R3L

Connecting Deep Reinforcement Learning To Recurrent Neural Networks For Image Denoising Via Residual Recovery

Zhang, Rongkai ; Zhu, Jiang ; Zha, Zhiyuan ; Dauwels, Justin ; Wen, Bihan

DOI
10.1109/ICIP42928.2021.9506323

Publication date
2021

Document Version
Final published version

Published in
2021 IEEE International Conference on Image Processing (ICIP)

R3L: CONNECTING DEEP REINFORCEMENT LEARNING TO RECURRENT NEURAL NETWORKS FOR IMAGE DENOISING VIA RESIDUAL RECOVERY

Rongkai Zhang[1], *Jiang Zhu*[1], *Zhiyuan Zha*[1], *Justin Dauwels*[2] *and Bihan Wen*[1*]

[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]Department of Microelectronics, Delft University of Technology, Netherlands

2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506323

ABSTRACT

State-of-the-art image denoisers exploit various types of deep neural networks via deterministic training. Alternatively, very recent works utilize deep reinforcement learning for restoring images with diverse or unknown corruptions. Though deep reinforcement learning can generate effective policy networks for operator selection or architecture search in image restoration, how it is connected to the classic deterministic training in solving inverse problems remains unclear. In this work, we propose a novel image denoising scheme via Residual Recovery using Reinforcement Learning, dubbed R3L. We show that R3L is equivalent to a deep recurrent neural network that is trained using a stochastic reward, in contrast to many popular denoisers using supervised learning with deterministic losses. To benchmark the effectiveness of reinforcement learning in R3L, we train a recurrent neural network with the same architecture for residual recovery using the deterministic loss, thus to analyze how the two different training strategies affect the denoising performance. With such a unified benchmarking system, we demonstrate that the proposed R3L has better generalizability and robustness in image denoising when the estimated noise level varies, comparing to its counterparts using deterministic training, as well as various state-of-the-art image denoising algorithms.

***Index Terms*—** Recurrent Neural Network, Deep Reinforcement Learning, Image Denoising, Residual Recovery.

1. INTRODUCTION

Image denoising is one of the most fundamental inverse problems, which aims to estimate the underlying clean $\mathbf{x}$ from its noisy observation $\mathbf{y}$, which is corrupted with noise $\mathbf{n}$ as:

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \,. \tag{1}$$

Assuming $\mathbf{n}$ to be the additive white Gaussian noise (AWGN), it follows the normal distribution, *i.e.*, $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$. Besides improving the image visual quality, it is also a necessary preprocessing step for many high-level vision tasks such as classification [1], segmentation [2], object detection [3] and tracking [4].

Classic image denoising algorithms are based on analytical models, *e.g.*, image non-local similarity [5, 6, 7], transform-domain sparsity [8, 9], etc. More recently, deep learning has demonstrated remarkable results in image denoising by training the highly flexible neural networks with deterministic loss functions using an end-to-end approach [10, 11, 12, 13]. While most deep denoisers exploit feed-forward convolutional neural networks (CNNs) [12], the models usually involve a huge amount of trainable parameters leading to high memory complexity. Alternatively, some recent works exploit recurrent neural networks (RNNs) with shared module parameters. For example, the non-local recurrent network (NLRN) [13] achieved both high parameter efficiency and denoising performance.

Comparing to the end-to-end supervised deep learning, few works to date exploited deep reinforcement learning (DRL) for image denoising. Some pilot works trained a separate policy network for operator selection [14, 15] or architecture search [16, 17] to assist image denoising. However, it is unclear how DRL can be "directly" applied to inverse problems, *e.g.*, how effective is the denoising network trained via DRL? To the best of our knowledge, no work to date has benchmarked DRL with supervised deep learning with deterministic loss functions in image denoising.

