Densely Connected Hierarchical Network for Image Denoising

Bumjun Park[1], Songhyun Yu[1], Jechang Jeong[2*]
[1]Department of Electronics and Computer Engineering, Hanyang University, Seoul, Republic of Korea
[2]Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea
kkbbbj@gmail.com, fkdlzmftld@gmail.com, jjeong@hanyang.ac.kr

Abstract

Recently, deep convolutional neural networks have been applied in numerous image processing researches and have exhibited drastically improved performances. In this study, we introduce a densely connected hierarchical image denoising network (DHDN), which exceeds the performances of state-of-the-art image denoising solutions. Our proposed network improves the image denoising performance by applying the hierarchical architecture of the modified U-Net; this enables our network to use a larger number of parameters than other methods. In addition, we induce feature reuse and solve the vanishing-gradient problem by applying dense connectivity and residual learning to our convolution blocks and network. Finally, we successfully apply the model ensemble and self-ensemble methods; this enables us to improve the performance of the proposed network. The performance of the proposed network is validated by winning the second place in the NTRIE 2019 real image denoising challenge sRGB track and the third place in the raw-RGB track. Additional experimental results on additive white Gaussian noise removal also establish that the proposed network outperforms conventional methods; this is notwithstanding the fact that the proposed network handles a wide range of noise levels with a single set of trained parameters.

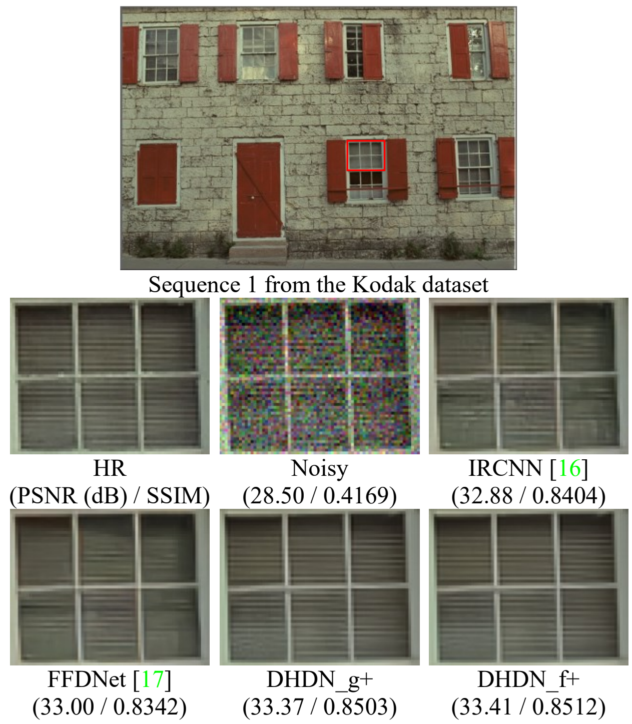


Figure 1: Denoising results of the conventional methods and the proposed method on noise level σ = 30.

1. Introduction

Image denoising is a process that generates a high quality image from a low quality image which is degraded by external noises such as additive white Gaussian noise (AWGN) [1, 15], speckle noise [3, 8], and impulse noise [7]. Image denoising is a major research area in image processing research field because of its wide range of use such as medical image denoising [5, 6, 28], satellite image denoising [2, 8], and compression noise denoising [9, 10]. Among many uses, object detection [11, 12] and recognition [26, 27, 28] in autonomous vehicles significantly increased attention of the researchers on image denoising; this is because it is essential to remove noise from an image to improve the performance of object recognition. Owing to these demands on image denoising research, numerous image denoising solutions have been proposed [3, 4, 25]. However, there was limited improvement in image denoising performance prior to the application of deep convolutional neural network (CNN). BM3D [13], which was proposed in 2007 by Dabov et al., was the most popular image denoising algorithm prior to the application of CNN. This reveals that image denoising research lacked progress in terms of performance improvement. Recently, the performance of numerous image processing solutions, including image denoising, improved substantially with the application of CNN [20, 22, 23, 24].

In 2016, Zhang et al. proposed deep CNN for image denoising (DnCNN) [15]; it is popular as an early stage CNN model for image denoising. They apply batch normalization (BN) [21] and residual learning [27] on their model and demonstrate improved performance. As an early stage model, DnCNN exhibits the scope for improvement in denoising performance with the application of CNN. In the following year, Zhang et al. proposed CNN denoiser for image restoration (IRCNN) [16]; it reduces the computational complexity with minimizing the performance degradation. They apply a dilated filter [37] to their seven-layer model to enlarge the receptive field [22]. As a result, IRCNN exhibits competent performance with considerable reduction in the computational complexity. Zhang et al. also proposed fast and flexible denoising network (FFDNet) [17]; it can flexibly handle a wide range of noise levels with a single image denoising network. They indicate that conventional image denoising models learn to remove noise from images with a specific noise level, which compels them to train multiple models for different noise levels [3, 15, 25]. To solve this problem, FFDNet trains their model using training images with different noise levels to handle multiple noise levels. However, FFDNet exhibits their limitation when removing noise of images with unspecified noise level because FFDNet requires the noise level of the image as an input data. Another approach to solving the limitation of conventional denoising networks wherein they require a model for each noise level was proposed by Lefkimmiatis. He proposed universal denoising network (UDN) [18], which halves the range of noise level and covers half range of the noise level with a single model. However, UDN also requires the noise level of images as an input data. Liu et al. proposed multi-level wavelet CNN (MWCNN) [19], which applies modified U-Net [28] architecture and wavelet transform [2] to their model. Whereas they enlarge the receptive field of their model while reducing its computational complexity, the limitation that their model is effective only for single noise level remained unsolved. Moreover, their application of wavelet transform can result in performance degradation by compelling their network to use feature information of the wavelet transform.

