ITERATIVE CONVOLUTIONAL NEURAL NETWORK FOR NOISY IMAGE SUPER-RESOLUTION

Wenbo Bao *Xiaoyun Zhang* [*] *Shangpeng Yan* *Zhiyong Gao*

Institute of Image Communication and Network Engineering,
Shanghai Jiao Tong University, China
{baowenbo,xiaoyun.zhang,ysp123,zhiyong.gao}@sjtu.edu.cn

ABSTRACT

Images captured by camera tend to be noisy and their qualities are often deteriorated in super-resolution. In this paper, we propose an end-to-end convolutional neural network to generate denoised, high-resolution image directly from its noisy, low-resolution counterpart. To preserve textures and eliminate noises simultaneously, the network is organized into an iterative structure for the recovery of high-quality image step by step. Each step of the structure is aimed to learn a better result with reference of its predecessor's output. Experiments show that our method is able to produce more desirable high-resolution images in both objective and subjective evaluations comparing to conventional ones as well as non-iterative network based one.

Index Terms— Convolutional Neural Network, iterative structure, super-resolution, denoising, image reconstruction

1. INTRODUCTION

Super-Resolution (SR) aims to enhance the spatial resolution of an image. It has been extensively investigated in the past decades and great achievements have been obtained in the literature [1–6]. However, in practical scenarios where images tend to be noisy [7], the application of SR is usually impeded [8]. This paper focuses on the super-resolution of image from its noisy one.

In the context of SR, many theories varying from interpolation based approaches [1] to sparse representation [2,4] or neighbor embedding [2,9] based ones have been proposed. Very recently, based on deep learning, the performance of SR is promoted to a new stage by incorporating Convolutional Neural Network (CNN). Representative algorithms are SRCNN [5] and VDSR [6]. Both in quality and speed, learning based methods have achieved great success over traditional ones such as A+ [2], ScSR [4], *etc*. However, the SR methods above are only effective for noiseless images. To extend the application of SR, some works are held for the combinational processing of denoising and super-resolution [7,8,10,11]. In [7,10,11], multiple consecutive frames of a video are used to synthesize a super-resolved one. For single image super-resolution, some researchers performed simultaneous denoising and super-resolution in spatial [12] or frequency domain [13,14]. Singh *et al*. [8] proposed a convex combination of orientation and frequency selective bands of noisy and denoised high-resolution images to obtain a desired one. Though this method is effective for the irregular textures such as hairy objects, but it pays little attention to the regular textures like edges.

Inspired by the achievement of SR, we take advantage of CNN in the problem of simultaneous noise removal and resolution enhancement. And the idea of residual learning is also adopted for deeper network and fast convergence. To further improve the performance, we organize the network into an iterative structure which is composed of multiple subnetworks. Specifically, the reconstruction is not finished in a single-shot network, but is iteratively refined by the way that each sub-network takes its predecessor's output as a reference to make better reconstruction. Conventionally, to accomplish the task of noisy image super-resolution, a naive scheme is to perform denoising, followed by super-resolution. Its drawback remains in that the image textures lost in denoising process can no longer be restored, or more severely, remaining noises are improperly amplified by super-resolution. However, by our iterative network structure, the lost details in previous sub-networks can be regenerated by latter ones and the remaining noises can be removed thoroughly. Extensive experiments validate that our method accomplishes both texture preservation and noise removal simultaneously.

The rest of this paper is organized as follows. Section 2 will introduce the proposed iterative convolutional neural network. Experimental results will be given in Section 3, and conclusions are made in Section 4.