To this end, we propose a novel image denoising scheme via Residual Recovery using Reinforcement Learning (R3L) for image denoising. We show that the proposed R3L is equivalent to a RNN denoiser trained using a stochastic reward, which provides a unified framework to compare DRL to other RNN-based image denoising schemes. To benchmark the effectiveness of DRL, we train a recurrent neural network with the same architecture as our R3L model for residual recovery using supervised learning with a deterministic mean square error, called R3N. The experiments show that the proposed R3L achieved more reliable denoising results when the estimated noise levels (*i.e.*, noise standard deviation σ) of degraded images deviate from the oracle. The average denoising PSNRs (over varied noise estimations) using our R3L outperform those by R3N as well as many state-of-the-art denoising algorithms.

2. RELATED WORK

Image denoising methods are classified into two categories: prior-based methods and learning based methods. Many classical methods, such as BM3D [6] and WNNM [7], are based on effective priors, and some of them applied the denoising operator recursively [7]. On the other hand, learning-based methods utilized more flexible models such as deep neural networks [10, 12, 18]. Though deep denoising models lead to superior image restoration, most of them involve a huge amount of parameters. One solution to enhance memory efficiency is applying a lighter neural network recursively, which results in many successful frameworks based on RNN or DRL. We provide a summary of RNN and DRL algorithms for image denoising. Table 1 summarizes the representatives of image denoising methods of different categories, as well as the proposed R3L.

2.1. RNN for Image Denoising

Deep RNNs have been widely applied for image denoising. Chen *et al.* [19] first used a deep RNN which exploits temporal-spatial information for video denoising. Putzky *et al.* [20] proposed a learning

*Bihan Wen (bihan.wen@ntu.edu.sg) is the corresponding author.

Table 1: Comparison between various image denoisers, including the proposed R3L and other existing methods.

Methods	Trainable kernels	Residual learning	Recursion	DRL
BM3D [6]				
WNNM [7]			✓	
DnCNN [12]	✓	✓		
NLRN [13]	✓	✓	✓	
pixelRL [15]			✓	✓
R3L	✓	✓	✓	✓

framework, dubbed Recurrent Inference Machines (RIM), in which they train a RNN to learn an inference algorithm for solving inverse problems. Liu *et al.* [13] proposed NLRN which incorporates the non-local operations into an RNN for image restoration achieving the state-of-the-art results. However, most of the RNN-based denoising models are trained over a corpus of images containing the similar noise distribution, using the deterministic loss function (*e.g.*, mean square error), thus hard to generalize to complex and inaccurately estimatesd noise in practice.

2.2. DRL for Image Denoising

DRL has recently gathered considerable interest showing great promise in many applications [21], including image processing tasks. Yu *et al.*[14] firstly attempt to apply DRL to learn a policy to select suitable operators from a pre-defined toolbox to progressively restore corrupted images. Their improved version [16] can dynamically select an appropriate route for different image regions in a multi-path CNN, to perform spatial-varying image denoising. Furuta *et al.*[15] proposes pixelRL, the first framework to do a pixel-wise restoration. Most of DRL based methods still rely on manually designed filters. What the DRL agent learns is the order to apply filters instead of directly modifying the pixel values, *i.e.*, residual recovery. Therefore, it remains unclear how DRL approaches to image denoising relate to other learning based methods.

3. PROPOSED R3L METHOD

3.1. Residual Recovery as Markov Decision Process

Residual recovery is commonly used in deep image denoising, which aims to obtain the residual image of the noisy input relative to the ground truth. As removing a residual can be considered as sequentially removing several inter-residuals, residual recovery is a sequential decision problem. Therefore, we modeled the denoising problem via residual recovery as a Markov Decision Process (MDP), which can be solved using DRL.