In this paper, we propose a CNN-based denoising solution that overcomes the limitations of the conventional methods and exceeds the performance of state-of-the-art denoising solutions. The proposed network applies the hierarchical structure of the modified U-Net, which enables our network to efficiently use limited memory. By reducing the memory used for storing information of feature maps, the proposed network can use a larger number of parameters than the conventional methods; thereby, our network exhibits better results than those of conventional methods. As our network has a larger number of parameters than conventional methods, it can suffer from the vanishing-gradient problem [40]. We apply dense connectivity [26] and residual learning [27] to our convolution block and network and successfully resolve the problem. Moreover, we train our model with noisy input images with a wide range of noise levels to enable our model to handle multiple noise levels with a single set of trained parameters. Most conventional denoising models exhibit a limitation wherein they can handle only one noise level with a single trained model. Notwithstanding the contribution of FFDNet and UDN toward overcoming this limitation, they are constrained by their need for the noise level of the input image as an input data. To solve this problem completely, we train our model to handle a wide range of noise levels without any input data of input noise level. Although our network handles multiple noise levels with a single set of trained parameters, the proposed network outperforms conventional solutions trained for a specific noise level. The proposed network exhibits further improvement in performance when it is trained for a specific noise level. Finally, we apply the self-ensemble [34] and model ensemble [35, 36] methods, enabling our network to improve the quality of output images.

Our main contributions are summarized as follows:

- We apply the hierarchical architecture of the modified U-Net, enabling our network to efficiently use limited memory. Thereby, our model can use a larger number of parameters than conventional networks.
- We apply dense connectivity and residual learning to our novel convolution blocks and network architecture to remove the noise of input images accurately and solve the vanishing-gradient problem.
- We apply the self-ensemble and model ensemble methods; this enables our proposed network to improve the objective and subjective quality of output images.
- We train our model to handle a wide range of the noise levels with a single set of trained parameters. As our network does not require information of input noise level, we completely overcome the limitation of conventional methods.

2. Related works

2.1. Hierarchical structure

As image processing researches are starting to apply CNN, it is important to use the limited memory efficiently to deepen networks. One of the solutions of conventional image processing algorithms is hierarchical architecture [38, 39]. Hierarchical structures have been used for numerous image processing researches to reduce the computational complexity and memory consumption of algorithms. For CNN models, Ronneberger et al. proposed U-Net [28]; it applies the concept of a hierarchical structure to the CNN model. A U-Net consists of two paths: contracting path and expanding path. In the contracting path, the U-Net halves the size of the feature map with a 2×2 max pooling operation with stride 2 while increasing the number of feature maps to two times. As a result, each downsampling step halves the amount of data the U-Net should handle. It enables the U-Net to use a larger number of parameters than

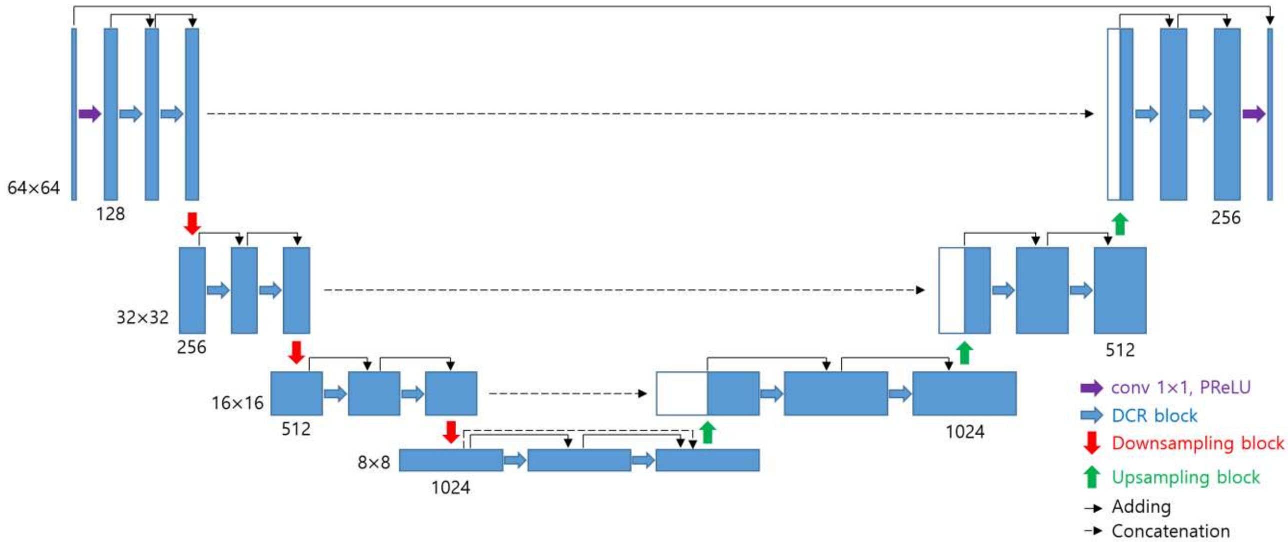


Figure 2: Architecture of proposed DHDN. The size of the input image is set to 64 × 64 as an example.

other models. Inspired by the U-Net, our proposed network applies the hierarchical structure with a modification.