2. PROPOSED METHOD

2.1. Convolutional Neural Network Model

In noisy image super-resolution, the source image has smaller size than the target. By our method, a preprocessing by bicu-

[*] Corresponding author

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

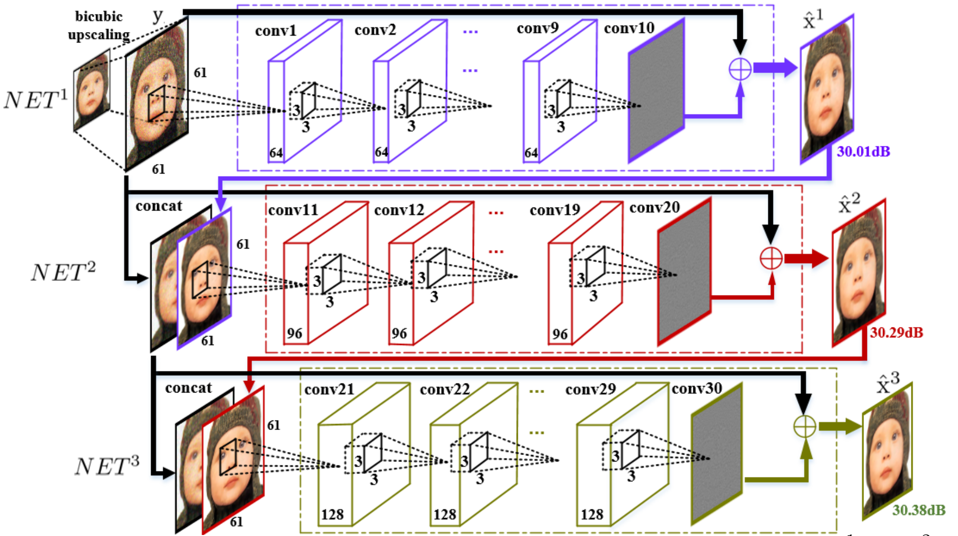

Fig. 1 Iterative CNN model, I-CNN-3. The **blue**, **red** and **green** color are used to distinguish between the modules of NET^1, NET^2 and NET^3 respectively.

bic interpolation is performed to guarantee that the source and target are of the equal size in order to use Convolutional Neural Network (CNN). We note the noisy image signal as y and its corresponding high-quality groundtruth as x. By learning based method, it is expected to transform y into $\hat{\mathrm{x}}$ which approximates x.

$$\hat{\mathrm{x}} = f(\mathrm{y}; \Theta) \tag{1}$$

Where Θ is the weight parameter set to be learned. It parameterizes a number of linear and non-linear functions to form a complex representation of $f(\cdot)$. In our CNN based method, we use convolution filters to perform linear transformation and the ReLU (Rectified Linear Unit) [15] for the non-linear part. CNN has been demonstrated to be useful for image applications since it exploits the spatially-local correlations in image and utilizes the weight sharing mechanism to reduce the amount of required weights. Besides, it is plausible to adopt the residual learning [16]. By our method, the network will try to learn the residual of the groundtruth and noisy image. The output of our network is to approximate the noise residual, which means that Eq.1 is reformulated into

$$\hat{\mathrm{x}} = f(\mathrm{y}; \Theta) + \mathrm{y} \tag{2}$$

2.2. Iterative Network Structure

In addition to the CNN model, we propose cascading multiple networks to obtain enhanced quality of output. Each of these networks is referred to as a *sub-network* (shorted as *subnet*) and is capable of generating a new output image when fed with original noisy data along with its previous subnet's result. Formally, a series of functions f^k is to be learned instead of a single-shot one as in Eq.2.

$$\hat{\mathrm{x}}^k = f^k(\mathrm{y}, \hat{\mathrm{x}}^{k-1}; \Theta^k) + \mathrm{y}, k = 2, 3, ..., K \tag{3}$$

The whole structure has K subnets and k indexes the step of a subnet, which is named as NET^k. Especially, at the step $k = 1$ which dose not have a predecessor, the mapping goes into the degraded form

$$\hat{\mathrm{x}}^1 = f^1(\mathrm{y}; \Theta^1) + \mathrm{y} \tag{4}$$

The weight parameter sets $\{\Theta^k\}_{k=1}^K$ make up of the full description of our model.

