NEAR-INFRARED IMAGE GUIDED NEURAL NETWORKS FOR COLOR IMAGE DENOISING

Xuehui Wang[⋆†] *Feng Dai*[⋆] *Yike Ma*[⋆] *Junbo Guo*[⋆] *Qiang Zhao*[⋆] *Yongdong Zhang*[⋆]

[⋆] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, China
[†] University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Noisy color image and guided near-infrared (NIR) image can be jointly employed to eliminate noise and enhance details. Existing methods mostly rely on explicit designed filters and hand-crafted objective function optimization. These methods usually introduce erroneous structures from guidance signal. Besides, they are time-consuming and not suitable for real time applications. In this paper, we come up with a learning based method. The noisy color image and NIR image are fused, then fed into a fully convolutional neural network. The network learns a directly map from degraded image to restored sharp image. Our architecture can effectively eliminate image noise and transfer detail structure from guided image. Our trained network accepts any resolution of input image and runs in constant time. We evaluate the presented approach on both synthetic and real images. Results show that our approach outperforms the state-of-art methods.

Index Terms— Denoise, color image, NIR image, convolutional neural networks

1. INTRODUCTION

Under low light condition, images are usually captured with high ISO setting. These images could be very noisy, as low photon count amplifies both shot noise and circuit noise. There are a large variety of denosing algorithms to restore clear image [1, 2, 3]. With the popularity of computational imaging, more novel imaging systems are developed to solve this problem. One caterory of system is to capture multiple modal signal, e.g. color image and near-infrared (NIR) image at the same time [4]. Regular silicon photo detectors are sensitive to near-infrared spectrum, thus NIR image records structure details with less noise. The color image and NIR image are highly correlated when taken from the same camera setting. It enables a system to jointly restore noisy color images with guidance signal.

The main challenge of guided denoising is to enhance degraded target image while avoiding transferring nonexistent structures from guidance signal to the target image. Some methods explicitly construct filters by considering neighboring pixels in guidance image [5, 6]. However, when local structures in two images are not consistent, these approaches might introduce texture-copying artifacts. Other methods are based on global optimization [7, 8]. These frameworks typically exploit common structures in both noisy and guidance images and minimize a global objective function iteratively. These approaches often utilize task-specific functions and are typically time-consuming. Recently, deep learning based methods are developed to solve this problem. Li et.al [9] proposes a network architecture for joint image filtering. Kim et.al [10] combines traditional optimization method with learning method for image restoration.

In this paper, we propose a dedicated fully convolutional network for color image denoising with dark-flashed NIR image as guidance signal. We make the following contributions in our paper: (1) Our framework concatenates noisy color and NIR image as input. The two modal signal are early fused, and it is beneficial for learning filters to eliminate noise and transfer structures from guidance signal. (2) Our architecture adopts multiple resblocks after stride convolution. Stride convolution can enlarge receptive field and reduce space size of feature maps at the same time. Resblocks ease the backpropagation of gradient during optimization. (3) In the end of our network, we replace deconvolution layers with upsampling layers to avoid checkerboard artifacts. Finally, we test our approach on both synthetic and real captured datasets. Results demonstrate that our model outperforms other methods on both synthetic and real noisy images.

2. METHODS

2.1. Framework Design

We present a NIR image guided fully convolutional neural network for color image denoising. Our framework accepts multiple modal signal as input. It concatenates noisy color and NIR image in the beginning. The fused multiple modal signal is benefical for learning effective filters. Then, we increase the depth of the network with residual blocks to enlarge the effective receptive field size. In the end, we replace

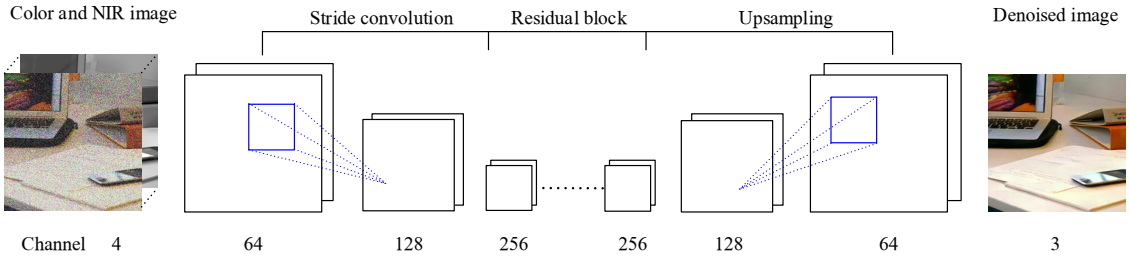


Fig. 1: The proposed network architecture for guided image denoising.

deconvolution layer with upsampling layer to avoid checkerboard artifacts. We adopt the residual formulation and batch normalization for fast converge.

