Experimenting with the Multi-Level Wavelet Convolutional Neural Networks (MWCNN) architecture

Suhrid Subramaniam
Electrical Engineering
University of California,San Diego
s7subram@eng.ucsd.edu

Punit Agrawal
Electrical Engineering
University of California, San Diego
p8agrawa@ucsd.edu

***Abstract*—CNNs have played a pivotal role in accelerating the research in Computer Vision by helping in achieving state-of-the-art performance in many image restoration applications. However, with an ever increasing demand for higher accuracy and close-to-perfect restoration, the models used for restoration have become computationally intensive owing to the demand for a larger receptive field. Although solutions like dilating filters have been proposed and used in the past for increasing the receptive field, they too suffer from drawbacks like the presence of gridding effect and lack of correlation between neighboring pixels in the resultant pixel map. Pooling layers used alongside CNNs tend to increase the receptive field at the cost of information loss. The goal of this architecture (MWCNN) is to maximize the receptive field and minimize computational intensity while ensuring that the run-time remains low. The metric used to compare the quality of restoration is PSNR (dB). In this project, we have implemented the MWCNN architecture and experimented with different inputs and variants of the MWCNN architecture so as to be able to prove the universality of its use or disprove the architecture's usage under certain circumstances. We have used two types of data as input: Daylight images and nightlight images. We have also experimented with different types of noise (Gaussian, Poisson) and degree of noise (sigma 15, 50) to see if changing the input or noise affects the model's performance in a drastic manner. To improve the model accuracy, we also experimented with different wavelets (Haar, Daubechies, Symlet and Biorthogonal) to see if using a particular wavelet impacts the results. Our source code can be found at https://github.com/SuhridS/Wavelets_MWCNN.**

***Index Terms*—MWCNN, receptive field, Wavelets, Haar, Daubechies, Symlet, Biorthogonal**

I. Introduction

Convolutional Neural Networks (CNNs) are being extensively used for many Computer Vision applications such as image restoration [21] [22], image denoising [3] [7] [6], image super resolution [5] and object classification. The reason for their popularity largely banks on two facts. Firstly, CNN-based solutions dominate on several simple tasks by outperforming other methods by a large margin. Secondly, CNNs can be treated as modular segments of a larger model and can be plugged into any existing traditional methods. As a result of this, the non-linearities in data which could not be handled by traditional methods are now being performed by CNNs, hence improving accuracy.

There are, however, a few downsides to using CNNs. The accuracy of a model which uses CNNs depends largely on the receptive field. The receptive field can be increased by either increasing the network depth, enlarging filter size or using pooling operation. But increasing the network depth or enlarging filter size can inevitably result in higher computational cost. Pooling can enlarge receptive field and guarantee efficiency by directly reducing spatial resolution of feature map. Nevertheless, it may result in information loss. To combat these downsides, dilated convolutions were proposed which introduced ”zero holes” in the convolutional kernel. Although using dilated filters increased the receptive field, they caused sparse sampling leading to a checkerboard like pattern, also called gridding effect.

To overcome these problems, a new network called Multi-Level Wavelet Convolutional Neural Networks (MWCNN) [1] [2] was proposed. This network uses Discrete Wavelet Transform (DWT) to replace the pooling layers and Inverse Discrete Wavelet Transform (IDWT) to replace the transpose convolutions used in the UDnCNN architecture [7]. Due to invertibility of DWT, none of image information or intermediate features are lost while down-sampling the images. Moreover, both frequency and location information of feature maps are captured by DWT which is helpful in preserving detailed texture when using multi-frequency feature representation.

A comparative study of different models and their speed of performance versus the accuracy of task performance is shown in figure 1. From the figure, it can be seen that the MWCNN architecture has a very large receptive field (181x181) and its speed of performance is pretty fast in comparison to models like RED30 and DRRN [5].

This project focuses on proving or disproving the efficacy of the MWCNN architecture. Basically, we have tried to see if the model fails under specific lighting conditions or different types of noises. We have then gone on to test different wavelet transforms to see their effects on the model and results. The subsequent sections go on to elaborate more on the experiments we performed and the insights we gained from them.

