GAN-based Noise Model for Denoising Real Images

Linh Duy Tran[1], Son Minh Nguyen[1], and Masayuki Arai[1]

Graduate School of Science and Engineering
Teikyo University, Utsunomiya, Tochigi 320-8551, Japan
{duylinh161287,nguyenminhson1110}@gmail.com
arai@ics.teikyo-u.ac.jp

Abstract. In the present paper, we propose a new approach for realistic image noise modeling based on a generative adversarial network (GAN). The model aims to boost performance of a deep network denoiser for real-world denoising. Although deep network denoisers, such as a denoising convolutional neural network, can achieve state-of-the-art denoised results on synthetic noise, they perform poorly on real-world noisy images. To address this, we propose a two-step model. First, the images are converted to raw image data before adding noise. We then trained a GAN to estimate the noise distribution over a large collection of images (1 million). The estimated noise was used to train a deep neural network denoiser. Extensive experiments demonstrated that our new noise model achieves state-of-the-art performance on real raw images from the Smartphone Image Denoising Dataset benchmark.

Keywords: deep learning, denoiser, generative network, real-world noisy images

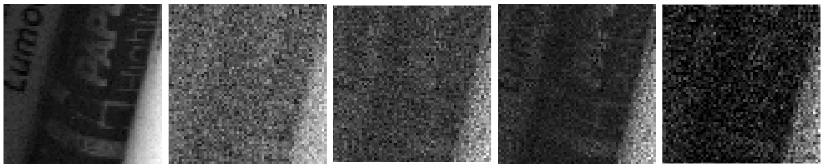
1 Introduction

Noise reduction is a fundamental task in computer vision, and it is used as preprocessing step in many subsequence image processing tasks. Traditional denoising methods include block matching and 3D filtering (BM3D) [1], k-means singular-value decomposition (KSVD) [2], principal component analysis with local pixel grouping (LPGPCA) [3], and weighted nuclear norm minimization (WNNM) [4]; they are designed to remove noise based on the properties of images and noise. In contrast, learning-based methods, such as a denoising convolutional neural network (DnCNN) [5], often use paired-image datasets for mapping from noisy images to clean images. Because the performance of learning-based methods depends on a large training dataset, these methods require a sufficient amount of data. As a result, noisy images are artificially created from clean images with a known type of noise (e.g., additive white Gaussian, salt and pepper, and Poisson). Learning-based methods outperform most of the traditional methods in synthetic denoising.

However, because synthetic noisy images are generally different from real-world noisy images, the learning-based methods work best on the same type of synthetic noise that they were trained on. They often output poor results when denoising real-world noisy images. Recent studies [6, 7] show that the traditional denoising methods outperform learning-based methods when evaluated with real images. In this study, we aimed to improve the performance of learning-based methods for denoising real images.

To resolve this problem, the first approach is collecting a large amount of data for training the models [8, 6, 7]. The nearly noise-free images are estimated by an expensive and time-consuming procedure. Extensive analyses have proved need of a dataset with high-quality image pairs for improving real-world denoising performance. For example, the Smartphone Image Denoising Dataset (SIDD) [6] has collected 30,000 image pairs for training and testing. Although the number of images is relatively large, the number of different scenes (10) is limited. Thus, the dataset may not be sufficient to train a large network.

However, creating synthetic data by adding artificial noise to images has a clear advantage: an unlimited amount of training data can be created. However, the learning-based methods that are trained on synthetic noisy data (e.g., Gaussian noise or Poisson noise) perform poorly on real data because the training noise is unrealistic. Thus, another approach is to focus on building a better noise model. Methods such as convolutional blind denoising network (CBDNet) [9] and unprocessing images (UPI) [10] are intended to build a realistic noise model. In particular, according to the UPI method [10], a combination of Gaussian noise and Poisson noise was added to the raw image by means of an inverting process. For data augmentation, GAN-CNN based blind denoiser (GCBD) [11] used a generative adversarial network (GAN) to learn noise rather than noisy images, and the method was tested with zero mean noise. Despite having some limitations, the GCBD method showed the potential of a learning-based method (GAN) to generate natural noise if the expectation of the unknown noise is available. Moreover, the advantages of applying the denoising algorithm before processing in a non-linear camera processing pipeline have been proven [12, 7].



