Image Denoising by Deep Neural Networks

Ashish Jha
Department of Artificial Intelligence
Indian Institute of Technology, Hyderabad
Hyderabad, India
ai22mtech13002@iith.ac.in

Rahul Raghav
Department of Artificial Intelligence
Indian Institute of Technology, Hyderabad
Hyderabad, India
ai22mtech13007@iith.ac.in

Soumyanetra Pal
Department of Artificial Intelligence
Indian Institute of Technology, Hyderabad
Hyderabad, India
ai22mtech14005@iith.ac.in

Dhanush Pittala
Department of Biomedical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India
bm20btech11004@iith.ac.in

Abstract

Discriminative model learning patterns for image denoising have demanded huge attention due to their excellent denoising performance in recent years. In this project, we will be implementing a paper based on feed-forward denoising convolutional neural networks(DnCNNs) [7]. DnCNN is based on residual learning (RL) approach, wherein noise is predicted from the received noisy image, and not the ground truth image itself. This approach is seminal in the sense that it not only handles the Gaussian Noise effectively, but also the distortions due to Single Image Super resolution and artifacts due to JPEG compression.

1. Introduction

Noise occurs in images due to various reasons. These may be naturally occurring e.g. gaussian noise. It may be sensor induced e.g. speckle or it may be a result of processing operation e.g. quantization. Hence, Image denoising has been a fundamental research topic in computer vision and image processing applications. The goal of such endeavours has been to reconstruct a clean image from a noisy observation. Noise can be modeled as additive y = x + z, where x is a clean image and z is some type of noise such as the additive Gaussian white noise. We can also model noise as multiplicative y = x.z where z is some type of noise such as speckle. Task of image restoration can be achieved through the modeling of image priors or using discriminative learning approaches. Various methods have been put forward for modeling image priors that the DnCNN paper cites. Some of these include non-local self-similarity (NSS) models, sparse models, gradient models etc. Traditional methods such as BM3D [5], LSSC [4] , EPLL [21], and WNNM [13] rely on image prior modeling and require domain knowledge. However, finding an effective prior is still difficult. It is well-known that BM3D is a highly engineered Gaussian image denoising algorithm. It involves a block matching process, which is not suitable for parallel computation on GPUs.

The paper [7] lists two major drawbacks from which prior-based methods typically suffer. Firstly, these methods generally involve a complex optimization problem in the testing stage, making the denoising process time consuming. Secondly, this involves several manually chosen parameters and since we know that these models in general are non-convex, definitely denoising performance can be improved using CNNs. Recently, as architectures became more flexible, deep learning techniques have gained the ability to overcome these drawbacks [10]. It is known from the universal approximation theorem [6] that a fully connected neural network with a large number of neurons in its hidden layer has the ability to represent any function we wish to learn, provided our activation functions satisfy some mild assumptions. However, when dealing with highly structured modalities such as images or videos, using a convolutional neural network (CNN) is typically the default model of choice. CNNs are particularly suitable for processing images as they can easily extract the statistics of their input and make use of them to solve the problem of reconstructing the clean image. Finally, CNN-based architectures, if properly designed, can be shown to share similarities with analytical methods (e.g., optimization-based iterative methods or deconvolution steps), which suggests that CNN-based models can be powerful tools for solving inverse problems in imaging. For example, a connection between CNNs and multilayer convolutional sparse coding was established [15], offering a fresh view and, potentially, a better understanding of CNNs but also potentially other architectures (e.g., residual networks) and the common tricks currently employed such as batch normalization and dropouts.

The proposed denoising convolutional neural network is referred to as DnCNN in the paper. Rather than directly outputing the denoised image x, the model DnCNN is designed to predict the residual image z, i.e., the difference between the noisy observation and the latent clean image. The batch normalization technique is further introduced to stabilize and enhance the training performance of DnCNN.The use of residual blocks in the network plays a significant role in training very deep models. This method of residual learning has been proven empirically to be more effective []. The intuition can be further developed from the observation that the noisy image and the ground truth almost form the identity mapping F(y) = x, whereas the variance between input image and the modelled noise is much more pronounced, providing larger room for optimisation. The report can be summarized as follows:

1) A deep CNN for Gaussian denoising is proposed. The network estimates the noisy observation using the residual learning strategy. In contrast most deep neural network-based methods directly estimate the latent clean image.

2) Residual learning and batch normalization not only speed up the training but also boost the denoising performance. For Gaussian denoising with a certain noise level, DnCNN outperforms state-of-the-art models with respect to quantitative metrics as well as visual quality.

3) The DnCNN can be easily extended to handle general imaging problems such as superresolution, JPEG deblocking and blind gaussian denoising i.e. when applied to gaussian denoising with unknown noise level. A single trained DnCNN yields good results on these general tasks.

4) Although,the deep learning methods have achieved good results there is still room for reducing the computational burden and maintain the desired denoising effect. Towards this, the filter sizes can be optimized to produce better results.