At each state t ($t = 0$ denotes the initial state) of denoising, taking the noisy image ($t = 0$) or the denoised estimate ($t \geq 1$) from the previous state as the input $I^t \in \mathbb{R}^N$, the DRL agent follows a policy π to output the probability $P(a_i^t|I^t)\ \forall i$. Here a_i^t denotes the estimated residual of the ith pixel ($1 \leq i \leq N$) at the state t. We apply a deep neural network to construct the policy π, denoted as the policy network with the trainable parameter θ_π. A is the action set, which consists of all discrete values in a predefined range, and $a_i^t \in A$. The estimated image is updated to I^{t+1} by applying the output actions, and the agent can obtain a reward r_i^t for each pixel. The denoising process repeats until the termination stage n, and outputs the final denoised image. The probability of an action trajectory J_i for each pixel i, denoted as $P(J_i|I^0, \theta_\pi)$, is calculated as:

$$\begin{aligned} P(J_i|I^0,\theta_\pi) &= P(a_i^1|I^0,\theta_\pi)P(a_i^2|a_i^1,I^0,\theta_\pi) \\ &\quad ...P(a_i^t|a_i^{t-1}...,a_i^1,I^0,\theta_\pi) \\ &= \prod_{t=1}^{T} P(a_i^t|J_i^{t-1},I^0,\theta_\pi) \end{aligned} \tag{2}$$

where $J_i \triangleq \left\{a_i^1, a_i^2, \ ..., \ a_i^n\right\}$.

Following the common setting in DRL, we use the long-term discounted reward $R_i^t(J_i)$ to evaluate a policy at the state i, which is defined as:

$$R_i^t(J_i) \triangleq r_i^t + \gamma r_i^{t+1} + \gamma^2 r_i^{t+2} + ... + \gamma^{n-t} r_i^n \tag{3}$$

Here, γ^j denotes the jth power of the discount factor $0 < \gamma < 1$, and r_i^t denotes the reward for pixel i at stage t.

The DRL agent can explore different trajectories towards learning the optimal policy π^*. Following π^*, the agent selects the optimal action at each state with the highest probability by maximizing the expectation of $R_i^0(J_i)$ as:

$$\pi^* = \underset{\pi}{\text{argmax}} P(J_i|I^0,\theta_\pi)R_i^0(J_i) \tag{4}$$

3.2. Proposed R3L Framework

Inspired by [15], we apply the fully convolutional network (FCN) based asynchronous advantage actor-critic (A3C) [22] framework in the proposed residual recovery reinforcement learning (R3L) scheme. We apply FCN as the encoder which is widely used and effective in image processing tasks for the pixel-level modification. We apply A3C with a policy network π and a value network V to make the training more stable and efficient [23].

The FCN-based encoder is denoted as E_{FCN}, which is shared by both π and V. E_{FCN} extracts the features of the input image I^t and outputs s^t, as the representation of state t. Taking s^t, the policy network π outputs the probability of selecting a certain residual value a_i^t for each pixel, and the value network outputs $V(s^t|\theta_v)$, which is the estimation of the long term discounted rewards R_i^t for each pixel. The reward r_i^t used in R3L for image denoising is defined as:

$$r_i^t \triangleq (x_i - I_i^{t-1})^2 - (x_i - I_i^t)^2 \tag{5}$$

where x denotes the clean image, and x_i denotes its i pixel. Without loss of generality, for convenience, we consider the one-stage learning case ($n = 1$) here. The gradients of the parameters of these two networks θ_π,θ_v are calculated as:

$$\begin{aligned} R_i^t &= r_i^t + \gamma V(s^{t+1}|\theta_v) \\ d\theta_v &= \nabla_{\theta_v} \frac{1}{N} \sum_{i=1}^{N} (R_i^t - V(s^t|\theta_v))^2 \\ d\theta_\pi &= -\nabla_{\theta_\pi} \frac{1}{N} \sum_{i=1}^{N} \log P(a_i^t|s^t,\theta_\pi)(R_i^t - V(s^t|\theta_v)) \end{aligned} \tag{6}$$

During training, the residual value a_i^t is sampled from a predefined range, *i.e.* $[-13, 13]$, according to the output from the policy network. In the testing phase, only the well-trained policy network is deployed and the residual value with the highest probability is greedily selected. The inference process is formulated as:

$$\begin{aligned} s^t &= \text{E}_{\text{FCN}}(I^t) \\ a^t &= \text{Greedy}(\pi(s^t|\theta_\pi^*)) \qquad t = 0, 1, 2, ..., T \\ I^{t+1} &= I^t + a^t \end{aligned} \tag{7}$$