(a) Noise free (b) Gaussian (c) Shot+Read (d) Our model (e) Real noise

Fig. 1: Different noise model outputs displayed in linear raw space (red channel only).

In light of the above analysis, we adopted a learning-based technique (GAN) to generate noise in raw image data in a large dataset. The proposed model allows us to learn not only synthetic noise but also real noise by using a more realistic dataset for training. Some noisy images generated by noise models are shown in Fig. 1.

The main contributions of the present paper include the following:

- We propose a GAN-based model that can be trained with both synthetic and real noise in raw image data. The generated noise is used to create a paired-image dataset for training a denoising deep neural network (DNN).
- Extensive experiments demonstrated that our method can improve the performance of available denoising DNNs when they denoise real images.

2 Related work

2.1 Deep Learning-based Denoisers

The CNN-based methods dominate image denoising, and they have obtained good performance for removing artificial noise, such as additive white Gaussian noise (AWGN). Among them, DnCNN [5] is the first CNN-based blind denoiser. The DnCNN method shows that the residual learning and batch normalization help boost the denoising performance and training speed as well. Many other denoising methods [9, 13, 14] use the same strategy by using deep neural networks for mapping from noisy images to denoised images. The FFDNet [15] model proposed non-blind denoising by using a noise level map as an additional input. In general, the learning-based methods require abundant paired-image datasets; thus, AWGN is chosen to create the training dataset. These methods can be applied for denoising real-world noisy images; however, due to a lack of real data, their performance on denoising real images is still limited.

Consequently, as another approach to improving real-world image denoising, some recent work has focused on collecting real paired-image datasets. In the studies that created the Darmstadt Noise Dataset (DND) [7] and SIDD [6], the authors proposed extensive procedures to obtain noise-free ground truth images. These procedures require a large number of images to produce a ground truth dataset. Moreover, [6] showed that a deep learning denoiser trained with a high-quality dataset outperforms the classic methods (e.g., BM3D [1]) when denoising real images. Despite this, the number of training images is still relatively small compared with other computer vision tasks (such as image classification). Moreover, it is difficult to apply their process to produce paired images for moving objects. If we can create realistic paired training data, this approach would be promising. We focus on this data augmentation approach because it is generally understood that training data play a critical role in improving the performance of CNN-based methods.

Image denoising can be applied to raw image data as demonstrated in the work of [10]. This method employed a process called “unprocessing” to invert the image processing pipeline. The signal-dependent noise then is added to the clean image to produce a noisy version.

2.2 GAN-based Denoisers

Generative adversarial networks [16] have been actively studied over the past few years. GAN models can minimize a loss function that classifies output images as real or fake. Given a training dataset, the GAN tries to generate new data with the same statistics as the training data. Recent GAN applications can produce impressive results [17, 18], indicating the ability of GANs to learn complex distributions. The idea of applying a GAN to image denoising was first introduced in the work of GCBD [11]. In this work, the generative network was trained to produce noise to create paired-image data. The paired-image data were then used to train a denoising network, such as DnCNN. We adopted this idea for two reasons: first, the GAN can be trained to learn sophisticated real noise. This realistic noise model helps the CNN learn real-world noisy images, thus further boosting the performance of a CNN-based denoiser. Second, the realistic noise model solves the problem of poor denoising performance due to a lack of data.

We improved on prior work [11] not only with some architectural choices but also with the noise formation process. Instead of training the GAN to learn noise in the standard red green blue (sRGB) color space, we trained the GAN model with raw image data obtained by the clean image inverting process described in [10]. We investigated the noise modeling with raw image data and demonstrated the advantages of a GAN for improving the denoising performance for real images.

3 Proposed method

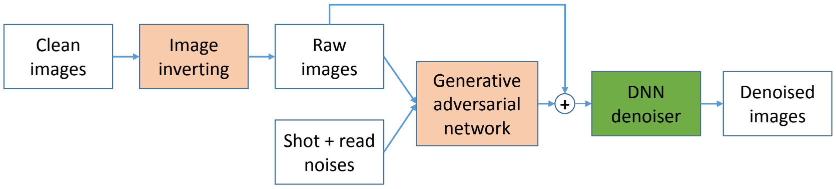


Fig. 2: Overview of proposed model. The "clean" images are converted to raw images through unprocessing. After adding noise, the resulting images are used to train a GAN to produce a noise model. A deep network denoiser uses the GAN output (noise) and clean images to learn how to map (create) clean images from noisy images.