2.2. Dense connectivity and residual learning

As CNN models deepened, another problem exhibited by numerous models is the vanishing-gradient problem [40], which is a critical issue because it completely hinders the models from training the parameters as the models deepened. To solve this problem, He et al. proposed deep residual learning for image recognition (ResNet) [27], and Huang et al. proposed densely connected convolutional network (DenseNet) [26]. ResNet solves the vanishing-gradient problem by applying skip connection, which enables a network to learn residual functions. Similar to ResNet, DenseNet solves this problem by connecting layers; however, they connect each layer to all the other layers in a feed-forward fashion to induce the network to reuse the information of previous feature maps. Combining the two methods, Zhang et al. proposed residual dense network for image restoration (RDN) [1]. Inspired by these methods, we organize our block and network structure using residual learning and dense connectivity.

2.3. Self-ensemble and model ensemble methods

The ensemble method is a technique that yields better output by combining more than one output. From among numerous methods, we apply the self-ensemble [20, 34] and model ensemble [35, 36] methods. In the self-ensemble method, the outputs of the transformed input images are averaged. It is a highly efficient ensemble method because it does not require any additional training process. In this study, we average eight output images of eight input images; these are generated by a combination of a flip and rotation of an input image. In the model ensemble method, the outputs of more than two separate networks are averaged. Unlike the self-ensemble method, it is necessary to train more than two networks to apply this method. In this study, we train two same models with training conditions that are identical except for the initialization of the parameters to apply the model ensemble method.

3. Proposed network architecture

Figure 2 shows the architecture of the proposed densely connected hierarchical image denoising network (DHDN). As mentioned above, the proposed network applies the hierarchical architecture of the modified U-Net [28]. As the input image comes in, the proposed network first executes a 1 × 1 convolution operation followed by a parametric rectified linear unit (PReLU) [41] to generate feature maps for our proposed densely connected residual block (DCR block). This initial convolution layer enables us to apply local residual learning in the DCR block. More importantly, we can use both grayscale and color images as the input of the proposed network without modifications because of the initial convolution layer. As mentioned in Figure 2, the initial convolution layer generates 128 feature maps for the DCR block. Then, there exist two DCR blocks in each level of our proposed network. The architecture of our proposed DCR block is discussed in Section 3.1. Next, the output feature maps of the two DCR blocks are downsampled by a factor of two by the downsampling block. When downsampling the feature maps, we double the number of output feature maps to prevent a severe decrease in the amount of information. The architecture of the downsampling block is discussed in Section 3.1. Our proposed network follows the above procedure three times along the contracting path; this causes our network to consider four resolution levels of feature maps. Then, our proposed network follows the expanding path, which is the

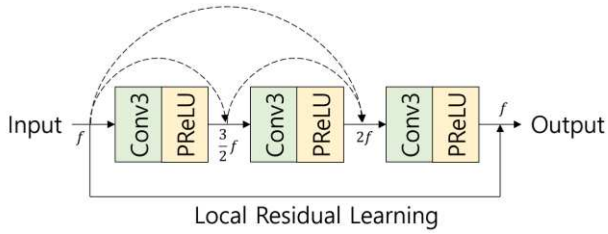
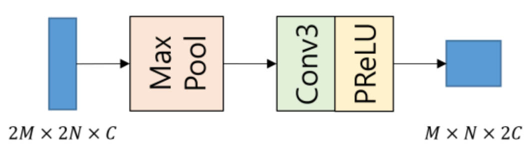


Figure 3: Architecture of DCR block.

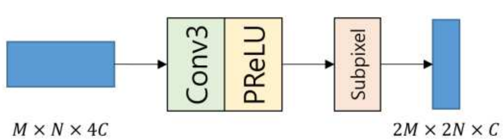
inverse process of the contracting path. After the operations of the two DCR blocks, the output feature maps of each level are upsampled by a factor of two by the upsampling block. When upsampling the feature maps, the number of feature maps is reduced to one-fourth because we apply the sub-pixel interpolation method [24] to our upsampling block. To prevent a severe decrease in the number of feature maps and induce our proposed network to use the information of the previous feature maps, the output of the upsampling block is connected to the input of the downsampling block which is located on the same level with the upsampling block by dense connectivity [26]. However, for the lowest level, we connect the input of the upsampling block to the output of the downsampling block. The architecture of the upsampling block is discussed in Section 3.1. Similarly, as in the contracting path, our network follows the above procedure three times along the expanding path. After the expansion procedure, the proposed network computes the final 1×1 convolution followed by PReLU [41] to generate the final output. The number of feature maps is set to one when the input is a grayscale image and three when it is a color image. Finally, we apply global residual learning [27] to our proposed network for generating output images by applying learned residual information to the input images.

3.1. Proposed block architectures

The proposed network architecture consists of three types of blocks: DCR block, downsampling block, and upsampling block. Figure 3 shows a architecture of DCR block; here, conv3 denotes a 3×3 convolution layer, and



(a)



(b)

Figure 4: The architecture of the downsampling block and upsampling block: (a) Downsampling block and (b) Upsampling block.

f denotes the number of feature maps. The DCR block consists of three convolution layers followed by PReLU. Each feature map is connected by dense connectivity to induce our model to use the information of previous feature maps. The growth rate [26] of DCR block was set to half of f; Moreover, the final convolution layer generates f feature maps as an output so that the DCR block can apply local residual learning. By applying dense connectivity and local residual learning, we can improve the information flow so that our proposed network can circumvent the vanishing-gradient problem [40] through an accurate removal of the noise.

Figure 4 (a) shows the architecture of the downsampling block. The downsampling block consists of two layers: a 2×2 max pooling layer, and a 3×3 convolution layer followed by PReLU. As the feature maps enter as the input, a 2×2 max pooling operation with stride 2 decreases the size of the feature maps. Then, the 3×3 convolution layer doubles the number of feature maps to prevent severe decrease in the amount of information. Thus, the output feature maps of the downsampling block are of one-fourth the size of the input feature maps, with two times as many feature maps.

