1. Raksha Ramakotti, 2. Surekha Paneerselvam, 3, Vinoth Kumar K, 4. Sachin Kumar, 5. Jebarani Evangeline S

Amrita School of Engineering, Bengaluru, India (1) (2), New Horizon College of Engineering, Bengaluru, India (3) & Post-Doctoral Fellow in South Ural State University, Chelyabinsk, Russian Federation (4), South Ural State University, Chelyabinsk, Russian Federation (4), SNS College of Engineering, Coimbatore India (5)
ORCID: 1. 0000-0001-8982-3879; 2. 0000-0003-4749-0564; 3. 0000-0002-3009-1658; 4. 0000-0003-3949-0302; 5. 0000-0002-2762-285X

doi:10.15199/48.2023.02.04

An Analysis of Denoising Neural Networks for Noise Removal in Images

***Abstract**. Clean images, when subjected to prolonged transmission, improper image acquisition or conditioned to multiple feature changes, lead to image tarnishing due to unwanted noisy pixels. This proposes to be a major threat in image-processing and computer vision fields. With the evolution of denoising models in the field of Neural Networks, efficient noise removal has become achievable, in a real-time scenario. In this work, two approaches to noise modelling have been considered, i.e., noise as an inverse problem and noise as a residual problem, this has been done by constructing convolutional auto encoders and denoising convolutional networks and their performance in the process of noise removal has been evaluated based on Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM).*

***Streszczenie.** In this place is allowed to use Google Translation tool Czyste obrazy poddane przedłużonej transmisji, niewłaściwej akwizycji obrazu lub poddane wielokrotnym zmianom cech prowadzą do zmatowienia obrazu z powodu niechcianych zaszumionych pikseli. Sugeruje to, że jest to poważne zagrożenie w dziedzinie przetwarzania obrazu i widzenia komputerowego. Wraz z ewolucją modeli odszumiania w dziedzinie sieci neuronowych, efektywne usuwanie hałasu stało się osiągalne w scenariuszu czasu rzeczywistego. W niniejszej pracy rozważono dwa podejścia do modelowania hałasu, tj. hałas jako problem odwrotny i hałas jako problem rezydualny. Dokonano tego poprzez skonstruowanie autoenkoderów splotowych i odszumianie sieci splotowych, a ich wydajność w procesie usuwania hałasu oceniane na podstawie stosunku sygnału szczytowego do szumu (PSNR) i wskaźnika podobieństwa strukturalnego (**SSIM) (Analiza możliwości wykorzysyania sieci neuronowych do odszumiania obraziu**)*

Keywords: Denoising Models, Auto encoders, Residual Networks, Deep Learning.
Słowa kluczowe: Modele odszumiania, automatyczne kodery, sieci rezydualne, głębokie uczenie.

.

Introduction

In the fields of Computer Vision and Image Processing, one of the major challenges faced is handling images that are tampered with noise [1]. Digital images, when subjected to prolonged exposure of environment, multiple feature changes or random variations of brightness and color information; lead to tarnishing of images with degradation of the image quality. Noise can be acquired during processes such as image acquisition, subjecting the image to heavy coding, transmitting the image in an unpredictable channel and during subjective processing. The challenge of a good image denoising technique is that not only does it have to identify and remove the noise, but conduct this process without compromising on the image reconstructed quality.

The conventional methods adapted though seem to be effective, however convey certain disadvantages. Firstly, based on the nature of the image, the adapted methodology's parameters have to be set manually, each and every time. Secondly, the process of image denoising is mathematically and computationally complex and expensive. Also, in order to ensure that the background image is not lost, some amount of pre-processing technique may have to be adapted [2]. However, with the development of Artificial Intelligence, the above challenges can be bridged with the introduction of Neural Networks [3]. In the previous work, various such Neural Networks have been developed to solve multiple objectives, like classification, prediction, image steganography, image spam classification, etc. [4-9]. Image denoising have also been performed using neural networks in the recent times. A survey of the recent models providing state of the art results is presented in [3]. Auto encoders with skip connections are used in the Denoising model architecture showed in [10]. Multiple objectives of denoising and data classification is done in [11]. In [12], a comparative analysis on various convolutional models for rain-streak modelling has been addressed showcasing the working of auto encoders and residual networks. Other related work can be referred from [13-15].