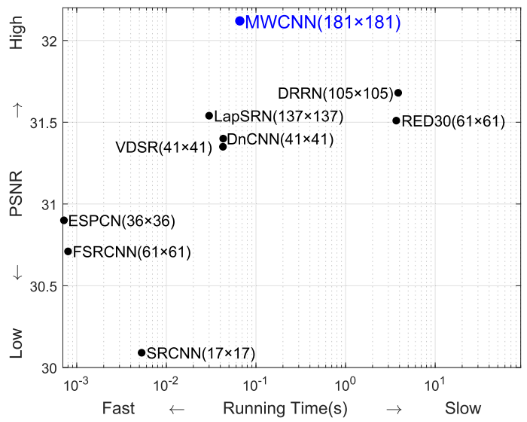


Fig. 1. Receptive field vs speed of performance for different models [1]

II. Related Work

The conventional methods used for image restoration are application specific. For image denoising, the architecture that is usually preferred is the feed-forward denoising convolutional neural networks (DnCNNs) [3], U-net DnCNN (UDnCNN) [7] or the Dilated convolutions UDnCNN (DUDnCNN) [6] architectures. These architectures make use of residues / skip connections to provide better detailing. They also use max pooling and transpose convolution in their contracting and expanding subnetworks.

A lot of research into the use and incorporation of Wavelets into the Denoising architectures has been done in recent years. Bae et al. [23] in their network WavResNets discovered that CNNs can benefit from learning on wavelet subbands with features having more channels. Hence, the first wave of wavelet based image restoration models came into existence. Guo et al. [24] focused on making the network deeper for improving super resolution results. These models focus on a single layer of wavelets to improve their task specific results. MWCNN however encorporates multi-level wavelet transform to enlarge receptive field while barely increasing computational complexity.

III. Network Architecture

The skeletal structure of our network is based on the multi-level wavelet packet transform (WPT) and U-net based DnCNN. On top of this skeleton, we add Convolutional Layers (CNNs) to account for the non-linearities in the network. DWT and IDWT handle the operations performed by the pooling and transpose convolutional layers. We also do not need to use skip connections as the wavelets encompass both, the frequency as well as the spatial information of a feature map. The network architecture can be broken down and explained as follows:

A. Wavelet Packet Transform

The wavelet packet transform has a number of applications. One of these involves the calculation of the "best basis", which is a minimal representation of the data relative to a particular cost function. The "best basis" is used in applications that include noise reduction and data compression. More specifically, Wavelet Packet Transform (WPT) can be regarded as a collection of orthonormal transforms, each of which can be readily computed using a very simple modification of the pyramid algorithm for the DWT (Discrete Wavelet Transform).

The forward network of the wavelet packet transform calculates a low pass (scaling function) result and a high pass (wavelet function) result. The low pass result is a smoother version of the original signal (the average, in the case of the Haar wavlet). The low pass result recursively becomes the input to the next wavelet step, which calculates another low and high pass result, until only a single low pass (20) result is calculated.

Once this is done, we find the best basis for the data by minimizing a cost function. The wavelet basis is the range in the vector over which the scaling and wavelet functions are non-zero. The wavelet transform takes an input data set and represents it in a new form. No information is lost, however, and the result of the wavelet transform can be perfectly reconstructed into the original data. Assuming that the data is not random, the representation of the data produced by the wavelet transform may be more compact (consisting of smaller values) than the original data set. The threshold function is a popular cost function.

The best basis obtained in the above step is then used to reconstruct the original image. This step is also known as the Inverse Transform from the Best Basis Set. This is done by iteratively stacking and combining the basis images [25].