Fig. 1 illustrates an instance of this network model. Since it is composed of 3 subnets, we refer to it as I-CNN-3. It has a convolution kernel size of 3×3 and the number of filters for NET^1 to NET^3 are 64, 96 and 128 respectively. Each subnet has 10 *convolution* layers (**conv**$*$) and 9 *ReLU* activation layers (not shown in the figure) where the last convolution layer that produces image pixels has no non-linear module. For all convolution filters, padding is used to guarantee an equal size of input patch y and output patch x^k. For NET^2 and NET^3, there is an additional *concatenation* layer (**concat**) that combines the raw input and previous subnet output into a two-channel image. For the task of simultaneous denoising and super-resolution (DnSR), in addition to the I-CNN-3 model, we also provide a simplified one, namely the I-CNN-2, in which the third subnet NET^3 is omitted.

Here, one may doubt whether a single-shot network with the similar capacity is capable enough to achieve the same reconstruction quality. However, we think that our network structure has some advantages comparing to that single-shot one and can perform better. First, this whole network is not as deep as a single-shot network is. Because at each subnet, the output is constrained to approximate its groundtruth image. The depth of the entire network can be identified as nearly equivalent to that of its single subnet. Second, each subnet can take use of the reconstruction information by its predecessor. This is particularly necessary for the severally distorted case when image has low-resolution and is also noisy. Previous step information can implicitly provide the prior that whether the true signal have been preserved or not.

Besides, we have examined this idea in the task of super-

Table 1 The average PSNR (dB) and SSIM of different DnSR methods at noise level $\sigma = 20$ and scale ratio $r = 2$ on dataset Set5 and Set14 as well as 5 image results of them. **Red** color indicates the best performance and Blue color indicates the second best performance.

Results	BM3D [17] +Bicubic		NLM [18] +LSE [9]		BM3D [17]+ SRCNN [5]		MLP [19]+ SRCNN [5]		BM3D [17]+ VDSR [5]		VD_DnSR [6]		I-CNN-2		I-CNN-3	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5 avg.	28.32	0.805	25.42	0.712	28.61	0.807	28.50	0.795	28.58	0.806	28.80	0.809	29.02	0.820	**29.12**	**0.825**
baby	30.53	**0.827**	29.04	0.741	**30.55**	**0.827**	30.42	0.825	30.53	0.826	30.16	0.813	30.36	0.821	30.38	0.824
bird	29.35	0.834	26.26	0.737	29.45	0.832	29.25	0.822	29.43	0.831	29.36	0.825	29.71	0.841	**29.81**	**0.844**
butterfly	24.44	0.811	21.82	0.772	25.19	0.825	25.03	0.785	25.07	0.822	26.36	0.861	26.49	0.871	**26.68**	**0.881**
head	29.98	0.710	28.36	0.627	29.98	0.710	29.88	0.710	29.98	0.709	29.85	0.700	29.95	0.707	**30.08**	**0.713**
woman	27.64	0.841	21.59	0.682	27.89	0.841	27.94	0.834	27.88	0.840	28.27	0.845	28.57	0.862	**28.63**	**0.864**
Set14 avg.	26.65	0.715	24.88	0.655	27.04	0.725	26.81	0.717	27.00	0.724	26.99	0.722	27.24	0.731	**27.33**	**0.740**
lena	29.90	0.808	28.64	0.732	30.08	0.809	29.94	0.802	30.06	0.808	29.85	0.795	30.22	0.809	**30.24**	**0.811**
baboon	22.64	0.471	22.37	0.469	22.85	0.496	22.83	0.500	22.86	0.497	22.85	0.505	22.86	0.502	**22.98**	**0.523**
comic	22.85	0.660	22.58	0.665	23.24	0.682	23.21	0.675	23.20	0.682	23.57	0.702	23.65	0.709	**23.75**	**0.721**
pepper	30.33	0.816	28.72	0.740	30.50	0.814	30.37	0.808	30.48	0.814	30.41	0.810	30.89	0.826	**30.99**	**0.827**
ppt3	25.22	0.890	22.42	0.782	27.13	0.907	25.81	0.873	27.00	0.905	26.76	0.904	27.92	0.932	**27.99**	**0.936**