A. Multiple modal fusion. To solve the image denoising problem, we adopt an implementation with multiple modal signal, i.e. noisy color image and NIR image. The two modal images are concatenated at the beginning and then fed into our fully convolutional neural networks. These two signal are early concatenated, and corresponding pixels are registered at the beginning. It is beneficial for learning effective filters to eliminate noise and transfer mutual structures. Experiments in Sec. 3.4 show that the earlier the two modal image are combined, the better restoration performance is achieved.

B. Deeper network and larger receptive field. It has been verified that the receptive field size of image denoising neural networks has a close relationship with the restoration performance [11, 12]. In [13], the authors analyze the effective patch size of several state-of-art denoising methods. These methods do filtering over the entire image, or using non-local information. However, there is a tradeoff between effectiveness and efficiency and we cannot build our network as deeper as possible. We adopt stride convolution in the first two convolutional layers which doubles receptive field size and halves feature map size at the same time. It reduces computation complexity on a large scale.

Residual block [14] has been shown to be helpful for the propagation of gradient in neural networks and improves the performance of visual tasks. We adopt a slightly different design [15]. Each block contains two 3×3 kernel convolution layers, and the nonlinearity following the addition is removed. It increases receptive field size of each neuron with respect to the input pixels, and allows the network not to occupy too much memory in the bottleneck.

C. Nearest upsampling. In order to recovery a lower resolution image to a higher one using neural networks, we generally do this with a deconvolution operation. Unfortunately, deconvolution easily creates uneven overlap and there are checkerboard artifacts on feature maps. As pointed in [16], we replace deconvolution with a nearest upsampling followed by a convolution layer. Nearest upsampling layer can create uniform overlap distribution among feature maps and effectively avoid checkerboard artifacts.

2.2. Network Architecture.

The final network architecture is shown in Fig. 1. The two modal signal are earlily fused as input. Our network begins with a convolution layer with 7×7 filter size. The first convolutional layer has a large filter size to extract low-level features, e.g. small curves and edges. The next are two stride-2 convolution blocks. In each block, feature maps are progressively halved spatially, while doubling in the channel dimension. We discard using pooling layer to reduce spatial dimension, as it losses signal information which is essential for image filtering. In the middle are several residual blocks. To recovery the spatial resolution of input image, two upsampling layers immediately follow the residual blocks. A convolutional layer comes along with each upsampling layer to half channel dimensions.

In image restoration applications, it usually requires restored images have the same size as input. It needs careful design to avoid boundary artifacts. We directly pad proper number of zeros in convolution layers to assure that output feature map has the same size. Results show this padding strategy does not cause any boundary artifacts. All convolutional layers are followed by a normalization layer and a nonlinearity. Batch normalization [17] has been shown to help training, as it alleviates internal covariate shift in middle layer features. In our network, we use the Rectified Linear Unit (ReLU) [18] as the point-wise nonlinearity.

2.3. Optimization

For model training, the network must be trained end-to-end, pixel-to-pixel. Dense prediction using CNN is first used in [19], similarly, we make our network fully convolutional to operate on variable-resolution images. A large dataset including color image and NIR image pairs is used for training. We test different noise levels to verify the effectiveness of our model. We also capture real color noisy image and NIR image under low light condition to test our model.

The network is trained on a set of image samples, $D = \{I_c, I_n, I_t\}$. $I_c \in R^{H \times W \times 3}$ is a noisy color image, $I_n \in R^{H \times W \times 1}$ is a less noisy NIR image and I_t is the noisy-free image. The parameters θ in the network F are optimized on

Table 1: Quantitative comparisons with the state-of-art methods in PSNR (BM3D [2], K-SVD [1] use single input).

Noise level	$\sigma = 50$	$\sigma = 64$	$\sigma = 96$
BM3D [2]	26.59	24.31	19.39
K-SVD [1]	24.56	24.47	22.29
Cross field [7]	25.19	23.61	19.45
Joint filter [9]	23.35	21.52	21.11
Our Method	**28.83**	**27.80**	**26.97**

the training set to minimize the following objective function:

$$\theta^* = \arg\min_{\theta} E_D[L(I_t, F(I_c, I_n; \theta))],$$

where loss function L describes how close the network output and the ground truth are.

The choice of an appropriate loss function L plays an important role in network optimization. We select to minimize the mean-squared error (MSE). We also experiment with other loss functions, e.g., perceptual losses that match feature activations [20]. We find that perceptual loss does not increase perceptual fidelity, as the semantic information is not useful in image denoising task.