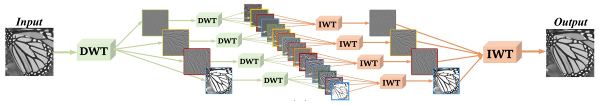


Fig. 2. Wavelet Packet Transform representation [1]

B. Multi Level WPT to MWCNN

In 2D discrete wavelet transform (DWT), four filters, i.e. $f_{LL}, f_{LH}, f_{HL}, f_{HH}$ are convolved with an image x. The convolution results are then down-sampled to obtain the four sub-band images x_1, x_2, x_3, x_4. For example x_1 is defined as $(f_{LL} \otimes x) \downarrow 2$. Even though the down-sampling operation is deployed, due to the bi-orthogonal property of DWT, the original image x can be accurately reconstructed by the inverse wavelet transform (IWT), i.e., $x = IWT(x_1, x_2, x_3, x_4)$. In multilevel WPT the sub-band images are further processed with DWT to produce decomposition results i.e. each x_i produce four more decomposed images.. Recursively we can have multilevel of decomposition. Similarly we deploy the four sub-band image filters at reconstruction stage. Consequently we can reconstruct the original image accurately.

In this work, we further extend WPT to multi-level wavelet-CNN (MWCNN) by adding a CNN block between any two levels of DWTs. After each level of transform, all the sub-band images are taken as the inputs to a CNN block to learn a compact representation as the inputs to the subsequent level of transform. It is obvious that MWCNN is a generalization of multi-level WPT, and degrades to WPT when each CNN block becomes the identity mapping. Due to the bi-orthogonal property of WPT, our MWCNN can use sub-sampling operations safely without information loss. Moreover, compared with conventional CNN, the frequency and location characteristics of DWT is also expected to benefit the preservation of detailed texture.

C. MWCNN Architecture

Our MWCNN architecture inserts CNN layers after each level of DWT/IDWT as shown in Fig 3. Each layer of CNN comprises of convolution kernel of $(3X3)$, stride of $(1, 1)$, batch normalization and rectified linear unit (ReLU) unit. ADAM optimizer is used while training the MWCNN model. At DWT/IDWT we use different wavelet filters e.g. Haar, Daubechies, Symlet, Biorthogonal. In general, we set the number of features extracted as 64 for single order filters, hence the number of network weights are in multiples of 64 throughout the network. For higher order filters, we used 66 features, hence, the number of model weights in any layer is a multiple of 66.

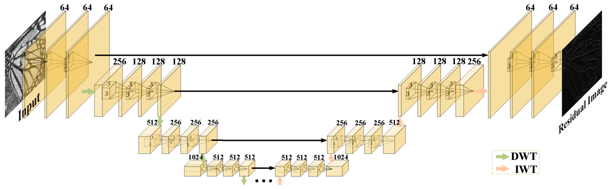


Fig. 3. Multi-Level Wavelet Convolutional Neural Networks architecture [1]

IV. Experiments

In this section, we will elaborate on the experiments we performed and the thought process behin choosing these experiments.

A. MWCNN vs DnCNN

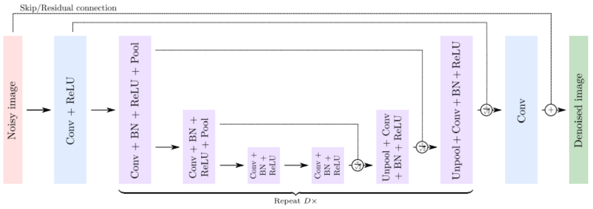


Fig. 4. Denoising CNN (DnCNN) architecture [3]

The first experiment we performed was to see if the MWCNN architecture (fig 3) performed better that its ”wavelet-less” counterpart DnCNN (fig 4) under all noise values. We trained the model using Gaussian noise using sigma 15, 30 and 50.

B. Effect of type and scale of noise

Next, we tested the MWCNN network’s efficacy by comparing the PSNR resuts using Gaussian and Poisson noise. We trained the model using Gaussian and Poisson noise of sigma 15, 30 and 50 on the same training set and evaluated the results on the Set5 dataset.

C. Effect of noise level and lighting conditions

The next experiment we performed was to test the models trained on different noise levels (sigma) to see if a specific model performed better than the others. We also tested the model on different datasets. Firstly, we tested the model on the same test set we were evaluating it on while training. We then tested it using a set of ’Daylight’ images which contained sharp contrast between colors. After this, we tested the model using Low-light’ images which were take during the night. The reason we chose this experiment was to see if different lighting conditions improves or worsens the denoising effect of the model.