resolution of clean images by training a I-CNN-3 model for it. **Table 2** shows the average results on test set Set5 and Set14 by different SR algorithms including A+ [2], SRCNN [5], VDSR [6] and I-CNN-3. Although our method I-CNN-3 (30 convolution layer) has larger capacity than VDSR [6] (20 convolution layers), but it also has been shown in [6] that more depth for VDSR may degrade the performance. In the evaluation of PSNR, our method outperforms state-of-the-art SR algorithm by about 0.09dB and 0.08 dB on Set5 and Set14 respectively. And the comparison of SSIM also confirms the superior performance. This experiment has demonstrated the effectiveness of the iterative CNN model but we will show later that it performs much more better for DnSR application.

Table 2 The average PSNR (dB) and SSIM results of SR algorithms at a scale ratio of $r = 2$ on Set5 and Set14.

Results	A+ [2]		SRCNN [5]		VDSR [6]		I-CNN-3	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	36.54	0.954	36.66	0.954	37.53	0.958	**37.62**	**0.959**
Set14	32.28	0.905	32.42	0.906	33.03	0.912	**33.11**	**0.913**

2.3. Loss Function

Back to our network, there are some critical issues to be clarified. The first one is the loss function. At each step, we wish to get a reconstructed version of the image. To reach that purpose, we implement a combined loss function to impose constraint on the output of each step. Firstly, each subnet NET^k has its own loss l^k, which is the Euclidean distance with respect to ground-truth image.

$$l^k = ||\hat{\mathrm{x}}^k - \mathrm{x}||_2^2 \tag{5}$$

Intuited by that a further refinement should have higher pixel accuracy, we then assign different importances for these losses in the total loss function.

$$L = \sum_{k=1}^{K} \alpha^{K-k} l^k, 0 < \alpha < 1 \tag{6}$$

It encourages the latter subnets to provide better results comparing with their predecessors. Hyperparameter α is used to balance the contribution of each subnet.

2.4. Learning Rate

The second issue is how to effectively train the iterative neural network. One may suggest to pretrain the subnet NET^1 to a converged state and then train the left network sequentially. However, to make the training process simple, we introduce a learning rate decay parameter β for it.

$$lr^k = \beta^{k-1} lr_{basic}, 0 < \beta < 1 \tag{7}$$

Where lr_{basic} is the basic learning rate and is also NET^1's learning rate. lr^k represents the learning rate of NET^k. The equation means that one certain subnet can adjust itself more quickly than its successors. Comparing to the strategy of an equal learning rate for all subnets, our method is helpful to achieve fast convergence. And the network can automatically reach to the status that better results are supplied step by step.

3. EXPERIMENTAL RESULTS

Training. To ensure a fair comparison, our training set is generated from 291 images as also used by VDSR [6]. They contain 200 images from Berkeley Segmentation Data [20] and 91 images from Yang *et al.* [4]. Moreover, the image data are augmented by flip and rotation (×8 enrichment). For DnSR task, these raw data are firstly downsampled and then distorted by additive Gaussian noise at zero mean and variance of $\sigma = 20$. For SR task illustrated in **Table 2**, noise is not added. Then the original and distorted image constitute a pair of training sample. For the optimization, we use stochastic gradient descent algorithm. The basic learning rate lr_{basic} is 0.01. We train the networks over 50 epochs and after every 20 epochs, the learning rate is decreased by 0.1. In training, the batch size is set to 64 for I-CNN-3 and 128 for I-CNN-2.