In this paper, a comparative analysis is done by treating noise as an implicit and inverse problem [16] and as a residual problem [17]. The former is addressed by constructing denoising auto encoders, which aid in reconstructing clean images in such a way that the models are trained to treat the output image as a function of noisy input images. Denoising Convolutional Neural Networks (DnCNNs) are built to treat noise as a residual issue. The datasets used are COCO and BSD, and are subjected to an additive Gaussian noise with a variance of 10 and 50. The criteria of mixed denoising is also explored to evaluate the stability of the models. The performance is estimated based on Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM). The following sections consist of Methodology, where the description of the models are presented and in the Analysis and Results section, the model performance based on PSNR and SSIM are evaluated. The paper culminates with Conclusion and Future Scope.

Methodology

In this section, a brief description of the datasets used is presented. Also, the various approaches to noise and the construction of the proposed denoising models are elucidated. The images are treated in their digital format and represented as (m x n x p), where (m x n), represents the image size and p, the number of color channels. The dataset chosen for training the model is COCO, i.e., Common Object in Context [18]. This color dataset consists of objects captured from everyday scenes. This adds some "context" to the objects captured in the scenes. COCO provides multi-object labeling, segmentation mask annotations, image captioning, key-point detection and panoptic segmentation annotations with a total of 81 categories, making it a very versatile, flexible and multi-purpose dataset. This dataset being open-source was particularly chosen as it introduces the concept of generalization amongst images through non-iconic images. A subset of 2000 images was selected for the analysis and were resized to a standard size of (128 x 128). For testing the models, 200 images of Berkeley Segmentation Dataset (BSD) were used [19]. This dataset is widely used to developing new boundary detection algorithms, and for developing a benchmark for the same. Noise is a synonym of the unwanted signal that does not contain any useful information. Noise on an image can simply be translated to random variation of brightness or color information in images. More often, its impact has always been posed as a major challenge in the Image Processing domain as it abases the constitution of the image thereby posing serious threats to image quality and the information that it contains. As pointed out in the previous section, it can be adapted during processes such as image acquisition, subjecting the image to heavy coding, transmitting the image in an unpredictable channel and during subjective processing. In this work, the clean COCO dataset is subjected to a Gaussian Noise, which usually arises as a threat during image acquisition. The nature of such a noise is treated to be additive in nature i.e,

(1) $Noisy\ Image\ (x') = Clean\ Image\ (x) + Noise$ (1)

The probability density function of a Gaussian variable z is given by,

(2) $PDF\ (z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

where μ represents mean and σ represents variance. Here, the noise's level is defined with zero mean and variances 10 and 50. The resultant image x' is a noisy image, which is the actual input to the denoising models. Noise as an inverse problem aims in mapping a function to implicitly learn the constituents of a clean image from that of an image tampered with noise. The reconstructed clean image obtained at the decoder end is simply a function of the input image blemished with noise. Such a type of approach is achieved by constructing the autoencoder models. These networks are unsupervised in nature which aim to learn a representation (encoding) for a set of data, "ignore" the noise component and reconstruct the image using a complimentary decoder. Here, it is ensured that the input and output image sizes are the same. A general architecture of autoencoders consist of the three components namely: The input to an encoder is a noisy image. The encoder has convolutional layers, which aid in xtracting important image features. It also has Max Pooling Layers which assist in reducing the dimensionality of the image to present in an "encoded form". Mathematically, the encoder tries to map a deterministic non-linear function g_1 to the output of the encoder l as shown in Equation 3,

(3) $l = g_1(Wx' + b)$

where, W is a 2 dimensional weight matrix, x' is the noisy image and b is the encoder bias vector and l is the latent space.

Latent Space: Sandwiched between the encoder and the decoder, the latent space presents the condensed or compressed form of the input image to a bottleneck.

Decoder: With the latent space as its input, the decoder's functionality is to reconstruct the clean image using convolution transpose and upsampling layers. It has a mirroring structure of the encoder. The mathematical representation is given in Equation 4,

(4) $r = g_2(Vl + b')$

where, V is a 2 dimensional weight matrix whose dimensions are same as that of the dimensions of transpose of W, b' is the decoder bias vector and r is the reconstructed clean image. Here, instead of constructing the clean image directly, the residue, or in other words, the noisy distribution is determined by the model. Then, it subtracts this residue with the noisy image in order to obtain the clean image. The mathematical representation is showed in Equation 5,