3. EXPERIMENTS

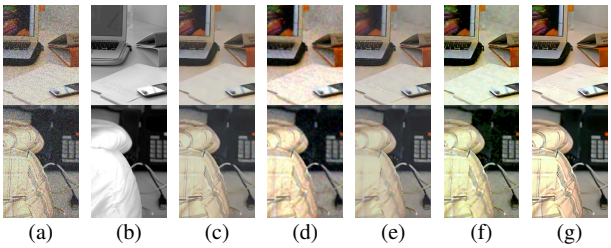
3.1. Dataset

Synthetic dataset. We train our networks with the IVRG RGB-NIR dataset [21]. It consists of 477 pair of images which are captured using separate exposures from modified SLR cameras. We use a random subset of 400 images for training, and the rest of 77 images for testing. To simulate the scenario of low-light photography, we generate degraded images by adding Gaussian noise to the color images. The variance σ of noise is set to [50, 64, 96] three levels.

Real captured dataset. Noise in real images is not *i.i.d.* Gaussian [22]. To evaluate our approach in practical scenes, we setup a low-light environment to capture noisy color image and NIR image under NIR illumination. The visible light intensity is controlled with a projector. Two settings are used to capture low-light image and noise-free image. We setup various indoor scenes and collect about 120 pair of real low light images. 90 of them are used for model finetune, and the left is for testing.

3.2. Training

We use the PyTorch toolbox [23] to train our model. We randomly crop image patch from the whole training dataset. Pixel values are scaled to range [0, 1]. We train our network with a batch size of 8 for 15 epochs over the training dataset. Adam gradient-based algorithm [24] is used for optimization. Experiments show it can produce good results and



(a) (b) (c) (d) (e) (f) (g)

Fig. 2: Synthesis denosing results with Gaussian noise variance $\sigma = 65$. (a) Noisy image, (b) NIR image, (c)-(g) are results of BM3D [2], K-SVD [1], Cross field [7], Joint filter [9], and Our method.

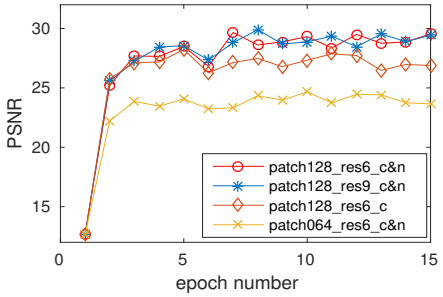


Fig. 3: Performance of model variants during training.

is much easier to converge. The learning rate is initialized with 2×10^{-3}. We decay the learning rate by a factor of 0.1 every 5 epochs. The whole training procedure lasts about 20 hours. After training, the model performance is evaluated on the test dataset. As our network is fully convolutional, it can handle images with variable resolution.

3.3. Synthetic Images Results

We show the quantitative denoising results of various methods in this part. Table 1 lists PSNR results of different methods on our test images under three noise level. The best restoration result for each noise level is bolded. Figure 2 shows visual results. As we can see, our method achieves the best performance among all competing methods.

From Fig. 2, we can see that BM3D [2] tends to produce over-smooth edges and textures. While preserving high PSNR value, K-SVD [1] generates artifacts within image. After tried various parameter, Cross-field method [7] achieves reasonable good result in low noise level. When the noisy level is high, it fails to balance noise removal and structure transfer. Joint filter [9] method has limit model capacity, thus the hue of image is not corrected, and there are still large artifacts block. In contrast, our guided image denoising method not only recovers sharp edges and textures but also yields visually pleasant result in the whole image. Specifically, in the

Fig. 4: Check-boarder artifacts.

first image, the words on paper is remained and there is no noise left in smooth regions.

3.4. Model Ablation

We also investigate the contributions of various design on the effectiveness of restoration. It includes input modal number, network depth, patch size, and sampling layer. Figure 3 shows how the performance varies as training epoch number grows.

One modal vs Two modal. To illustrate guidance signal is crucial for image denoising, we have trained our network with only noisy color images. NIR image is excluded from network inputs. The result is shown in Fig. 3 as *'patch128-res6-c&n'*, in which *'patch128'* is patch size, *'res6'*' is resblock number and *'c&n'* denotes color and NIR images. We can see that there is a large gap between single modal and two modal implementation.

Network depth. We have tried different resblock numbers in the bottleneck part. In Fig. 3, the performance of network with 6 and 9 resblocks, denoted as *'patch128-res6-c&n'* and *'patch128-res9-c&n'*, is shown. The network with 9 resblocks does not show prominent improvement but cost more training time.

Patch size. We have experimented with different training patch size, e.g. 64 and 128 pixels. In Fig. 3 *'patch128-res6-c&n'* and *'patch064-res6-c&n'*, we can see larger patch size achieves better result. Our network has a receptive field of ~ 128 pixels and bigger patch size provides more non-local information for model to learn filters.