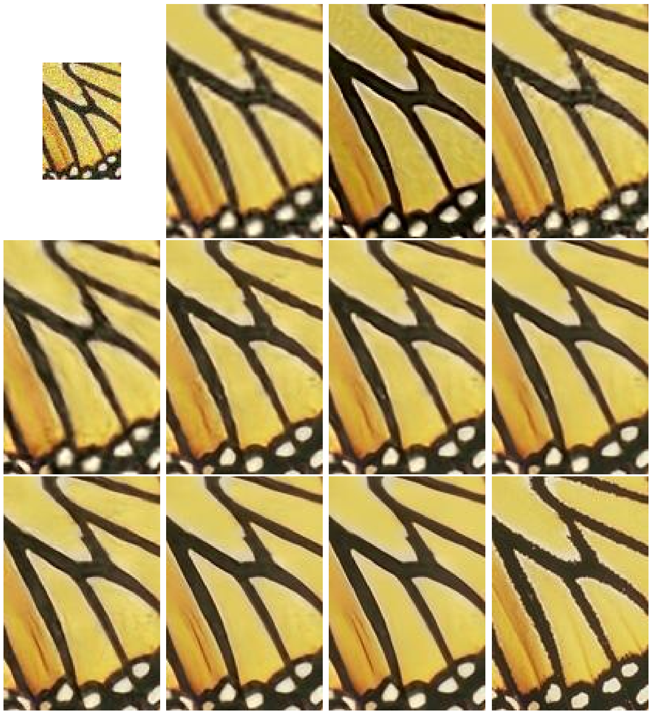

Fig. 2 The enlarged view of *butterfly* by different denoising and super-resolution methods at noise level $\sigma = 20$ and scale ratio $r = 2$. The first row are results of original data, BM3D [17]+Bicubic, NLM [18]+LSE [9] and BM3D [17]+SRCNN [5]. The second row are results of MLP [19]+SR-CNN [5], VD_DnSR [6], NET^1, NET^2 of I-CNN-2. The third row are results of NET^1 to NET^3 of I-CNN-3 and the groundtruth.

The sample image size for the two networks are 61×61 and 41×41 respectively. The momentum, weight decay and the gradient clipping are set to 0.9, 0.0001, and 5.0 respectively. As for the hyperparameters proposed in this paper, loss parameter α is set to 0.4, and learning rate decay parameter β is set to 0.5. The training is implemented using Caffe [21].

Test. We choose Set5, Set14 as the test sets to evaluate the performance of our networks. The PSNR and SSIM of the final output with respect to its original groundtruth are calculated. There are many algorithms involved in DnSR task. Firstly, the conventional cascading schemes of state-of-the-art denoising and super-resolution algorithms are used. There are five of them, including BM3D [17]+Bicubic, NLM [18]+ LSE [9], BM3D [17]+ SRCNN [5], MLP [19] + SRCNN [5], and BM3D [17] + VDSR [5]. Besides, to make a full comparison, we use the network in [6], which is originally for SR only, to train a DnSR network, and rename it as VD_DnSR. Finally, we provide two DnSR network models, *i.e*., the I-CNN-2 and I-CNN-3.

Results. The result of objective comparison is shown in **Table 1**. In this table, in addition to the average results on Set5 and Set14, we also pick 5 image results from each set for a detailed illustration. As one can observe, our algorithms perform better than conventional cascaded ones in most cases. On the two test sets, I-CNN-2 achieves 29.02dB and 27.24dB on average respectively, which are superior than 28.80dB by VD_DnSR [6] and 27.04dB by BM3D [17]+SRCNN [5]. And the I-CNN-3 further improve the performance at about 0.1dB advantage comparing to I-CNN-2. Furthermore, the subjective comparison of enlarged view of *butterfly* is depicted in **Fig. 2**. Conventional cascading methods exhibit blurry textures and remaining noises. This can be explained by that the denoising algorithm may blur the edges and SR can not enhance them effectively and denoising sometimes produces structural artifacts which are improperly enhanced by SR. This phenomenon can be also observed in VD_DnSR's result that the noise of flatten area is not removed completely. By contrast, our algorithms could produce better results. In the third row of **Fig. 2**, the first to third images are results of NET^1 to NET^3. They demonstrate that the noises are removed and edges are sharpened step by step.