D. Effect of different wavelets

We trained the model using Haar, Symlet5, Dabuchies 5, Bi-Orthogonal 2.4 wavelets and saw if there are any improvements in the model’s denoising performance.

E. Cropped model vs New model

The higher order wavelets we used produced 4 sub-band images of size larger than the results produced by Haar as expected. We originally had to crop the four sub-band outputs to keep the dimensions the same as that produced by the Haar filter. Later, we thought of remodelling the number of weights to accommodate the higher order filters without cropping.

V. Results

A. MWCNN vs DnCNN

This section holds all results pertaining to the first experiment we performed (IV-A).

In fig 5 we compare the performance of our MWCNN network with DNCNN network. Here we observe that MWCNN network converges faster than DNCNN network and offers higher PSNR performance. This is as expected.

B. Effect of type and scale of noise

In fig 6 we compare the performance of MWCNN network for Gaussian and Poisson noise distribution with different noise levels. We observe that MWCNN performance is similar for both noise distribution. Hence, our model is noise invariant.

C. Effect of noise level and lighting conditions

The figure below shows a scatter plot of different test images along with their PSNR.

In fig 7, we compare the performance of our network under different test conditions. We train our network for various noise level. And then test it for various noisy datasets. Summary of PSNR achieved is also given at fig 8. It is observed that its best to train the network over moderate scale noise

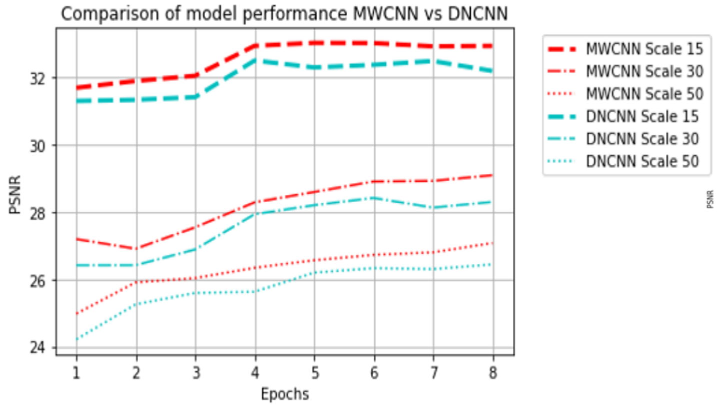


Fig. 5. MWCNN VS DNCNN peroformance comparison. DNCNN PSNR is plotted for every 10th epoch

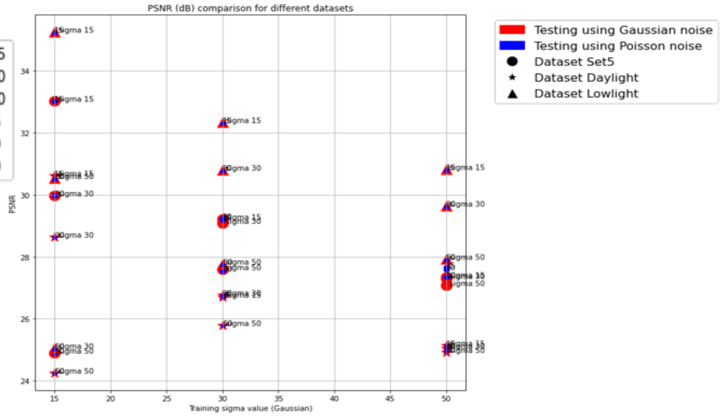


Fig. 7. Comparison of performance under various noise distribution and data sets.