(5) $r = x' - f(n)$

where, r is the reconstructed clean image, x' is the noisy input image, n is the noise quantity and f(n) is the function of the noisy pattern obtained by the model. The objective of the model is to present the reconstructed image r very similar to that of the ground truth or the clean image x. The model fabricated to achieve this is known as Denoising Convolutional Neural Network (DnCNN). The DnCNN has a convolutional layer that aids in extracting the important features of the image. The model also has sandwiched domino layers of convolution and batch normalization. The input images are is divided into several parts which are called mini-batches, for a single epoch. The batch normalization layer Normalises a layer input by subtracting the mini-batch mean and dividing it by the mini-batch standard deviation. As observed in Equation 6, for the kth mini-batch, the batch normalized output $b\hat{n}_k$ is given by,

(6) $b\hat{n}_k = \frac{bn_k - E[bn_k]}{\sqrt{Var[bn_k]}}$

where $E[bn_k]$ represents the expectation of the kth mini-batch and the denominator represents its standard deviation. The usage of batch normalization alleviates the effect of internal-covariance shift, which is caused due to effect of randomness in the parameter initialization and input data during training process. This process thus standardizes inputs to a layer for each mini-batch thereby stabilizing the learning process reducing the number of training epochs. The problem of model overfitting is resolved and the process of model learning is accelerated [20]. The residual pattern obtained is then subjected to a subtraction layer in order to obtain the reconstructed clean image.

Proposed Denoising Models

The architectures of the proposed models used for the comparative analysis are discussed in the following sections.

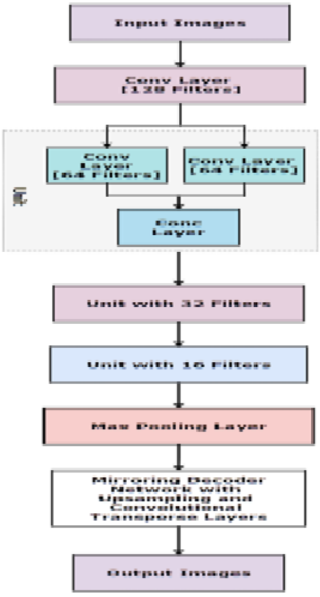

Fig. 1 Architecture of DLCA Model

1) Double Layer-Concatenation Autoencoder (DLCA)

The DLCA architecture is presented in Fig. 1 treats noise as an inverse problem. Here, the input images are fed to a single layer containing 128 filters. Then, the output is fed to 2 layers in a parallel fashion. The resultant feature maps are concatenated using the conc layer, before feeding to the next unit, where a unit consists of 2 parallel convolutional layers (conv layer) containing same filter size and count and one concatenation layer. DLCA consists of 3 such units with a uniform filter size of (3x3) dimensionality and the resultant feature map is fed to a Max Pooling layer that reduces the dimensionality of the image to (64 x 64) from (128 x 128). The decoder consists of a mirroring structure to that of the encoder where the up sampling layer and convolutional transpose layers are the corresponding replacements to Max Pooling layer and convolutional layers present in the encoder.

2) Varied-Filter Size Residual Network (VFRN)

As observed in Fig. 2, the VFRN aids in treating noise as a residual problem. Here, the input image is fed to a single layer containing 128 filters of (3 x 3) filter size. Post that, the model is subjected to 15 layers of convolution + batch normalization layers. The first ten layers consist of 64 filters where the filter sizes vary from (5 x 5) to (1 x 1). The larger filters such as (3 x 3), (4 x 4) and (5 x 5) aim in capturing the bigger picture, thereby providing a substantial understanding of the background image. Smaller filter sizes aid in capturing the intricate details of the image. The subsequent five layers consist of 32 filters of a uniform (3 x 3) filter size, which is followed by a convolution layer and a subtraction layer. The output of the latter layer presents the reconstructed clean image.

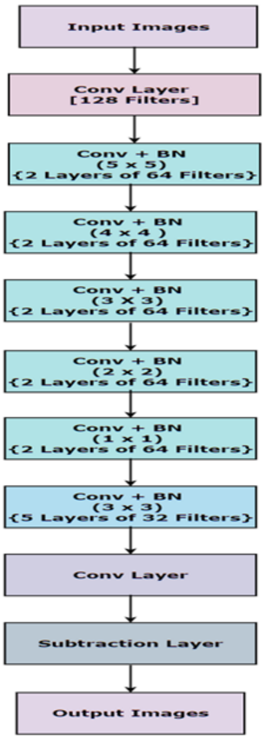
Input Images
Conv Layer [128 Filters]
Conv + BN (5 x 5) {2 Layers of 64 Filters}
Conv + BN (4 x 4) {2 Layers of 64 Filters}
Conv + BN (3 x 3) {2 Layers of 64 Filters}
Conv + BN (2 x 2) {2 Layers of 64 Filters}
Conv + BN (1 x 1) {2 Layers of 64 Filters}
Conv + BN (3 x 3) {5 Layers of 32 Filters}
Conv Layer
Subtraction Layer
Output Images