Fig. 2 shows the proposed architecture. The proposed method consists of three steps: First, we follow the process that is shown in [10] to invert clean images from sRGB space to raw image data. The inverted images are assumed to be noise-free images. Second, we obtain a noisy version by adding shot and read noise, and we then train the GAN model to learn the noise distribution from the generated data. During this step, the GAN model is also fine-tuned with a real noise dataset (SIDD) to learn the real noise distribution. Finally, the generative network output (noise) and the clean image are used to produce a paired-image dataset, which is fed to a DNN.

3.1 GAN Noise Generator

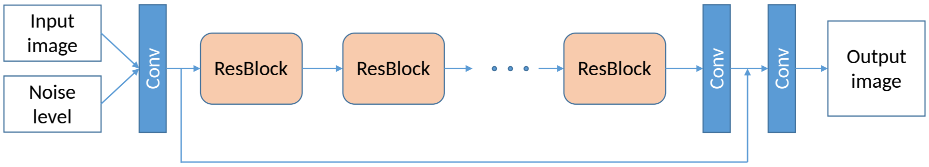


Fig. 3: GAN generator architecture. The model consists of five residual blocks (ResBlocks, see Fig. 4). Input to the network consists of concatenations of clean images and noise levels, and the generator network outputs estimated noise.

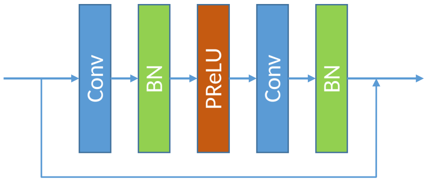


Fig. 4: Residual block (ResBlock)

Architecture: Conventional GAN models [11, 19] often map from a random noise vector z to an output image y. In our proposed GAN model, the noise is added to the input of generative network as described in Fig. 3. We pass the additional input "Noise level" to the generator network, in particular, we estimate the shot and read noise parameters of input image and compute a per-pixel of the standard deviation of that noise. Inspired by the super-resolution GAN (SRGAN) model [18], we adopted that paper's model and removed the up-sampling blocks.

Discriminator: We followed a design similar to that of SRGAN. The discriminator network is depicted in Fig. 5.

Training objective: As described in a previous paper on GANs [16], the aim is to solve a min-max problem between discriminator D and generator G.

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}[\log(D(x))] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log(1 - D(\tilde{x}))], \tag{1}$$

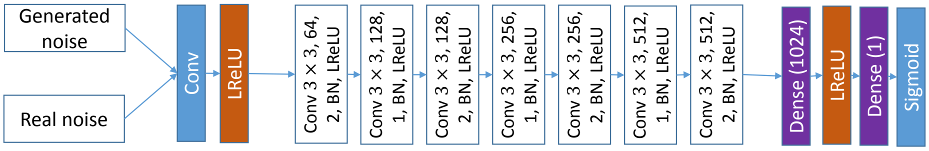

Generated noise
Real noise
Conv
LReLU
Conv 3 × 3, 64, 2, BN, LReLU
Conv 3 × 3, 128, 1, BN, LReLU
Conv 3 × 3, 128, 2, BN, LReLU
Conv 3 × 3, 256, 1, BN, LReLU
Conv 3 × 3, 256, 2, BN, LReLU
Conv 3 × 3, 512, 1, BN, LReLU
Conv 3 × 3, 512, 2, BN, LReLU
Dense (1024)
LReLU
Dense (1)
Sigmoid

Fig. 5: Discriminator architecture. The convolutional units are shown with their corresponding kernel size, number of outputs, and stride. All leaky rectified linear unit (LReLU) layers are set with a negative slope value of 0.2

where $\mathbb{P}_g$ is the synthetic data distribution and $\mathbb{P}_r$ is the real data distribution. Because the output of the generative network is noise, we found that the SRGAN training loss function leads to unstable model training. Arjovsky et al. [20] proposed an alternative objective function, called the Wasserstein distance, which measures the difference between two distributions. Moreover, the Wasserstein GAN with gradient penalty (WGAN-GP) [21] improves the stability of GAN training by introducing a gradient penalty.

$$Loss = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2]. \tag{2}$$

In our experiments, we used the Adam optimizer [22] and kept the hyperparameter values from the original WGAN-GP paper ($\alpha = .0002, \beta_1 = 0.5, \beta_2 = 0.9, and\ \lambda = 10$).