Figure 4 (b) shows the architecture of the upsampling block. The upsampling block consists of two layers: a 3×3 convolution layer with PReLU and a sub-pixel interpolation layer [24]. Unlike U-Net [28], which uses a 2×2 deconvolution layer, the proposed upsampling block uses a sub-pixel interpolation layer [24] to expand the size of the feature maps more efficiently and accurately. Before the sub-pixel interpolation layer expands the size of the feature maps, the 3×3 convolution layer refines the feature maps to enable the sub-pixel interpolation layer to interpolate the feature maps accurately. Thus, the output feature maps of the upsampling block are two times larger in size than the input feature maps, with one-fourth the number of channels of the input feature maps.

3.2. Multiple noise level denoising

Conventional CNN-based denoising solutions exhibit a critical limitation wherein they are required to train a model for each noise level [15, 16, 25]. Although there were attempts to overcome this limitation through certain methods [17, 18], they cannot completely circumvent the external noise level information. To solve this problem, we train our model with training data that has random noise level so that our model can handle a wide range of noise levels without external input information. As our proposed network has an adequate amount of learnable parameters that can remove the noise accurately regardless of the noise level, our proposed network can handle a wide range of noise levels without external information of the noise level. The experimental results demonstrate that our proposed model can handle a wide range of noise levels with better

Method	Kodak [30]						BSD68 [31]					
	σ = 10		σ = 30		σ = 50		σ = 10		σ = 30		σ = 50	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Noisy	28.24	0.6607	18.93	0.2755	14.87	0.1557	28.30	0.7128	19.03	0.3380	15.00	0.2007
CBM3D [14]	36.57	0.9432	30.89	0.8459	28.62	0.7772	35.89	0.9512	29.71	0.8426	27.36	0.7632
DnCNN [15]	36.58	0.9446	31.28	0.8579	28.94	0.7915	36.12	0.9536	30.32	0.8611	27.92	0.7882
IRCNN [16]	36.70	0.9448	31.24	0.8581	28.92	0.7939	36.06	0.9533	30.22	0.8607	27.86	0.7889
FFDNet [17]	36.80	0.9462	31.39	0.8596	29.10	0.7949	36.14	0.9540	30.31	0.8603	27.96	0.7881
DHDN_g	37.30	0.9509	31.98	0.8743	29.72	0.8170	36.05	0.9532	30.12	0.8579	27.71	0.7874
DHDN_g+	37.31	0.9510	31.99	0.8744	29.73	0.8170	36.27	0.9556	30.41	0.8654	28.02	0.7965
DHDN_f	37.33	0.9508	31.95	0.8736	29.67	0.8160	36.45	0.9572	30.41	0.8639	28.02	0.7961
DHDN_f+	37.37	0.9511	32.01	0.8744	29.74	0.8175	36.48	0.9574	30.54	0.8671	28.01	0.7950

Table 1: Average PSNR (dB) and SSIM results of conventional methods and proposed method for color images. The best result is highlighted with red and the second best result is highlighted with blue.

Method	Kodak [30]						BSD68 [31]					
	σ = 10		σ = 30		σ = 50		σ = 10		σ = 30		σ = 50	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Noisy	28.22	0.6573	18.87	0.2729	14.78	0.1998	28.26	0.7094	18.97	0.3348	14.92	0.1984
BM3D [13]	34.39	0.9127	29.12	0.7877	26.98	0.7140	33.32	0.9158	27.75	0.7731	25.60	0.6858
DnCNN [15]	34.90	0.9223	29.62	0.8071	27.49	0.7368	33.88	0.9270	28.36	0.7999	26.23	0.7189
IRCNN [16]	34.76	0.9215	29.52	0.8056	27.45	0.7342	33.74	0.9262	28.26	0.7989	26.19	0.7171
FFDNet [17]	34.81	0.9226	29.69	0.8123	27.62	0.7437	33.76	0.9266	28.39	0.8032	26.29	0.7245
DHDN_g	34.43	0.9153	29.93	0.8211	27.88	0.7528	33.42	0.9213	28.55	0.8110	26.44	0.7296
DHDN_g+	34.54	0.9174	30.00	0.8237	27.93	0.7546	33.50	0.9230	28.59	0.8120	26.47	0.7308
DHDN_f	35.22	0.9278	30.06	0.8239	27.95	0.7579	34.02	0.9301	28.54	0.8103	26.38	0.7310
DHDN_f+	35.24	0.9281	30.11	0.8250	28.01	0.7591	34.04	0.9303	28.58	0.8116	26.43	0.7324

Table 2: Average PSNR (dB) and SSIM results of conventional methods and proposed method for grayscale images. The best result is highlighted with red and the second best result is highlighted with blue.

performance than the conventional methods. To illustrate the superiority of our proposed network, we also train our proposed network with a fixed noise level and demonstrate further improved results.

4. Experiments

4.1. Training details

There are numerous training sets for CNN-based image processing methods. Recently, Timofte et al. released the DIV2K dataset [29] for image restoration. The DIV2K training dataset consists of 800 high quality images. The resolution of each of these images is similar to the FHD resolution (1920 × 1080). The DIV2K validation dataset consists of 100 images; the quality of each image is similar to that of the training dataset. As DIV2K training and validation datasets provide an adequate amount of high quality images, numerous state-of-the-art image processing solutions use the DIV2K dataset for their network [20, 23]. For a similar reason, we use the DIV2K training and validation datasets for our proposed network. When training our model, we extract patches from the training images; the width and height of each patch is set to 64 pixels. For the global noise level model, which is trained to handle a wide range of noise levels, we randomly add AWGN to our training patches with the noise level ranging from 5 to 50. For the fixed noise level model, we train our model with noise levels of 10, 30, and 50. The input patches of the proposed network are randomly flipped and rotated for data augmentation, and the batch size of the training patches is set to 16. We use the Adam optimizer [42] with an initial learning rate of 1e-4. We halve the learning rate for every three epochs and we use L1 loss for the loss function [33]. For the test datasets, we use the Kodak dataset [30] and BSD68 dataset [31]; these are used by numerous state-of-the-art denoising networks [18, 19]. The Kodak dataset consists of 24 images, each of which has a resolution of 768×512. The BSD68 dataset consists of 68 images, each of which has a resolution of 321 × 481.