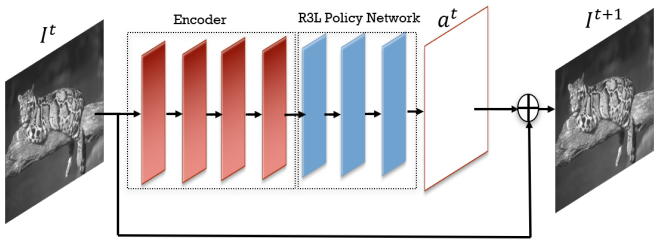

Fig. 1: The inference process of R3L at state t.

where Greedy$(\cdot)$ denotes the deterministic greedy sampling operator [24, 25], a^t denotes the residual image built by a_i^t $(1 \leq i \leq N)$, and T denotes the number of total stages. Here, we use $T = 5$ as a hyperparameter to balance the processing time and the performance. The inference process of R3L at state t is illustrated as Fig. 1.

3.3. Connection of R3L and RNNs

Theorem 1. *Greedily selecting the action with the highest policy from the output of policy network reduces the inference process of R3L to a RNN.*

Proof. In general, the recurrent inference process in an RNN is:

$$I^{t+1} = f_\theta(I^t) \quad \forall t \,, \tag{8}$$

where f_θ is the recurrent module which is parameterized by θ. Based on (7), the inference process of R3L follows (8), with the corresponding module f_θ as the following form:

$$f_\theta(I^t) = I^t + \text{Greedy}(\pi_\theta(\text{E}_{\text{FCN}}(I^t))). \tag{9}$$

Theorem 1 shows that the inference process of R3L follows an RNN. However, the R3L model is trained using DRL using the stochastic reward with no hidden states. In RNNs, the same network will be applied recursively until the termination. Our R3L also exploits the same recursive property and benefits a high parameter efficiency from it. However, most RNN based methods mainly focus on learning the final residual in an end-to-end manner, which makes the learning outcome a deterministic one-to-one mapping from the noisy input to the residual. R3L makes the solution a stochastic combination of different inter-residuals in a certain order rather than a deterministic mapping, and therefore has more flexibility. Moreover, RNNs use hidden states to summarize the modifications in the previous stages. In R3L, we assume that the action only depends on the current state input, so no hidden states are needed.

3.4. Benchmarking R3L with R3N

Although R3L is connected to the existing learning based methods, since inference in the R3L is equivalent to applying an RNN, there is a lack of RNN based methods to do a fair comparison, because usually RNNs are combined with some other techniques and involve hidden states. To achieve a fair comparison and verify how the different training methods can help R3L, we propose a simplified RNN based benchmark named residual recovery RNN (R3N) for image denoising.

In the R3N, the input image I^t is encoded via E_{FCN} to s^t. Taking s^t as input, a RNN block RNN$(\cdot|\theta_R)$ outputs the residual res^t, and I^t is updated to I^{t+1} by adding the residual to it. The whole process is formulated as:

$$\begin{aligned} s^t &= \text{E}_{\text{FCN}}(I^t) \\ \text{res}^t &= \text{RNN}(s^t|\theta_R) \qquad t = 0, 1, 2, \ldots, T \\ I^{t+1} &= I^t + \text{res}^t \end{aligned} \tag{10}$$

and the gradient for the parameters of R3N is formulated as:

$$d\theta_R = \nabla_{\theta_R} \frac{1}{N} \sum_{i=1}^{N} (I_i^{T+1} - x_i)^2 \tag{11}$$

where $T = 5$ follows the same setting as R3L.