Deconvolution vs Upsampling. We also have trained our network with both deconvolution layer and upsampling layer on our dataset. We extract feature activations after these layers and select an patch with largest variance. An example patch is shown in Fig. 4. The orgin noisy color image and NIR image are also listed. We can see after deconvolution layer, there are clearly checkerboard effects and features after nearest upsampling is much more natural.

3.5. Real Captured Images Results

To make our work more solid, we experiment the proposed method with real captured low light images, in which the noise type and level are unknown. We evaluate all methods on real captured image under low light condition. Since BM3D [2] and K-SVD [1] works for non-blind denoising, we have tested various parameters and selected the best result.

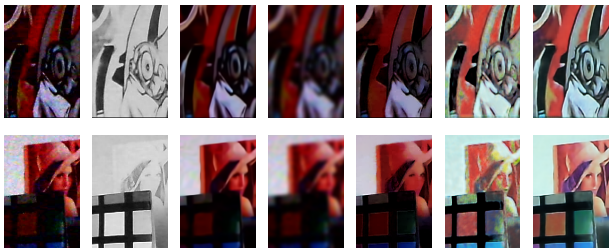


Fig. 5: Denoised result of real captured low light noisy image. (a) Noisy image, (b) NIR image, (c)-(g) are results of BM3D [2], K-SVD [1], Cross field [7], Joint filter [9], and Our method.

Sample results are shown in Fig. 5. Single image denoising methods, like BM3D [2] and K-SVD [1] produce too smooth results and lose sharp textures. Cross-field [7] has difficuties in learning mutual structure. We can see the nose of Lina does not exist in NIR image, thus this part in restored image is lost. Joint filter [9] method recovers sharp edges and textures, however, the visual effect is not satisfied. In contrast, our method yields visually pleasant result in whole image. The noise in degraded color image is clearly removed and sharp content is also transferred from guidance signal. Our network can generalize well to true noisy images.

4. CONCLUSION AND FUTURE WORKS

In this work, we present a NIR signal guided fully convolutional neural network for color image denoising. Instead of explicit filter construction or global optimization, our model implicitly learns mutual structures from both noisy and guidance image to eliminate noise in the target color image. We design a novel architecture which accepts multiple modal inputs and has a large receptive field to make use of non-local information in input images. The residual blocks in bottleneck improve the non-linearity of our model and ease the backpropagation of gradient. Stride convolution and upsampling layer reduce memory usage during training while generating output with the same size as input. Upsampling layer avoid checkboard artifacts. We conduct experiments on various settings. Results show that our proposed model is efficient and achieves superior performance against the state-of-art algorithms on both synthetic and real captured images.

5. ACKNOWLEDGEMENTS

This work is supported by NationalKeyR&DProgramofChina (2016YFB0800403), National Natural Science Foundation of China (61525206, 61702479, 61771458), Beijing Municipal Science and Technology Project (Z171100000117010), and the Science and Technology Service Network Initiative of the Chinese Academy of Sciences (KFJ-STS-ZDTP-070).

6. REFERENCES

[1] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE TIP*, vol. 15, no. 12, pp. 3736–3745, 2006.

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE TIP*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE International Conference on*. IEEE, 2009, pp. 2272–2279.

[4] Dilip Krishnan and Rob Fergus, "Dark flash photography," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 96–1, 2009.

[5] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE PAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.

[6] Xiaopeng Zhang, Terence Sim, and Xiaoping Miao, "Enhancing photographs with near infra-red images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[7] Qiong Yan, Xiaoyong Shen, Li Xu, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Jiaya Jia, "Cross-field joint image restoration via scale map," in *ICCV*, 2013, pp. 1537–1544.

[8] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia, "Mutual-structure for joint filtering," in *ICCV*, 2015, pp. 3406–3414.

[9] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep joint image filtering," in *ECCV*. Springer, 2016, pp. 154–169.

[10] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn, "Deeply aggregated alternating minimization for image restoration," *arXiv preprint arXiv:1612.06508*, 2016.

[11] Viren Jain and H Sebastian Seung, "Natural image denoising with convolutional networks," pp. 769–776, 2009.

[12] Harold C Burger, Christian J Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *CVPR*. IEEE, 2012, pp. 2392–2399.

[13] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[15] Sam Gross and Michael Wilber, "Training and investigating residual nets," *Facebook AI Research, CA.[Online]. Avilable: http://torch.ch/blog/2016/02/04/resnets. html*, 2016.

[16] Augustus Odena, Vincent Dumoulin, and Chris Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*. Springer, 2016, pp. 694–711.

[21] Clément Fredembach and Sabine Süsstrunk, "Colouring the near-infrared," in *Color and Imaging Conference*. Society for Imaging Science and Technology, 2008, vol. 2008, pp. 176–182.

[22] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE TIP*, vol. 17, no. 10, pp. 1737–1754, 2008.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

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