4.2. Performance comparison

We compare our proposed network with BM3D [13, 14], DnCNN [15], IRCNN [16], and FFDNet [17], which are state-of-the-art image denoising solutions. To compare the objective performance, we determined the peak-signal-to-noise-ratio (PSNR) [44] and the structural similarity (SSIM) [43] of the result images. Table 1 lists the average PSNR and SSIM results of the conventional methods and the proposed method for color images. In Table 1, DHDN_g denotes the proposed model trained for the global noise level; moreover, DHDN_g+ denotes the result of applying the self-ensemble method to the DHDN_g model. Similarly,

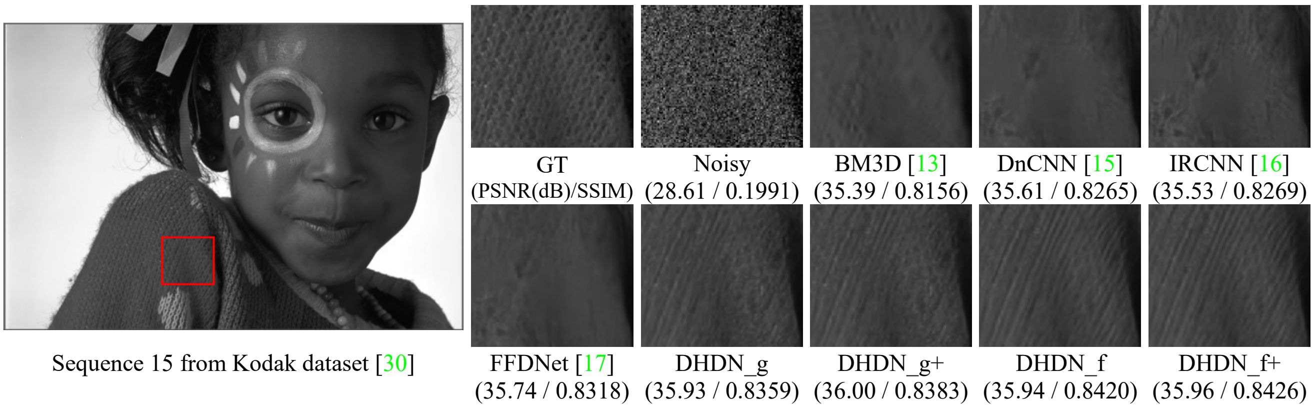


Figure 5: Denoising results of conventional methods and proposed method on noise level σ = 30.

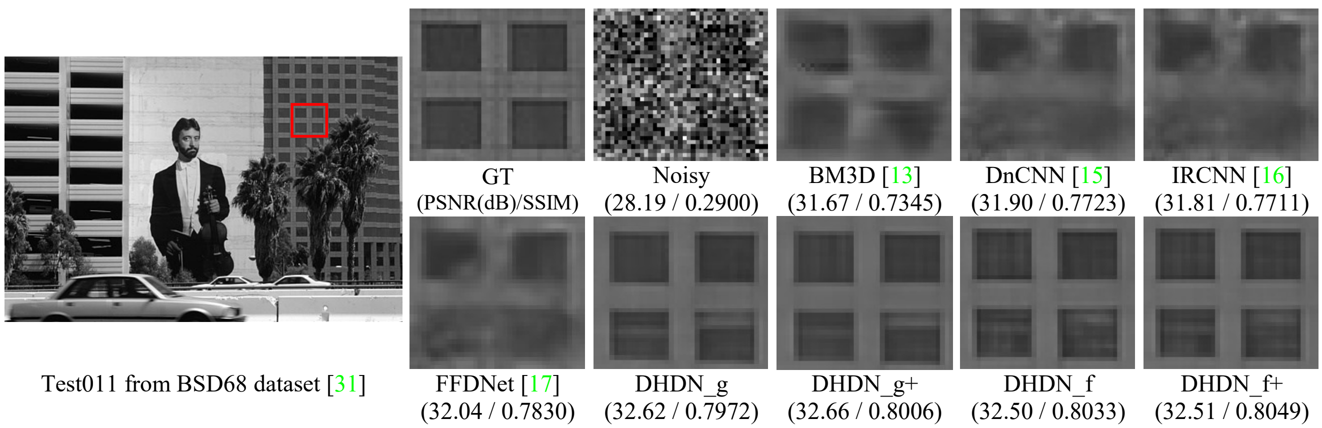


Figure 6: Denoising results of conventional methods and proposed method on noise level σ = 50.