Training steps: For learning real-world noise, we used the following two-step training approach:

- The GAN is pre-trained with a large dataset (MIR dataset) with realistic synthetic noise.
- The pre-trained GAN is then re-trained with another dataset that has real noise (e.g., the SIDD benchmark) but fewer data.

With this training strategy, we can train the GAN to generate more realistic noise while avoiding overfitting of synthetic noise.

3.2 Denoising Neural Network

We produced paired-image data after training the GAN model. The inverted raw image is used as a clean ground truth. In particular, for image denoising networks, we adopt two architectures: DnCNN [5] and UNet-based [23] denoisers.

DnCNN: The DnCNN consists of 17 units with 3 types; more network details can be found in the DnCNN paper [5]. Batch normalization also has been used to speed up the training process and boost the denoising performance. The model predicts the residual image. For the training objective, we used mean squared error (MSE) loss, as suggested in the original paper.

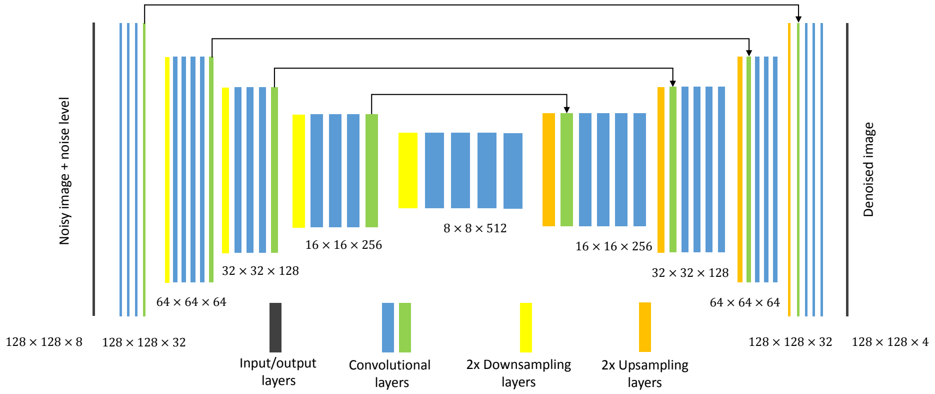


Fig. 6: UNet denoising architecture. Skip connections were added between same-scale encoder and decoder blocks. A 4-channel raw noisy image and noise level are input, and the output is a 4-channel denoised raw image.

UNet: The UNet architecture is depicted in Fig. 6. The model uses skip connections between same-scale encoder and decoder blocks. The input for the UNet denoiser is different from that for the aforementioned DnCNN denoiser. To make the model more robust, we added random noise (shot and read noise) as additional input for the denoising network. The network was trained to predict clean images directly. During the experiment, we found that the MSE loss did not work well as a training objective; thus, we replaced it with the sum of absolute differences loss.

4 Experiment

We performed our experiment on a single machine with an Intel i7-6800 CPU running at 3.40 GHz with 32 GB of RAM and a GeForce GTX 1080 Ti double GPU. The proposed model was implemented in the PyTorch framework [24].

4.1 GAN Training

To train the GAN in a manner similar to that in [10], we used the MIR Flickr extended dataset [25], which contains 1 million high-quality photographic images. The dataset is considered to contain “clean” images. We then randomly cropped each image to the size of 256×256. We randomly flipped the images horizontally and vertically (with probability of 0.5) for data augmentation. Because we did not resize the original image, some small images were not included in the training and validation datasets. We reserved 5% of the data for evaluation. The model learning rate was 10^{-3}, the batch size was 32, and we trained the generative and the discriminative networks for 250,000 iterations.