The inference process of R3N is basically the same as R3L shown in Fig.1, but with the policy network replaced by the RNN block. The specific design of the layers in R3N and R3L is summarized in Table 2. The numbers in the table denote the filter size, dilation factor, and output channels, respectively.

Table 2: Specific design of the layers

$\text{E}_{\text{FCN}}(I^t)$ (in both R3L and R3N)		Conv+ReLU	Conv+ReLU	Conv+ReLU	Conv+ReLU
		3x3, 1, 64	3x3, 2, 64	3x3, 3, 64	3x3, 4, 64
R3L	Policy	Conv+ReLU	Conv+ReLU	Conv+ReLU+Softmax	
		3x3, 3, 64	3x3, 2, 64	3x3, 1, |A|	
	Value	Conv+ReLU	Conv+ReLU	Conv	
		3x3, 3, 64	3x3, 2, 64	3x3, 1, 1	
R3N		Conv+ReLU	Conv+ReLU	Conv+tanh	
		3x3, 3, 64	3x3 ,2, 64	3x3, 1, 1	

4. EXPERIMENTS AND RESULTS

4.1. Experimental Settings

We use the BSD400 dataset as training images and BSD68 [26] dataset as test images to verify the performance of R3L and R3N on image denoising. We train the models with additional Gaussian noise and the noise levels are selected as $\sigma = 25$ and $\sigma = 35$. However, in practice, it is difficult to estimate the noise level exactly accurately, and the noise level can vary in a range. Hence, besides testing the performance when the estimated noise level is accurate, we also test the cases when the noise level is estimated wrongly. For the model trained with noise level $\sigma = 25$, we test their performance when the noise levels are σ = 15, 20, 25, 30 and 35. For the model trained with noise level $\sigma = 35$, we test their performance when the noise levels are σ = 25, 30, 35, 40 and 45. Following the same setting, we also test the performance of BM3D, WNNM and DnCNN as baselines. The results are measured in terms of peak signal-to-noise ratio (PSNR) and shown in the next section.

4.2. Results

Table 3 and Table 4 summarize the PSNR results of our proposed frameworks and several state-of-the-art denoising methods. It shows that though our proposed R3L does not perform the best when the estimation is accurate, it can outperform the baselines when the estimation is inaccurate. More specifically, for the cases when the estimation error is relatively large, for instance $\pm$ 10, R3L can still maintain a good denoising performance with a higher PSNR. It should be emphasised that R3L achieves these performance using far fewer parameters than DnCNN.

Fig.2 and Fig.3 demonstrate the visual quality of the denoised image for the different methods. It shows that when the noise level is

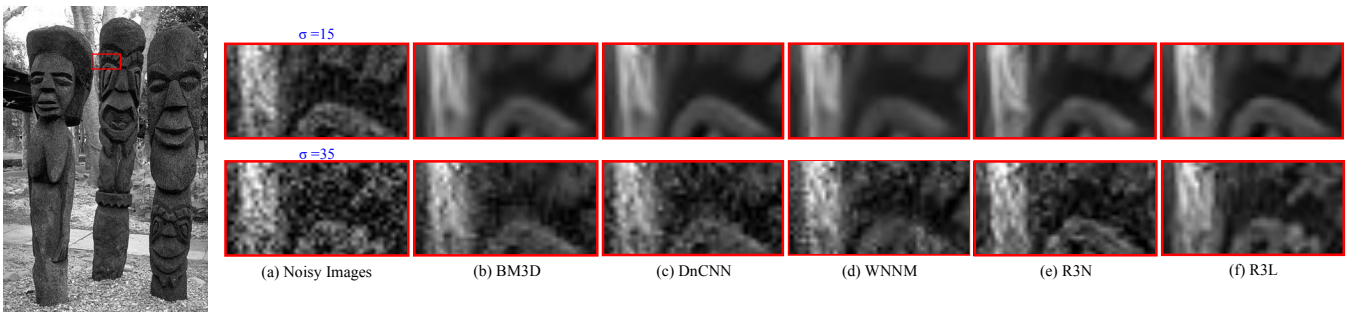

Fig. 2: Example of of denoising results using (b) BM3D [6], (c) DnCNN [12], (d) WNNM [7], (e) proposed R3N and (f) proposed R3L, with the zoom-in region highlighted. All the methods are set/trained with the estimated noise level $\sigma = 25$. The first row are results for noisy images with $\sigma = 15$ and the second row are results for noisy images with $\sigma = 35$.