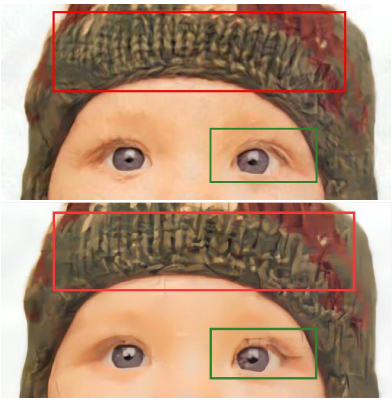

Fig. 3 The comparison between Singh's [8] (upper) and I-CNN-3 (bottom) on *baby*. Red box contains the irregular texture area while green box contains the regular texture area.

In addition to these comparisons above, we further discuss the DnSR method proposed by Singh *et al*. [8]. The results are depicted in **Fig. 3**. Their method is mainly effective for some of the irregular texture areas such as animals' fur and hairy object since denoising operation usually produces washing effect on them. In regular texture areas, they failed to keep the sharpness and suffers from blurry effect. However, our method is good at providing visually clean image with sharp edges.

4. CONCLUSION

In this paper, we incorporate convolutional neural network in the task of DnSR, and propose the iterative network structure to achieve superior performance. The subnet of iterative CNN is demonstrated to learn better results from their predecessor, which enables our method to win over others in objective and subjective evaluations.

Acknowledgment. This work was supported in part by National Natural Science Foundation of China (61301116, 61521062, 61133009, 61420106008), Chinese National Key S&T Special Program (2013ZX01033001-002-002), the 111 Project (B07022), the Shanghai Key Laboratory of Digital Media Processing and Transmissions(STCSM 12DZ2272600).

5. REFERENCES

[1] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[2] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision (ACCV)*. Springer, 2014, pp. 111–126.

[3] M. Türkan, D. Thoreau, and P. Guillotel, "Iterated neighbor-embeddings for image super-resolution," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 3887–3891.

[4] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 184–199.

[6] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] J. Suo, Y. Deng, L. Bian, and Q. Dai, "Joint non-gaussian denoising and superresolving of raw high frame rate videos," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1154–1168, 2014.

[8] A. Singh, F. Porikli, and N. Ahuja, "Super-resolving noisy images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2846–2853.

[9] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 2, p. 12, 2011.

[10] M. Elad and A. Feuer, "Restoration of a Single Super-resolution Image from Several Blurred, Noisy, and Undersampled Measured Images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[11] F. Liu, J. Wang, M. Gleicher, and Y. Gong, "Noisy video super-resolution," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 713–716.

[12] F. Dekeyser, P. Bouthemy, P. Perez, and E. Payot, "Super-resolution from noisy image sequences exploiting a 2d parametric motion model," in *in Proceeding of IEEE Conference on Pattern Recognition (ICPR)*. IEEE, 2000, pp. 350–353.

[13] F. Qiu, Y. Xu, C. Wang, and Y. Yang, "Noisy image super-resolution with sparse mixing estimators," in *4th International Congress on Image and Signal Processing*. IEEE, 2011, pp. 1081–1085.

[14] M. B. Chappalli and N. K. Bose, "Simultaneous noise filtering and super-resolution with second-generation waveletss," *Signal Processing Letters*, vol. 12, no. 11, pp. 772–775, 2005.

[15] V. Nair and G. 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.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[18] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 60–65.

[19] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2392–2399.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2001, pp. 416–423.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.