To train the GAN on a real-world noisy distribution, we then fine-tuned it with a real image dataset (SIDD). The batch size was unchanged, a smaller learning rate (10^{-4}) was used, and the number of training iterations was about 100,000.

4.2 Denoiser Training

The noise produced from the generator network is used to prepare a paired-image dataset for training DNN denoisers. We applied the synthetic data to two denoiser architectures, as described in subsection 3.2:

- DnCNN: We kept the batch size of 32, learning rate of 10^{-3}, and training epochs of 10.
- UNet: The model used a batch size of 64, learning rate of 10^{-4}, and 6 training epochs.

4.3 Test Datasets

SIDD: The SIDD benchmark consists of 30,000 image pairs comprising both raw images and images in the sRGB color space. The evaluation set contained 256×256 size image patches at 32 random non-overlapping regions for each image. A total of 40 images were used for evaluation (total of 1,280 image patches). The dataset was captured under different settings (cameras, camera settings, and light conditions), resulting in 200 scene instances, of which 160 were for training and 40 were for evaluation purposes. The dataset also provided the estimated noise level for each image, which is used as input in many denoising algorithms. To evaluate denoising method, the SIDD dataset provides an online submission system[1].

DND: The DND consists of 50 pairs of images with real noise and corresponding clean images. The (nearly) noise-free image is obtained by averaging a number of noisy images of the same scene. The dataset contains the images taken from four consumer cameras with wide range of different film speeds. An online submission system determined the denoising performance in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)[2].

4.4 Results

Real-world denoising performance: Table 1 shows the denoising results for the SIDD evaluation dataset. We applied our model to denoise raw image, the metrics are evaluated in both raw image data and after converting to s-RGB space. The proposed model outperforms previous denoising models in both metrics (PSNR and SSIM). In particular, the DnCNN with GAN-based modeling surpassed the original DnCNN by 2.25 dB. In the second test, we see that the proposed noised model improved the performance of UNet by 3.09 dB.

[1] https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php

[2] https://noise.visinf.tu-darmstadt.de/benchmark/

Table 1: Performance comparison of denoising methods on SIDD benchmark.

	Raw		sRGB	
Method	PSNR	SSIM	PSNR	SSIM
BM3D	45.52	0.980	30.95	0.863
DnCNN	43.30	0.965	28.24	0.829
UNet	45.69	0.976	32.93	0.854
GAN + DnCNN (ours)	45.55	0.980	32.05	0.809
GAN + UNet (ours)	**48.78**	**0.986**	**35.78**	**0.919**

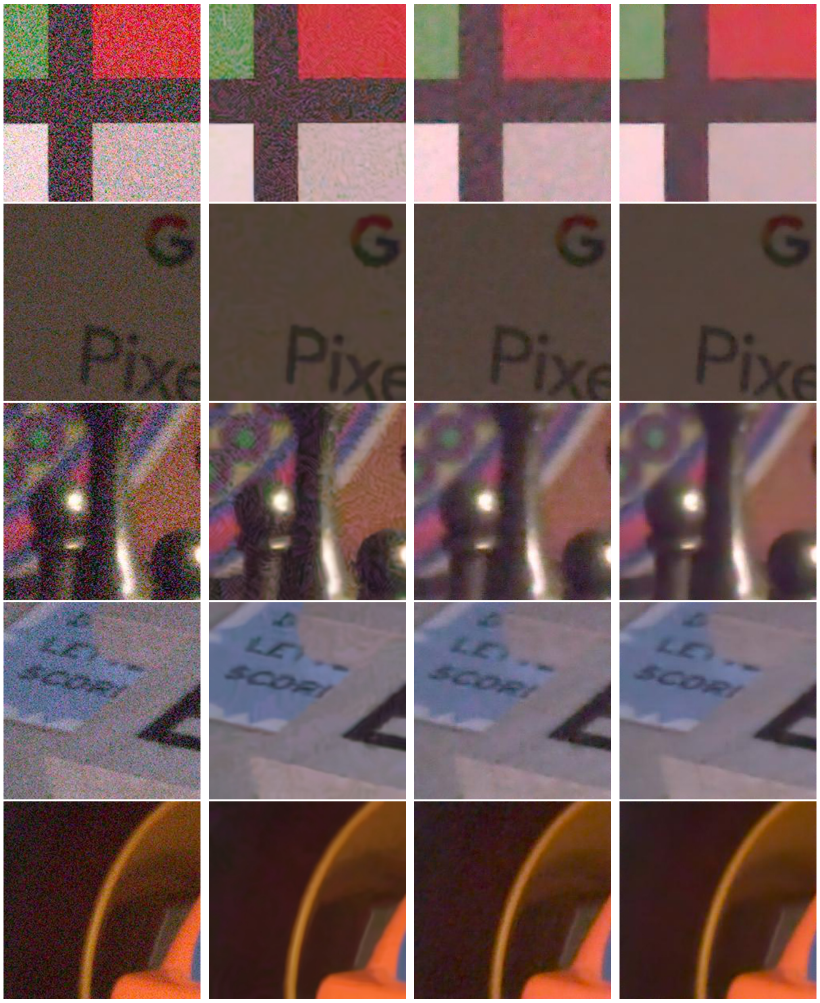
Table 2: Performance comparison of denoising methods on DND benchmark.