DHDN_f denotes the proposed model trained for the fixed noise level, and DHDN_f+ denotes the result of applying the self-ensemble method to the DHDN_f model. In Table 1 and 2, the best result is highlighted with red, and the second best result is highlighted with blue. Note that we apply only the self-ensemble method [34] in the AWGN denoising experiment. The model ensemble method [35, 36] is applied only for the NTIRE 2019 denoising challenge [45] models. As illustrated in Table 1, the proposed network outperforms the conventional methods for all the conditions. For the Kodak [30] sequence, the proposed network outperforms the conventional methods by up to 0.74 dB for PSNR and 0.0078 for SSIM, when the noise level of the input image is 10. The proposed network outperforms the conventional methods by up to 1.1 dB for PSNR and 0.0285 for SSIM when the noise level is 30, and 1.11 dB for PSNR and 0.1398 for SSIM when the noise level is 50. The proposed network exhibits further improvement in performance when it is trained for the fixed noise level. The proposed network trained with the fixed noise level outperforms the proposed network with the global noise level by up to 0.06dB for PSNR. For the BSD68, [31] the proposed network outperforms the conventional methods by up to 0.38 dB for PSNR and 0.0044 for SSIM when the noise level of the input image is 10. The proposed network outperforms the conventional methods by up to 0.7 dB for PSNR and 0.0228 for SSIM when the noise level is 30, and 0.66 dB for PSNR and 0.0333 for SSIM when the noise level is 50. The proposed network exhibits further improvement in performance when it is trained for the fixed noise level. However, when the noise level is 50, the proposed network trained for the fixed noise level exhibits performance similar to that of the global noise level model. Moreover, the performance deteriorates when the self-ensemble method is applied. There can be two reasons for this phenomenon. One is that the performance is completely saturated for the proposed network so that the ensemble method exhibits similar or lower performance. The other reason is that training the network with multiple noise levels enhanced the performance for the high noise levels. In [22], the authors demonstrate that training the model with multiple scales can enhance the performance for large scales. Similarly, training the model with multiple noise levels enhances the performance for high noise levels; moreover, the experimental results reveals higher performance improvement for the high noise levels than for the low noise level when comparing the global noise level model with the fixed noise level model.

Table 2 presents the average PSNR and SSIM results of the conventional methods and of the proposed method for

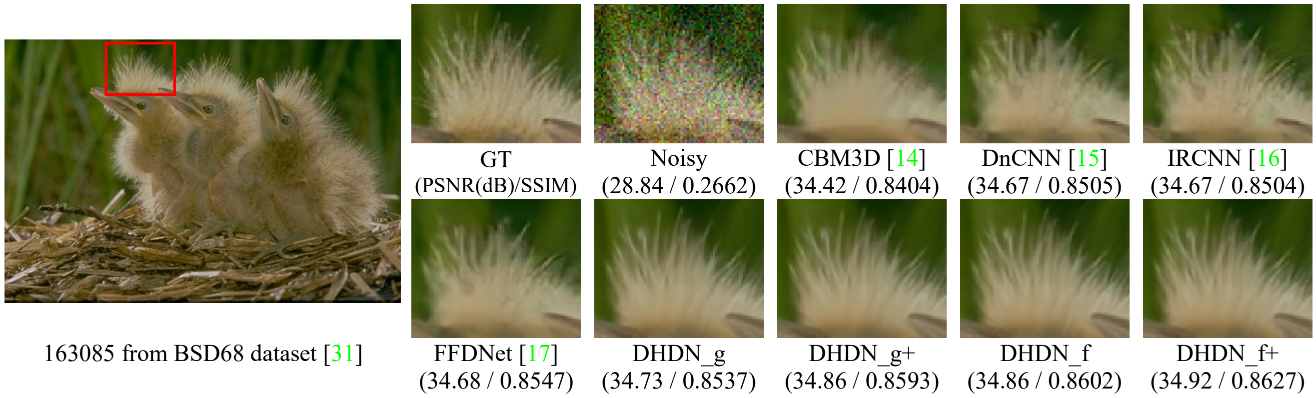


Figure 7: Denoising results of the conventional methods and the proposed method on noise level σ = 30.

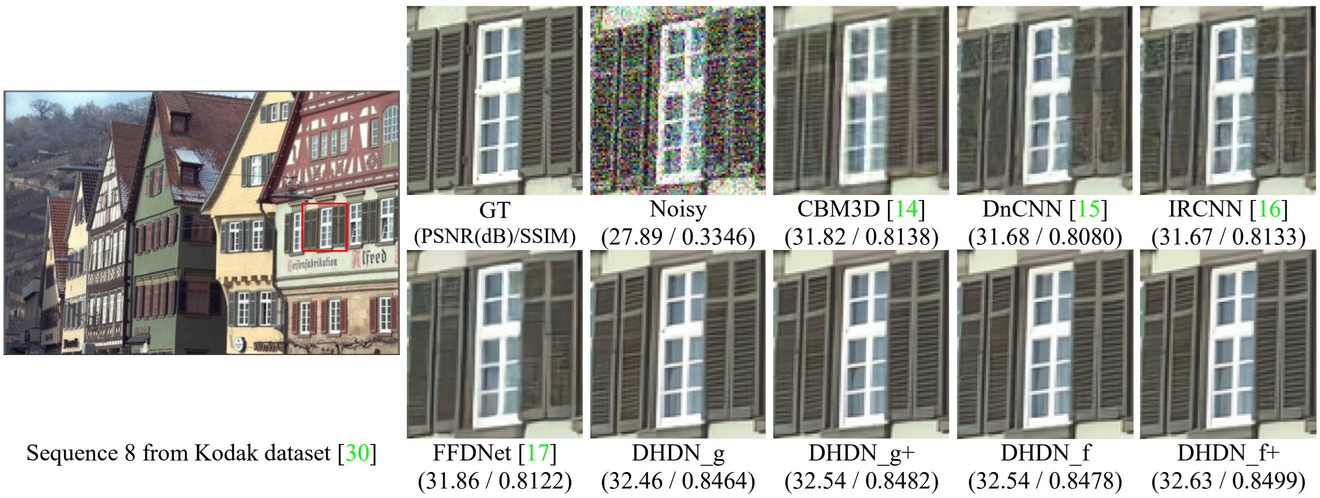


Figure 8: Denoising results of conventional methods and proposed method on noise level σ = 50.