Fig. 2 Architecture of VFRN Model

Analysis and Results

The evaluation metrics used to estimate the performance of the model architectures are presented in this section. Also the description of the model analysis and results is showcased. All the models were implemented in Python 3 script in Google Colab. In Image Processing and Computer Vision applications, where both the input and the output are digital images, the quantifying evaluation metrics used to determine the performance of the are PSNR and SSIM. The details are provided in the following subsections. The Peak Signal to Noise Ratio (PSNR) [21] represents the ratio between the maximum power value of a signal and the power of distorting noise that affects the quality of its representation. The higher the PSNR, the better the quality of the reconstructed image. It is measured in decibels. The formula for PSNR is given as,

$$PSNR = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad (7)$$

where, MSE stands for Mean Squared Error and the numerator term represents the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. The Structural Similarity Index (SSIM) [22] is an enduring metric that quantifies the degradation of an image quality due to extensive image processing or modelling. SSIM is given by,

$$SSIM(I,K) = \frac{(2m_I m_O + c_1)(2s_{IO}+c_2)}{(m_I^2+m_O^2+c_1)(s_I^2+s_O^2+c_2)} \qquad (8)$$

where, m_I represents the average of the input image I, m_O represents the average of the output image O, s_I represents the variance of the input image I, s_O represents the variance of the output image O, s_{IO} represents the covariance of the images I and O. For the analysis between the proposed models, certain parameters were maintained constant throughout. These are tabulated in Table 1.

Table 1. Constant Parameters for the Analysis

Parameters	Values
Train Image Count	2000
Test Image Count	200
Image Size	(128 x 128 x 3)
Mean of Gaussian Noise	0
Noise Factor	0.005
Batch Size	10
Epochs	100
Loss	Mean Squared Error
Activation Function	ReLU
Optimizer	Adam
DLCA Model Parameters	496,067
VFRN Model Parameters	616,707

The models were evaluated by taking the mean of all the images individual PSNRs and SSIMs observed in Table 1, the total parameters of DLCA and VFRN are 4,96,067 and 6,16,707 respectively. The analysis was performed on 2 different noise levels, i.e., for variance values 10 and 50. These values are presented in Tables 2 and 3 respectively. The evaluation metrics for train dataset is computed on 2000 images of COCO dataset and tested on 200 images of BSD dataset. All the images are resized to (128 x 128) and have 3 color channels. As observed in Table 2, For a lower noise level of σ = 10, there is an increase of 7.846 dB and 11.097 dB respectively in the train and test PSNRs when VFRN model was used. Also, SSIMs greater than 0.99 was obtained. The varied-filter denoising approach presented state of the art results by minimizing the loss after the 100th epoch. The test loss was in the order of 10^{-5}. For a higher noise level of σ = 50, the DLCA model showcased similar performance in both PSNR and SSIMs. Infact the test PSNR was improved by 1.666 dB. In case of the VFRN model, the train SSIM was maintained. However, there was a marginal drop in the performances of the other metrics. The train PSNR showcased a performance dip by 4.03%. There was however a significant drop in the test PSNR's value, and this was observed to be by 28.62%. The MSE losses after the 100th epoch for the test and train datasets were 0.0031 and 0.0001 respectively. These values are showed in Table 3. A case of mixed denoising was addressed in order to evaluate the stability of the model. Sometimes, due to improper data segregation, the images may not have the same noise level. This also tests the robustness of the model as mixed denoising introduces randomization. Therefore, half the images were subjected to a variance of 10 and the remaining with a variance of 50. From Table 4, it is evident that, DLCA still showcased a consistent PSNR performance of around 31 dB and SSIM of 0.95 for both COCO and BSD datasets. The loss obtained were between 0.0007-0.0008. However, VFRN showcased improved performance by presenting a train PSNR of 37.562 dB, train SSIM of 0.988, test PSNR of 35.503 dB and test SSIM of 0.960. The MSE loss obtained were 0.0002 and 0.0005 respectively.