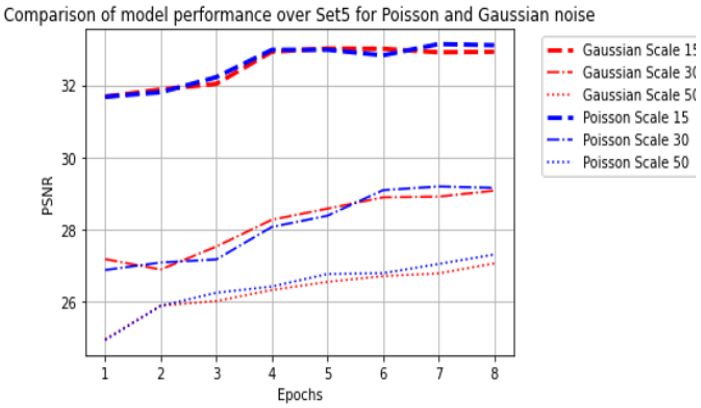
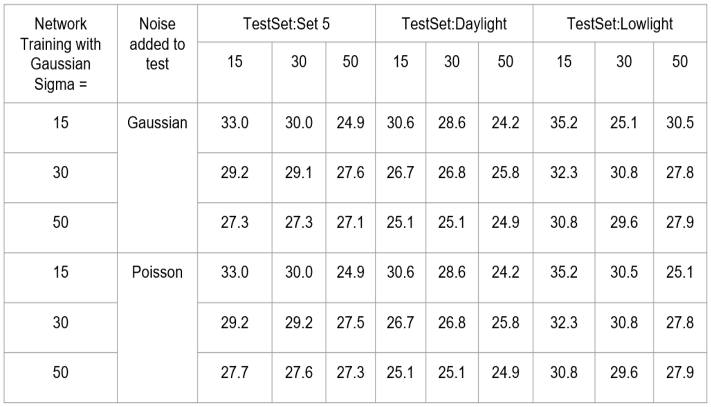


Fig. 6. Gaussian vs Poisson training PSNR



Network Training with Gaussian Sigma =	Noise added to test	TestSet:Set 5			TestSet:Daylight			TestSet:Lowlight		
		15	30	50	15	30	50	15	30	50
15	Gaussian	33.0	30.0	24.9	30.6	28.6	24.2	35.2	25.1	30.5
30		29.2	29.1	27.6	26.7	26.8	25.8	32.3	30.8	27.8
50		27.3	27.3	27.1	25.1	25.1	24.9	30.8	29.6	27.9
15	Poisson	33.0	30.0	24.9	30.6	28.6	24.2	35.2	30.5	25.1
30		29.2	29.2	27.5	26.7	26.8	25.8	32.3	30.8	27.8
50		27.7	27.6	27.3	25.1	25.1	24.9	30.8	29.6	27.9

Fig. 8. Data in table

level if the test dataset is unknown, also MWCNN performs well under different test sets like set, Daylight, Lowlight.

It can also be seen that MWCNN denoises the low-light images the best followed by the evaluation set followed by the daylight images. This can be explained as the lowlight images have a large portion of the image covered in black pixels while the daylight images have a lot of contrast between different color regions. Denoising on a low level is basically performing low pass fitering. This affects the highly contrasting daylight images more than the relatively smoother lowlight images.

Table 8 shows a tabular representation of the scatter plot for better understanding.

D. Effect of different wavelets

We compare the performance of MWCNN by changing the wavelets used. Here we compare the performance of Haar, Biorthogonal2.4, Symlet5,Daubechies5. Since the filter length is higher in other wavelets compared to Haar, we truncated our inputs to CNN network in successive stages to keep the dimensions of our CNN network same. Hence there was a loss of information, which is clearly observed from the PSNR performance in fig 9 . From the plot, it is evident that Haar performs the best in the truncated network.

E. Cropped model vs New model

Furthermore we now change the dimension of our network to accommodate larger input, accounting for larger filter size compared to Haar, fig 10 shows Daubechies5 psnr improvement with updated network. Haar offers best performance among all other wavelets for our image denosing use case.