grayscale images. As illustrated in Table 2, the proposed network outperforms the conventional methods except for the case wherein the noise level is 10. The proposed network exhibited up to 0.38 dB lower PSNR and 0.0049 lower SSIM when the noise level is 10. However, when the noise level is 30, the proposed network outperforms the conventional methods by up to 0.88 dB for PSNR and 0.036 for SSIM for the Kodak dataset, and 0.84dB for PSNR and 0.0389 for SSIM for the BSD68 dataset. When the noise level is 50, the proposed network outperforms the conventional methods by up to 0.95 dB for PSNR and 0.0406 for SSIM for the Kodak dataset, and 0.87 dB for PSNR and 0.045 for SSIM for the BSD68 dataset. It is additional evidence that training the network with multiple noise levels enhances the performance for the high noise levels [22]. The effect of the enhancement is substantial enough for the global noise level model to outperform the fixed noise level model in certain cases when the noise level is high. In addition, FFDNet [17], which is trained with multiple noise levels, exhibits a similar phenomenon. FFDNet exhibits lower performance than the fixed noise level models such as DnCNN [15] and IRCNN [16] when the noise level is low. However, as the noise level increased, FFDNet exceeds the performance of the fixed noise level models. It also illustrates that training the model with multiple noise levels enhances the performance for the high noise levels.

To compare the subjective performance, we compare the result images of each method. Figure 5 and 6 show the grayscale result images of the conventional methods and the proposed method. As shown in Figure 5, the conventional methods cannot restore the details of the clothe pattern, whereas the proposed method restores the patterns. In Figure 6, the conventional methods fail to restore the windows of the sequence. However, the proposed method restores the windows accurately for both the global noise level model and fixed noise level model. Figure 7 and 8 show the color result images of the conventional methods and proposed method. As shown in Figure 7, the proposed method recovers the pattern accurately, whereas the conventional methods yield blurred results. Similarly, unlike the conventional methods, which are not able to restore the details of the window shutter, the proposed method restores the details irrespective of whether it is trained with multiple noise levels or the fixed noise level in Figure 8.

	sRGB track				Raw-RGB track			
Rank	Team	Model	PSNR	SSIM	Team	Model	PSNR	SSIM
1	1st	1st	39.932	0.9736	1st	1st	52.114	0.9969
2	**Eraser**	**DHDN**	**39.883**	**0.9731**	Eraser	DIDN	52.107	0.9969
3	Eraser	DIDN	39.818	0.9730	**Eraser**	**DHDN**	**52.092**	**0.9968**
4	4th	4th	39.675	0.9726	4th	4th	51.947	0.9967
5	5th	5th	39.611	0.9726	5th	5th	51.939	0.9967

Table 3: Result PSNR (dB) and SSIM of NTIRE 2019 real image denoising challenge.

The proposed network ourperforms the conventional methods owing to a large number of the parameters, and it was possible by applying the hierarchical structure of the modified U-Net [28], which enabled our proposed network to use limited memory efficiently. For example, DnCNN, IRCNN, FFDNet, and DHDN have 558K, 188K, 851K, and 168M of parameters, respectively. Notwithstanding the number of parameters, comparison of multiply-accumulate operation (Mac) exhibits that our proposed network has a competitive computational complexity. While RDN [1] has 90.13G Macs with 22M parameters, our proposed network has 63.75G Macs with 168M parameters.

4.3. NTIRE 2019 real image denoising challenge

The proposed method is initially proposed to participate in NTIRE 2019 real image denoising challenge [45]. The purpose of the challenge is to remove unspecified noise from images. NTIRE 2019 real image denoising challenge consists of two tracks: raw-RGB track and sRGB track. As a team named Eraser, we submitted two denoising networks for each track: DHDN and deep iterative down-up network (DIDN), respectively. The results of the challenge establish the superiority of our proposed networks for removing unspecified noise from images. Table 3 illustrates the performance of the proposed models and the models of the other participants. As these tables illustrate, DHDN took the second place in the sRGB track and third place in the raw-RGB track. DHDN was trained with smartphone image denoising dataset (SIDD) [32] while participating in the challenge. SIDD consists of 320 images, each of whose resolution is similar to the UHD resolution (3840 × 2160). All the training condition was identical to that of the AWGN experiment except for the fact that we applied the model ensemble method [35, 36] while participating in the challenge. Figure 9 shows the challenge-result images of the proposed method on the validation dataset of SIDD. In Figure 9, DHDN++ denotes the result of the proposed method with the application of the model ensemble method. As shown in Figure 9, our proposed network successfully removes unspecified noise from noisy images.

5. Conclusion

In this study, we proposed a denoising network with a hierarchical structure. By applying the hierarchical structure of the modified U-Net, our proposed network efficiently used limited memory by decreasing the size of the feature maps. When upsampling the feature maps, we applied the sub-pixel interpolation method, enabling our model to interpolate the feature maps accurately and efficiently. Moreover, our proposed DCR block successfully removed the noise from the images and solved the vanishing-gradient problem by applying dense connectivity and residual learning. Finally, we applied the self-ensemble method and model ensemble method to improve the performance of the proposed network. As a result, our proposed network attained high ranks in NTIRE 2019 real image denoising challenge, establishing its superiority for removing unspecified noise from images. Additional experiments on AWGN demonstrated that our proposed network outperforms the conventional methods, handling a wide range of noise levels with a single set of trained parameters. The proposed network exhibited further higher performance when it was trained for the fixed noise level.