Table 2. Evaluation Metrics Tabulation for Noise level of 10

Evaluation Metrics	DLCA	VFRN
Train PSNR (dB)	31.385	39.231
Test PSNR (dB)	32.276	43.373
Train SSIM	0.960	0.991
Test SSIM	0.962	0.996
MSE Train Loss	0.00110	0.00010

Table 3. Evaluation Metrics Tabulation for Noise level of 50

Evaluation Metrics	DLCA	VFRN
Train PSNR (dB)	31.805	37.650
Test PSNR (dB)	33.942	30.957
Train SSIM	0.969	0.990
Test SSIM	0.978	0.919
MSE Train Loss	0.00090	0.00010

Table 4. Evaluation Metrics Tabulation for Mixed Noise level

Evaluation Metrics	DLCA	VFRN
Train PSNR (dB)	31.089	37.562
Test PSNR (dB)	31.688	35.503
Train SSIM	0.957	0.988
Test SSIM	0.956	0.960
MSE Train Loss	0.0008	0.00020

From the above analysis, it can be observed that, the proposed models DLCA and VFRN showcased consistent performance, on subjecting the images to various noise levels. However, treating noise as a residual problem, led to better cleaning of the models. The output images are presented in Fig. 3 and 4, where the first row contains the sample images of COCO dataset and the last row of BSD dataset. On comparison, it can be observed that the top-down varied filter sizes aid in capturing and restoring the intricate details of the images in all the cases and hence proves to be the best model.

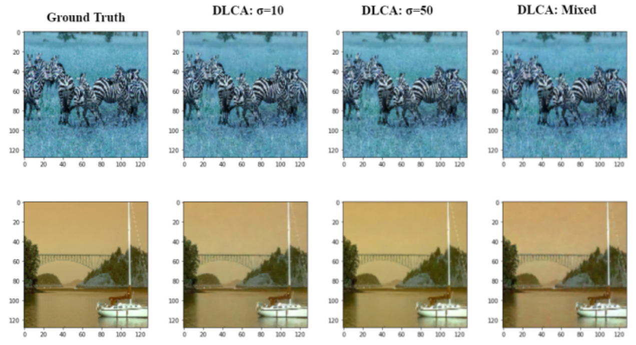

Fig.3.Train and Test Sample Output Images for Various Noise Levels obtained from DLCA Model

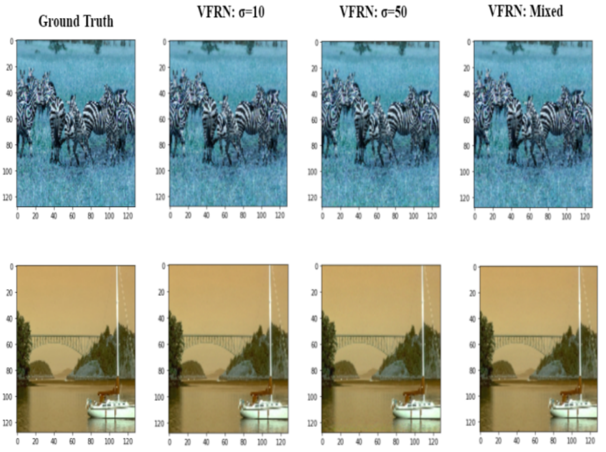

Fig.4.Train and Test Sample Output Images for Various Noise Levels obtained from VFRN Model

Conclusion and Future Scope

In this work, a comparative analysis on noise modelling is done. Two architectures were proposed, i.e., Double Layer-Concatenation Autoencoder and Varied-Filter Size Residual Network, where VFRN showcased an edge in the performance, when subjected to 2000 train images from COCO dataset and 200 test images from BSD dataset. The model showcased a train PSNR of 37.562 dB, train SSIM of 0.988, test PSNR of 35.503 dB and test SSIM of 0.960. This work can be improved by including skip connections in autoencoders. Also, GANs and ensemble techniques can be adopted to improvise the model's performance. The models can also be trained by subjecting the images to other noise types and explore the working of the model for real-time noisy images.

