of the synthesized pictures. We are making available many other generated examples at: https://live.ece.utexas.edu/research/liheng/distortiongan/.

C. Study of the Sparsity Constraint

Figure 3 shows the effects of varying the sparsity weight λ. Severe, unrealistic artifacts are observed without the sparsity loss ($\lambda = 0$). We found choosing $\lambda = 0.1$ reduces excessive artifacts, while yielding perceptually similar results to ground truth JPEG distortions. Larger weights caused the artifacts to more subtle, since the learned residuals are regularized too heavily. Selecting $\lambda = 0.1$ also yielded low JSD_{NSS} values, indicating MSCN distributions closer to the ground truth.

D. Why Not Train With Pixel-Wise Losses?

Of course, we also explored the simplest training method: minimizing pixel-wise errors between the ground truth distorted patches and the synthesized distortions to obtain the generative network G. Unfortunately, severe complications arise when applying these methods.

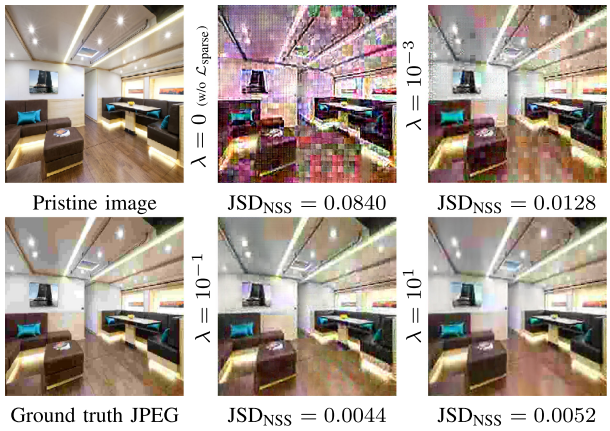

Fig. 3. Effect of the sparsity loss $\mathcal{L}_{\text{sparse}}$ in (2) using different values of λ.

Fig. 4. Examples of learned JPEG distortions using pixel-wise loss functions.

We observed that using simple ℓ_p norms ($p = 1, 2$) as the loss function always yields blurry results. Figure 4 shows an example generated using two common pixel losses. Prior studies have shown that ℓ_p losses lead to blurry images when the data are drawn from multi-modal distributions [41]. This is understandable: the optimal solution to the ℓ_p reconstruction loss of the decoder is the expectation of a set of images [42], hence details may be expected to be "averaged out."

E. Limitations

Despite the successes we have obtained, we still observe several limitations of our model. Some distortion characteristics are hard to learn, such as large, continuous false contours ("banding") [43] on JPEG-compressed smooth regions. Since banding is a relatively global effect, banding may be better learned by deepening the network to produce broader-scale feature maps.

IV. CONCLUSION AND FUTURE WORK

We have shown that GANs can be used to produce different types of realistic distortions. In the future, this idea may be extended to generate distortions that are hard to collect, or cannot be synthesized to build ground truth datasets. The proposed JSD_{NSS} metric could be further justified by conducting a series of subjective tests, or improved by aggregating more NSS features into (6). We are also seeking ways to use this kind of model for picture quality assessment research. Large numbers of generated distorted pictures could be used in perceptual studies [44]. Moreover, learned generative models could be "unrolled" [45] to study distortions, or plugged into end-to-end training protocols to act as "surrogates" [46] for non-differentiable degradation modules.

REFERENCES

[1] Z. Tu, Y. Wang, N. Birkbeck, B. Adsumilli, and A. C. Bovik, "UGC-VQA: Benchmarking blind video quality assessment for user generated content," 2020, *arXiv:2005.14354*.

[2] Z. Tu, J. Lin, Y. Wang, B. Adsumilli, and A. C. Bovik, "BBAND index: A no-reference banding artifact predictor," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 2712–2716.

[3] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 769–776.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[6] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI*, 2019, pp. 8794–8802.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[8] P. Chatterjee and P. Milanfar, "Is denoising dead?," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, Apr. 2010.

[9] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, Jun. 2009, pp. 1964–1971.

[10] H. Al-Ghaib and R. Adhami, "On the digital image additive white Gaussian noise estimation," in *Proc. Int. Conf. Ind. Autom. Inform. Commun. Technol.*, Aug. 2014, pp. 90–96.

[11] P. Gupta, C. G. Bampis, Y. Jin, and A. C. Bovik, "Natural scene statistics for noise estimation," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, Apr. 2018, pp. 85–88.

[12] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian–Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[13] A. Norkin and N. Birkbeck, "Film grain synthesis for AV1 video codec," in *Proc. Data Compression Conf.*, Mar. 2018, pp. 3–12.

[14] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006.

[15] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman, "Blur kernel estimation using the radon transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 241–248.

[16] A. Abdelhamed, M. Brubaker, and M. Brown, "Noise Flow: Noise modeling with conditional normalizing flows," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 3165–3173.

[17] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018, pp. 3155–3164.

[18] D.-W. Kim, J. R. Chung, and S.-W. Jung, "GRDN: Grouped residual dense network for real image denoising and GAN-based real-world noise modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2019, pp. 2086–2094.

[19] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, Jul. 2016.

[20] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang, "Motion blur kernel estimation via deep learning," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 194–205, Jan. 2018.

[21] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[23] A. C. Bovik, "Automatic prediction of perceptual image and video quality," *Proc. IEEE*, vol. 101, no. 9, pp. 2008–2024, Sep. 2013.

[24] J. H. Lim and J. C. Ye, "Geometric GAN," 2017, *arXiv:1705.02894*.

[25] D. Tran, R. Ranganath, and D. M. Blei, "Hierarchical implicit models and likelihood-free variational inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5523–5533.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representation*, 2015, pp. 1–15.

[27] K. Ma *et al.*, "Waterloo Exploration Database: New challenges for image quality assessment models," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1004–1016, Feb. 2017.

[28] A. Borji, "Pros and cons of GAN evaluation measures," *Comput. Vis. Image Understanding*, vol. 179, pp. 41–65, Feb. 2019.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[30] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

[31] Z. Wang, A. C. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[32] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.*, Nov. 2003, pp. 1398–1402.

[33] D. Chandler and S. Hemami, "VSNR: A wavelet-based visual signal-to-noise ratio for natural images," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2284–2298, Sep. 2007.

[34] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[35] H. Ko, D. Y. Lee, S. Cho, and A. C. Bovik, "Quality prediction on deep generative images," *IEEE Trans. Image Process.*, vol. 29, pp. 5964–5979, 2020.

[36] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[37] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[38] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal Process. Lett.*, vol. 17, no. 5, pp. 513–516, May 2010.

[39] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3350–3364, Dec. 2011.

[40] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[41] L. Cai, H. Gao, and S. Ji, "Multi-stage variational auto-encoders for coarse-to-fine image generation," in *Proc. SIAM Int. Conf. Data Mining*, May 2019, pp. 630–638.

[42] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information maximizing variational autoencoders," in *Proc. AAAI*, 2019, pp. 5885–5892.

[43] Z. Tu, J. Lin, Y. Wang, B. Adsumilli, and A. C. Bovik, "Adaptive debanding filter," *IEEE Signal Process. Lett.*, vol. 27, pp. 1715–1719, 2020.

[44] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 372–387, Jan. 2016.

[45] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," 2019, *arXiv:1912.105578*.

[46] L.-H. Chen, C. G. Bampis, Z. Li, A. Norkin, and A. C. Bovik, "ProxIQA: A proxy approach to perceptual optimization of learned image compression," *IEEE Trans. Image Process.*, vol. 30, pp. 360–373, 2020.