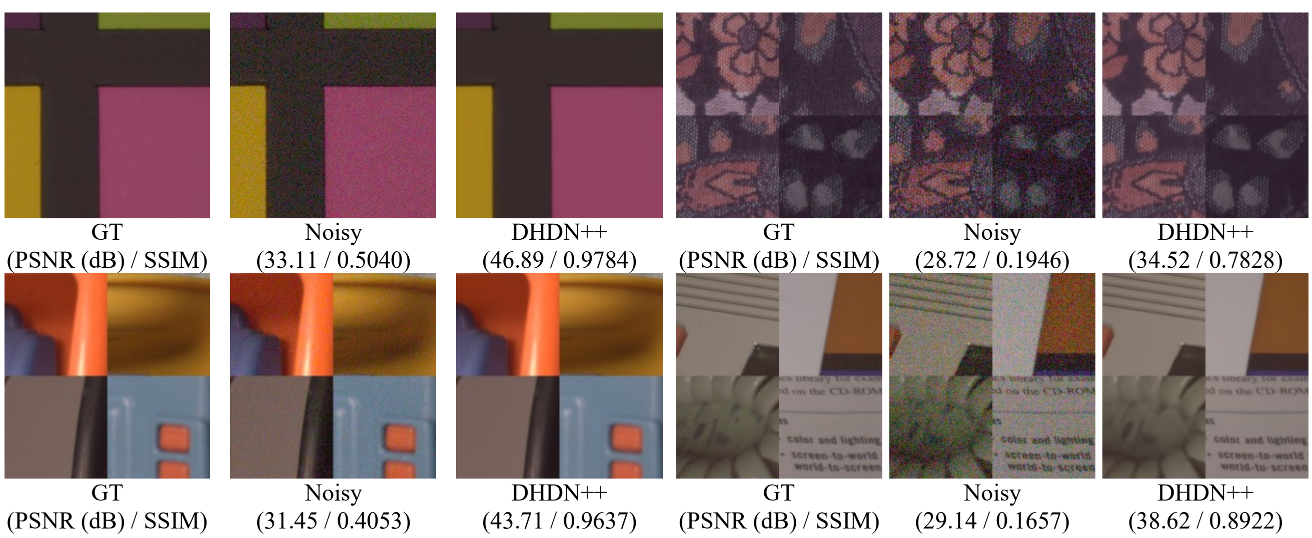


Figure 9: NTIRE 2019 real image denoising challenge results of proposed method on validation dataset of SIDD with unspecified noise.

References

[1] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image restoration. *arXiv preprint arXiv:1812.10477*, 2018.

[2] M. Mastriani and A. Giraldez. Microarrays denoising via smoothing of coefficients in wavelet domain. *arXiv preprint arXiv:1807.11571,* 2018.

[3] M. Mafi, S. Tabarestani, M. Cabrerizo, A. Barreto, and M. Adjouadi. Denoising of ultrasound images affected by combined speckle and Gaussian noise. *IET Image Processing,* 12(12):2346–2351, 2018.

[4] T. Tier and R. Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2019.

[5] W. Jifara, F. Jiang, S. Rho, M. Cheng, and S. Liu. Medical image denoising using convolutional neural network: a residual learning approach. *The Journal of Supercomputing*, 75(2):704–718, 2019.

[6] S. Li, H. Yin, and L. Fang. Group-sparse representation with dictionary learning for medical image denoising and fusion. *IEEE Transactions on Biomedical Engineering*, 59(12):3450–3459, 2012.

[7] Y. Dong and S. Xu. A new directional weighted median filter for removal of random-valued impulse noise. *IEEE Signal Processing Letters*, 14(3):193–196, 2007.

[8] V. Soni, A. K. Bhandari, A. Kumar, and G. K. Singh. Improved sub-band adaptive thresholding function for denoising of satellite image based on evolutionary algorithms. *IET Signal Processing*, 7(8):720–730, 2013.

[9] Y. Kwon, K. I. Kim, J. Tompkin, J. H. Kim, and C. Theobalt. Efficient learning of image super-resolution and compression artifact removal with semi-local Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1792–1805, 2015.

[10] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366,* 2016.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR 2016*.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS 2015*.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.,* 16(8):2080–2095, 2007.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP 2007.*

[15] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7): 3142–3155, 2017.

[16] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR 2017.*

[17] K. Zhang, W. Zuo, and L. Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[18] S. Lefkimmiatis. Universal denoising networks: a novel CNN architecture for image denoising. In *CVPR 2018*.

[19] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-CNN for image restoration. In *CVPRW 2018*.

[20] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, K. Enhanced deep residual networks for single image super-resolution. In *CVPRW 2017*.

[21] S. Ioffe, S and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[22] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR 2016*.

[23] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *CVPR 2018*.

[24] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, and D. Rueckert. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR 2016*.

[25] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *ICCV 2017*.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR 2017*.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR 2016*.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015*.

[29] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW 2017*.

[30] R. Franzen. Kodak lossless true color image suite. source: http://r0k.us/graphics/kodak, vol. 4, 1999.

[31] 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 *ICCV 2001*.

[32] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR 2018*.

[33] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017.

[34] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *CVPR 2016*.

[35] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *NIPS 2018*.

[36] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[37] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122,* 2015.

[38] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[39] B. W. Jeon, G. I. Lee, S. H. Lee, and R. H. Park. Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. *IEEE Transactions on Consumer Electronics*, 49(3): 499-508, 2003.

[40] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem

solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[41] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV 2015.*

[42] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.,* 13(4):600–612, 2004.

[44] A. Hore and D. Ziou. Is There a Relationship Between Peak-Signal-to-Noise Ratio and Structural Similarity Index Measure?. *IET Image Process.,* 7(1):12–24, 2013.

[45] A. Abdelhamed, R. Timofte, M. S. Brown, et al. NTIRE 2019 Challenge on Real Image Denoising: Methods and Results. In *CVPRW 2019.*