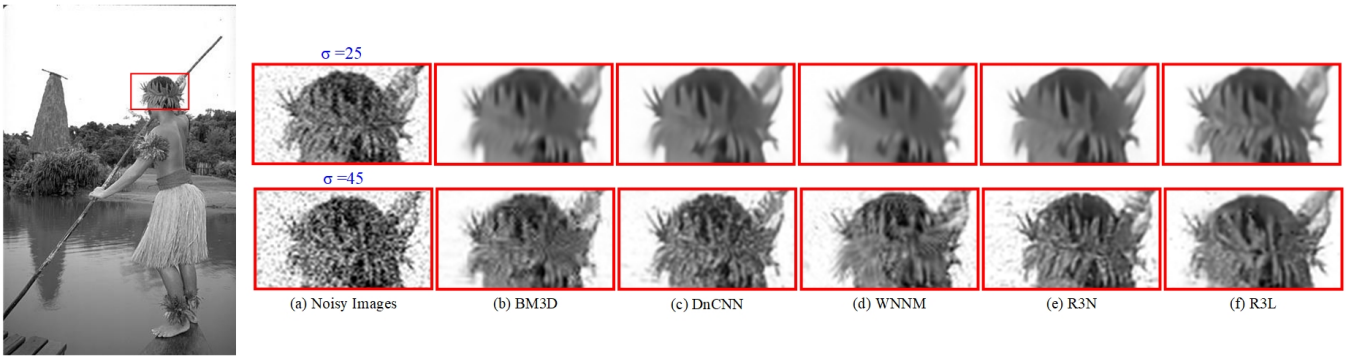

Fig. 3: Example of of denoising results using (b) BM3D [6], (c) DnCNN [12], (d) WNNM [7], (e) proposed R3N and (f) proposed R3L, with the zoom-in region highlighted. All the methods are set/trained with the estimated noise level $\sigma = 35$. The first row are results for noisy images with $\sigma = 25$ and the second row are results for noisy images with $\sigma = 45$.

Table 3: The average PSNR (dB) results of different methods. All the methods are set/trained with $\sigma = 25$. The best and the second best results are highlighted in red and blue respectively.

σ	BM3D [6]	WNNM [7]	DnCNN [12]	R3N	R3L
15	29.05	28.15	29.17	28.83	29.64
20	28.87	28.57	29.42	29.07	29.30
25	28.56	28.80	29.23	28.95	28.73
30	27.48	26.94	26.40	26.70	27.44
35	24.88	23.77	22.86	23.15	25.16
Average	27.77	27.25	27.41	27.34	28.05

Table 4: The average PSNR (dB) results of different methods.All the methods are set/trained with $\sigma = 35$. The best and the second best results are highlighted in red and blue respectively.

σ	BM3D [6]	WNNM [7]	DnCNN [12]	R3N	R3L
25	27.54	26.80	27.68	27.61	28.00
30	27.37	27.14	27.85	27.74	27.67
35	27.09	27.29	27.69	27.44	27.18
40	26.32	26.08	25.68	25.58	26.24
45	24.39	23.58	22.53	22.56	24.60
Average	26.54	26.18	26.28	26.19	26.74

overestimated oversmoothing is a critical issue, however, the images processed by R3L have more detailed textures remaining. Moreover, when the noise level is underestimated, the denoised images from R3N and the other baselines still have obvious noise remaining and may also involve some artifacts, but R3L can remove most of the noise with no additional artifacts resulting in a more natural and better visual quality.