Authors. Raksha Ramakotti, Surekha Paneerselvam, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru Campus. Vinoth Kumar K, New Horizon College of Engineering, Bengaluru, India – 560103 & Post-Doctoral Fellow in South Ural State University, Chelyabinsk, Russian Federation, Dr. Sachin Kumar, South Ural State University, Chelyabinsk, Russian Federation, Dr. Jebarani Evangeline S, SNS College of Engineering, Coimbatore India.
E-mail: kvinoth_kumar84@yahoo.in, p_surekha@blr.amrita.edu, bl.en.p2ebs19009@bl.students.amrita.edu, kumars@susu.ru, reachjebarani@gmail.com

REFERENCES

[1] Tian, Chunwei, Yong Xu, and Wangmeng Zuo. "Image denoising using deep CNN with batch renormalization." *Neural Networks* 121 (2020): 461-473.
[2] Tian, Chunwei, Yong Xu, Lunke Fei, Junqian Wang, Jie Wen, and Nan Luo. "Enhanced CNN for image denoising." *CAAI Transactions on Intelligence Technology* 4, no. 1 (2019): 17-23.
[3] Tian, Chunwei, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. "Deep learning on image denoising: An overview." *Neural Networks* (2020).
[4] Akarsh, S., K. Simran, Prabaharan Poornachandran, Vijay Krishna Menon, and K. P. Soman. "Deep learning framework and visualization for malware classification." In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 1059-1063. IEEE, 2019.
[5] Selvin, Sreelekshmy, R. Vinayakumar, E. A. Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. "Stock price prediction using LSTM, RNN and CNN-sliding window model." In 2017 international conference on advances in computing, communications and informatics (icacci), pp. 1643-1647. IEEE, 2017.
[6] Subbiah, Uma, Rahul Vinod Kumar, Shruti Ajithkumar Panicker, R. Aathith Bhalaje, and S. Padmavathi. "An Enhanced Deep Learning Architecture for the Classification of Cancerous Lymph Node Images." In *2020 Second*

International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 381-386. IEEE, 2020.

[7] Srinivasan, Sriram, Vinayakumar Ravi, V. Sowmya, Moez Krichen, Dhouha Ben Noureddine, Shashank Anivilla, and Soman Kp. "Deep convolutional neural network based image spam classification." In *2020 6th conference on data science and machine learning applications (CDMA)*, pp. 112-117. IEEE, 2020.

[8] Chaumont, Marc. "Deep learning in steganography and steganalysis." In *Digital Media Steganography*, pp. 321-349. Academic Press, 2020.

[9] Raksha, R., and P. Surekha. "A Cohesive Farm Monitoring and Wild Animal Warning Prototype System using IoT and Machine Learning." In *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 472-476. IEEE, 2020.

[10] Bajaj, Komal, Dushyant Kumar Singh, and Mohd Aquib Ansari. "Autoencoders based deep learner for image denoising." Procedia Computer Science 171 (2020): 1535-1541.

[11] Zhang, Jinsong, Yi Zhang, Lianfa Bai, and Jing Han. "Lossless-constraint denoising based auto-encoders." Signal Processing: Image Communication 63 (2018): 92-99.

[12] Akaash, B., and R. Aarthi. "An Analysis of Rainstreak Modeling as a Noise Parameter Using Deep Learning Techniques." In *Advances in Computing and Network Communications*, pp. 465-477. Springer, Singapore, 2021.

[13] Nishio, Mizuho, Chihiro Nagashima, Saori Hirabayashi, Akinori Ohnishi, Kaori Sasaki, Tomoyuki Sagawa, Masayuki Hamada, and Tatsuo Yamashita. "Convolutional auto-encoder for image denoising of ultra-low-dose CT." Heliyon 3, no. 8 (2017): e00393.

[14] Tian, Chunwei, Yong Xu, Lunke Fei, Junqian Wang, Jie Wen, and Nan Luo. "Enhanced CNN for image denoising." CAAI Transactions on Intelligence Technology 4, no. 1 (2019): 17-23.

[15] Haque, Kazi Nazmul, Mohammad Abu Yousuf, and Rajib Rana. "Image denoising and restoration with CNN-LSTM Encoder Decoder with Direct Attention." arXiv preprint arXiv:1801.05141 (2018).

[16] Peng, P., Jalali, S., & Yuan, X. (2020). Solving inverse problems via auto-encoders. *IEEE Journal on Selected Areas in Information Theory*, *1*(1), 312-323.

[17] Zhang, Kai, 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* 26, no. 7 (2017): 3142-3155.

[18] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In *European conference on computer vision*, pp. 740-755. Springer, Cham, 2014.

[19] Martin, David, Charless Fowlkes, Doron Tal, and Jitendra Malik. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 416-423. IEEE, 2001.

[20] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In *International conference on machine learning*, pp. 448-456. PMLR, 2015.

[21] Hore, A., & Ziou, D. (2010, August). Image quality metrics: PSNR vs. SSIM. In 2010 20th international conference on pattern recognition (pp. 2366-2369). IEEE.

[22] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.