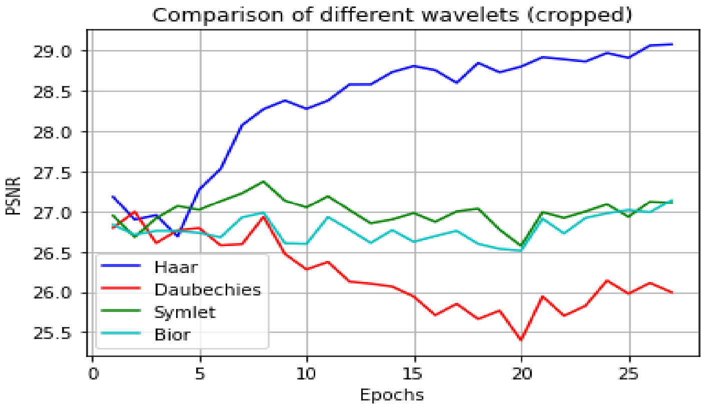


Fig. 9. Comparison of performance using different wavelets

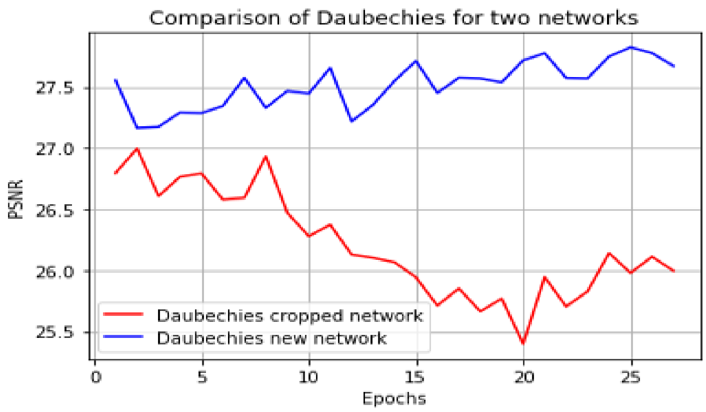


Fig. 10. Training PSNR comparison under cropped input vs resized network

Comparison of Haar vs Daubechies5 is shown in fig 11 we see that by taking difference between input image and reconstructed image on rightmost figures. It is clearly seen that Haar is preseving the image features.

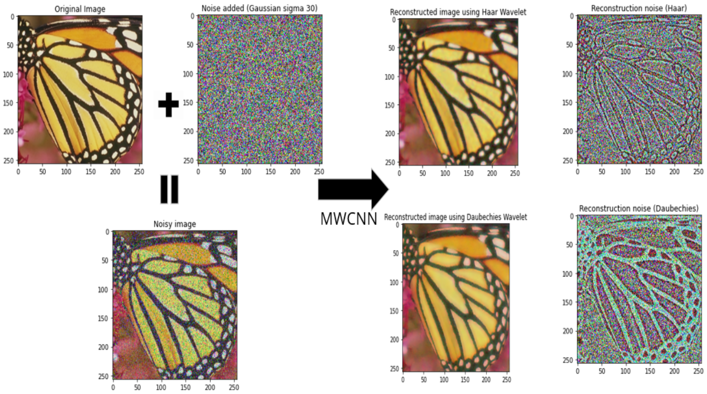

Original Image
Noise added (Gaussian sigma 30)
Noisy image
MWCNN
Reconstructed image using Haar Wavelet
Reconstructed image using Daubechies Wavelet
Reconstruction noise (Haar)
Reconstruction noise (Daubechies)
0
50
100
150
200
250

Fig. 11. Haar vs Daubechies Performance comparison

VI. Conclusion

To conclude, we see that using wavelets definitely improves denoising PSNR as seen from figure 5. The model trained is noise invariant to some extent. Gaussian and Poisson noises are almost denoised to the same extent as seen from figure 6. For real life applications, training the denoising model for sigma 30 is ideal. This model is capable of denoising the images for different noise levels to a very good extend in comparison to other models. Sharpening filters can then be used to try to reconstruct the edges to further improve the results.

For Denoising application, Haar performs the best. This is followed by Bi-orthogonal 2.4, Symlet and Daubechies wavelets for the cropped network. On remodelling the network, Daubechies performs much better than the cropped Daubechies network. This shows that the other wavelets also have scope for improvement and could exceed Haar's performance.

Acknowledgment

We would like to thank Prof.Truong Nguyen and TA Yiqian Wang for guiding us throughout this project and teaching us Filter Banks and Wavelets and their real world applications.

VII. Contribution

Both of us worked equally on the project. Our overall contribution to the project is 50-50.