The experimental results show that R3L is a more robust denoiser with high parameter efficiency. We explain the robustness from two points. First, compared with very deep end-to-end frameworks, R3L has less complexity, which endows R3L more generalization ability. Second, R3L is trained using a stochastic state-wise reward. The stochastic training process help R3L explore more different states and generate a more general policy than R3N, which is trained using a deterministic end-to-end loss.

5. CONCLUSION

In this paper, we propose a novel DRL based framework, namely R3L, to learn residual recovery for image denoising, and eventually close the gap of directly applying DRL for image denoising. We position R3L well by showing that R3L reduces to a deep RNN that is trained using the stochastic reward, and thus build the connection among R3L and the other methods. With the help of the proposed R3N, we benchmark R3L and verify how the different training method benefits R3L. The extensive experiment results reveals that R3L is a more robust denoiser with high parameter efficiency. Trained for a specific noise level, R3L can still be applied for a range of noise levels, which makes R3L a suitable framework for real-life scenarios, where the noise level estimation can be inaccurate.

6. REFERENCES

[1] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S Huang, "When image denoising meets high-level vision tasks: a deep learning approach," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 842–848.

[2] Ding Liu, Bihan Wen, Jianbo Jiao, Xianming Liu, Zhangyang Wang, and Thomas S Huang, "Connecting image denoising and high-level vision tasks via deep learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 3695–3706, 2020.

[3] S Milyaev and I Laptev, "Towards reliable object detection in noisy images," *Pattern Recognition and Image Analysis*, pp. 713–722, 2017.

[4] Taesik Na, Minah Lee, Burhan A Mudassar, Priyabrata Saha, Jong Hwan Ko, and Saibal Mukhopadhyay, "Mixture of pre-processing experts model for noise robust deep learning on resource constrained platforms," in *International Joint Conference on Neural Networks*, 2019, pp. 1–7.

[5] Yifei Lou, Paolo Favaro, Stefano Soatto, and Andrea Bertozzi, "Nonlocal similarity image filtering," in *Image Analysis and Processing – ICIAP 2009*, 2009, pp. 62–71.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, pp. 2080–2095, 2007.

[7] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

[8] Shujun Liu, Guoqing Wu, Hongqing Liu, and Xinzheng Zhang, "Image restoration approach using a joint sparse representation in 3d-transform domain," *Digital Signal Processing*, pp. 307–323, 2017.

[9] Bihan Wen, Saiprasad Ravishankar, and Yoram Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision*, pp. 137–167, 2015.

[10] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Toward convolutional blind denoising of real photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1712–1722.

[11] Yunjin Chen and Thomas Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1256–1272, 2017.

[12] 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*, p. 3142–3155, 2017.

[13] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang, "Non-local recurrent network for image restoration," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 1680–1689.

[14] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2443–2452.

[15] Ryosuke Furuta, Naoto Inoue, and Toshihiko Yamasaki, "Fully convolutional network with multi-step reinforcement learning for image processing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 3598–3605.

[16] Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy, "Path-restore: Learning network path selection for image restoration," 2019.

[17] Kyle Vassilo, Cory Heatwole, Tarek Taha, and Asif Mehmood, "Multi-step reinforcement learning for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 512–513.

[18] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, pp. 4608–4622, 2018.

[19] Xinyuan Chen, Li Song, and Xiaokang Yang, "Deep rnns for video denoising," in *Applications of Digital Image Processing*, 2016, p. 99711T.

[20] Patrick Putzky and Max Welling, "Recurrent inference machines for solving inverse problems," *arXiv preprint arXiv:1706.04008*, 2017.

[21] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[22] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[23] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, pp. 1057–1063.

[24] Wouter Kool, Herke van Hoof, and Max Welling, "Attention, learn to solve routing problems!," in *International Conference on Learning Representations*, 2019.

[25] Rongkai Zhang, Anatolii Prokhorchuk, and Justin Dauwels, "Deep reinforcement learning for traveling salesman problem with time windows and rejections," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[26] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 860–867.