	Raw		sRGB	
Method	PSNR	SSIM	PSNR	SSIM
WNNM	46.3	0.9707	37.56	0.9313
EPLL	46.31	0.9679	37.16	0.9291
BM3D	46.64	0.9724	37.78	0.9308
BM3D + VST	47.15	0.9737	37.86	0.9296
DnCNN	47.37	0.9760	38.08	0.9357
N3Net	47.56	0.9767	38.32	0.9384
UPI	48.89	0.9824	40.17	**0.9623**
GAN + DnCNN (Ours)	47.46	0.9769	38.34	0.9418
GAN + UNet (Ours)	**49.04**	**0.9827**	**40.21**	0.9600

Table 2 shows the quantitative results for the DND benchmark [7]. The proposed model performs better than classic methods (e.g., BM3D, WNNM, and EPLL[26]). Compared with other deep learning-based method that using same baseline (DnCNN [5]), our model consistently yields higher PSNR and SSIM values on both raw data and after conversion to s-RGB space. Notably, the proposed model outperforms state-of-the-art method UPI [10] which used the same baseline (UNet) by 0.15dB in denoising raw image.

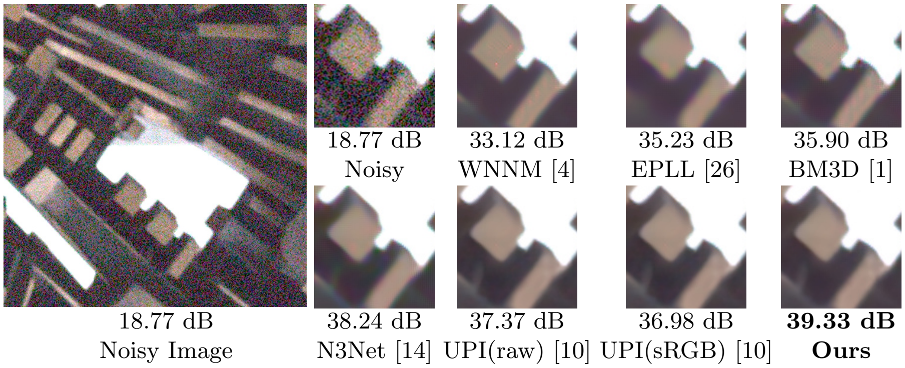
Qualitative results: Fig. 7 and Fig. 8 show the output of our model and some state-of-the-art denoisers with the SIDD and DND benchmarks.

Ablation studies: Table 3 shows ablation studies for various configurations of our model. In the table, “Gaussian, blind” indicates the training data were generated with AWGN and applied to a blind denoiser; “Non-GAN” indicates that instead of using the GAN to produce noisy images, we added noise to the input images using the procedure given in [10]; and “GAN, blind” indicates that we used the GAN noise model but did not add the noise level to the network input. Using the noise level information improves the denoising performance by 3.7% (PSNR) and 6.7% (SSIM) compared with blind denoising. The GAN noise



(a) Noisy (b) BM3D (c) DnCNN+ (d) UNet+

Fig. 7: Qualitative comparison of denoising methods. First column: noisy images; second column: images denoised using BM3D ($\sigma = 50$); third column: images denoised using DnCNN and GAN-based noise model; fourth column: images denoised using UNet and GAN-based noise model. Zoom in for a better view.