References

[1] Pengju Liu and Hongzhi Zhang and Wei Lian and Wangmeng Zuo (2019). Multi-level Wavelet Convolutional Neural NetworksCoRR, abs/1907.03128.

[2] P. Liu, H. Zhang, W. Lian and W. Zuo, "Multi-Level Wavelet Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 74973-74985, 2019, doi: 10.1109/ACCESS.2019.2921451.

[3] Kai Zhang and Wangmeng Zuo and Yunjin Chen and Deyu Meng and Lei Zhang (2016).Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image DenoisingCoRR, abs/1608.03981.

[4] Ying Tai and Jian Yang and Xiaoming Liu and Chunyan Xu (2017). MemNet: A Persistent Memory Network for Image RestorationCoRR, abs/1708.02209.

[5] Tai, Y., Yang, J., & Liu, X. (2017). Image Super-Resolution via Deep Recursive Residual Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[6] Wang, Tianyang & Sun, Mingxuan & Hu, Kaoning. (2017). Dilated Deep Residual Network for Image Denoising. 1272-1279. 10.1109/ICTAI.2017.00192.

[7] Olaf Ronneberger and Philipp Fischer and Thomas Brox (2015). U-Net: Convolutional Networks for Biomedical Image SegmentationCoRR, abs/1505.04597.

[8] Singh, R., Vasquez, R., & Singh, R. (1997). Comparison of Daubechies, Coiflet, and Symlet for edge detection. In Visual Information Processing VI (pp. 151-159).

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (pp. 1097–1105). Curran Associates Inc..

[10] Fisher Yu, & Vladlen Koltun. (2016). Multi-Scale Context Aggregation by Dilated Convolutions.

[11] Zhang, Kai & Zuo, Wangmeng & Zhang, Lei. (2017). FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. IEEE Transactions on Image Processing. PP. 10.1109/TIP.2018.2839891.

[12] Akito Takeki, Daiki Ikami, Go Irie, & Kiyoharu Aizawa. (2018). Parallel Grid Pooling for Data Augmentation.

[13] G. Strang and T. Nguyen, Wavelets and Filter Banks. SIAM, 1996.

[14] A. K. Moorthy and A. C. Bovik, "Visual Importance Pooling for Image Quality Assessment," in IEEE Journal of Selected Topics in Signal Processing, vol. 3, no. 2, pp. 193-201, April 2009, doi: 10.1109/JSTSP.2009.2015374.

[15] Agustsson, E., & Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[16] Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., & others (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[17] Timofte, R., Gu, S., Wu, J., Van Gool, L., Zhang, M.H., Haris, M., & others (2018). NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[18] Timofte, R., Gu, S., Wu, J., Van Gool, L., Zhang, M.H., Haris, M., & others (2018). NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[19] Ignatov, A., Timofte, R., & others (2019). PIRM challenge on perceptual image enhancement on smartphones: report. In European Conference on Computer Vision (ECCV) Workshops.

[20] Anaya, J., & Barbu, A. (2018). RENOIR – A dataset for real low-light image noise reduction Journal of Visual Communication and Image Representation, 51, 144–154.

[21] C. Dong, C. C. Loy, K. He and X. Tang, ”Image Super-Resolution Using Deep Convolutional Networks,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.

[22] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, Ming-Hsuan Yang. (2017). Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution.

[23] T. Guo, H. S. Mousavi, T. H. Vu and V. Monga, ”Deep Wavelet Prediction for Image Super-Resolution,” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, 2017, pp. 1100-1109, doi: 10.1109/CVPRW.2017.148.

[24] Y. Han and J. C. Ye, ”Framing U-Net via Deep Convolutional Framelets: Application to Sparse-View CT,” in IEEE Transactions on Medical Imaging, vol. 37, no. 6, pp. 1418-1429, June 2018, doi: 10.1109/TMI.2018.2823768.

[25] Baig, Sobia Rehman, Fazal Mughal, M.. (2006). Performance Comparison of DFT, Discrete Wavelet Packet and Wavelet Transforms, in an OFDM Transceiver for Multipath Fading Channel. 1 - 6. 10.1109/INMIC.2005.334509.