18.77 dB Noisy | 33.12 dB WNNM [4] | 35.23 dB EPLL [26] | 35.90 dB BM3D [1]

18.77 dB Noisy Image | 38.24 dB N3Net [14] | 37.37 dB UPI(raw) [10] | 36.98 dB UPI(sRGB) [10] | **39.33 dB Ours**

Fig. 8: The results of denoising raw image from DND [7] dataset. Zoom in for a better view.

Table 3: Ablation study on SIDD benchmark. The number in parenthesis is the relative improvement to the our best performing model. To compute the relative improvements, we change PSNR to RMSE ($RMSE = \sqrt{10^{-PSNR/10}}$) and SSIM to DSSIM ($DSSIM = (1 - SSIM)/2$) and then calculate the relative reduction in error.

	Raw		sRGB	
Method	PSNR	SSIM	PSNR	SSIM
Gaussian, blind	45.69 (29.9%)	0.976 (41.7%)	32.93 (28%)	0.854 (44.5%)
Non-GAN	47.76 (11.1%)	0.983 (17.6%)	34.39 (14.8%)	0.897 (21.4%)
GAN, blind	48.45 (3.7%)	0.985 (6.7%)	35.37 (4.6%)	0.913 (6.9%)
GAN, non-blind	48.78 (0.0%)	0.986 (0.0%)	35.78 (0.0%)	0.919 (0.0%)

model increases the performance by 11.1% (PSNR) and 17.6% (SSIM), showing the usefulness of our proposed noise model.

5 Discussion and Conclusion

5.1 Noise Level Estimation

In section 4.4 we present the denoising performance with both blind and non-blind denoisers. It is noted in [27] that incorporating information about the precise noise level may boost the performance of a denoiser. In the UNet denoiser, we added the random noise level as an additional network input. During the testing phase, the noise level was provided in the metadata. However, in real-world applications, the noise level is generally not available. We think that using a noise level estimation algorithm can further improve our results, and we will investigate this in future work.

5.2 Conclusion

In the present paper, we introduce a GAN-based model for real noise estimation in raw image data. The model consists of an "unprocessing" process to convert images from the sRGB space to raw image data. By using generated noise, we can generate a large amount of data for training a deep DNN. Although our approach contains some limitations, such as relying on prior statistics stored in the metadata and approximations of the noise level, the experimental results show the effectiveness of our data augmentation approach for real-world image denoising.

References

1. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16** (2007) 2080–2095
2. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing **54** (2006) 4311–4322
3. Zhang, L., Dong, W., Zhang, D., Shi, G.: Two-stage image denoising by principal component analysis with local pixel grouping. Pattern recognition **43** (2010) 1531–1549
4. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 2862–2869
5. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26** (2017) 3142–3155
6. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 1692–1700
7. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1586–1595
8. Anaya, J., Barbu, A.: Renoir-a benchmark dataset for real noise reduction evaluation. Journal of Visual Communication and Image Representation (2018) 144–154
9. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1712–1722
10. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11036–11045
11. Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 3155–3164
12. Park, S.H., Kim, H.S., Lansel, S., Parmar, M., Wandell, B.A.: A case for denoising before demosaicking color filter array data. In: 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, IEEE (2009) 860–864

13. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG) **35** (2016) 1–12
14. Plötz, T., Roth, S.: Neural nearest neighbors networks. In: Advances in Neural Information Processing Systems. (2018) 1087–1098
15. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing **27** (2018) 4608–4622
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134
18. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4681–4690
19. Zhu, F., Chen, G., Heng, P.A.: From noise modeling to blind image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 420–429
20. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
21. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. (2017) 5767–5777
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
24. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
25. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In: Proceedings of the international conference on Multimedia information retrieval. (2010) 527–536
26. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: 2011 International Conference on Computer Vision, IEEE (2011) 479–486
27. Liu, X., Tanaka, M., Okutomi, M.: Single-image noise level estimation for blind denoising. IEEE transactions on image processing **22** (2013) 5226–5237