2. Literature Review

Image Denoising has been one of the most visited areas in Image Processing and Computer Vision. Some noticeable works done in the past are listed below-

A. Image Denoising using Averaging-Based Techniques

In [14], Toni Buades, Yefei Lou, J. M. Morel, and Zhongwei Tang found an alternative to increasing aperture time while taking low-light images. They proposed taking images in a burst and averaging them to get a better quality image. In [1], A. Buades, B. Coll, and J. M. Morel introduced the concept of method noise and used the NL-means algorithm to denoise images. They have also demonstrated their performance as compared to conventional denoising techniques. These techniques have satisfying results but fail to generalise well in case of different types of noises.

B. Image Denoising using Deep Neural Networks

In [16], Viren Jain, and Sebastian Seung combined two ideas. They used convolutional networks as an image-processing architecture and synthesised more training data from specific noise models using unsupervised learning procedure. They got great results with very low computational complexity. In [3], Harold C. Burger, Christian J. Schuler, and Stefan Harmeling kept things very simple. They executed the denoising task with MLPs. In [18], Yunjin Chen and Thomas Pock suggested a trainable nonlinear reaction diffusion (TNRD) model and posed it as a feed-forward deep network by backpropagating a fixed number of gradient descent inference steps. Among the above deep neural networks based methods, MLP and TNRD based models can achieve comparable performance as BM3D. However, for MLP [3] and TNRD [18] based models, the models are trained specifically for some certain noise levels. But it remained unknown at the time of this paper, as to how to develop a CNN model for general image denoising task, not focussing on the specific noise distribution.

C. Image Denoising using Residual Learning and Batch Normalization

In [8], Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun solved a very important propblem of regular CNNs. In CNNs, the training error also starts to increase after a certain increment in the number of layers. This happens because of some issues, most prominent of them is vanishing gradients problem. This issues were solved by RNNs proposed in [8]. This also enhances the efficiency of the netwrok. In [11], Sergey Ioffe, Christian Szegedy proposed to alleviate the internal covariate shift by incorporating a Batch normalization step and a scale and shift step before the applying nonlinearity in each convolution layer. Batch normalization enables fast training, better performance, and low sensitivity to initialization. The paper has used this integration of residual learning and batch normalization for fast and stable training and better efficiency, which we have implemented in our project work.

3. Review of Recent works on Neural Network based Denoising

There have been advances in Image Denoising tasks, but the scope of improvement still exists. The motivation for most of the denoising approaches have been on the use of Non- Linear filters to preserve edge information, adaptive and non adaptive filters filters, to reduce artifacts in low frequency regions, and also on reducing the computation cost through sparse representations or dictionary learning methods.

However, the robustness of these models for common (or blind) denoising tasks is yet to be achieved. For blind denoising, a fast and flexible denoising CNN (FFDNet) [20] trained CNN model for blind denoising task. ADFNet [12] proposes Dynamic Convolution Block (DCB) and Multiscale dynamic convolution block (MDCB) and has achieved better results over DnCNN. Restormer [19] Restoration Transformer, uses transformer architecture and achieves state-of-the-art results on several image restoration tasks, motion deblurring, focus deblurring and image denoising.

4. Implementation of the Model DnCNN

A. Model Architecture from Paper

DnCNN utilizes VGG like architecture, however, unlike the VGG network, the number of channels in each layer are preserved. We have implemented the approach based on the use of residual learning and batch normalization (as presented in the paper) so as to improve the denoising performance and train faster by eliminating Internal Covariate Shift. The size of convolutional layers are set to 3*3 with no pooling layers. We retrieved patches of size 40*40 for reproducing the results.
We feed the noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{v}$ into the DnCNN, where (v) represents generated noise and train a Residual mapping $\mathbf{R}(\mathbf{y}) \approx \mathbf{v}$. The loss function used here is the averaged mean squared error between the desired residual images and the estimated ones from the noisy input.

$$l(\mathbf{\Theta}) = \frac{1}{2N} \sum_{i=1}^{N} ||R(\mathbf{y_i}; \mathbf{\Theta}) - (\mathbf{y_i} - \mathbf{x_i})||^2$$

The N noisy-clean training image pairs are represented by $(\mathbf{y_i}, \mathbf{x_i})_{i=1}^{N}$. So, for the first layer the paper has proposed implementation of 64 filters of size 3*3*c to generate 64 feature maps and uses ReLU for non-linearity. c represents the number of channels. From the 2nd layer to the 16th layer, it uses 64 filters of size 3*3*64, if 17 is the depth of the proposed network. The authors then add batch normalization between the convolution and ReLU to mitigate ICS. Last layer has c filters of size 3*3*64 to regenerate image of original scale (gray or three channel). Inputs have been Zero padded before convolution so that the size of the input image and the size of each feature map of the middle layers are the same. It has been observed that by padding this way, no boundary artifacts were found.

B. Implementation Details

We have implemented two CNN-based models, DnCNN-S and DnCNN-B, that share a similar base architecture. DnCNN-S is the model that is trained for known Noise while DnCNN-B is trained with images having widely varying(0-55) noise variance for blind denoising. Both models begin with an initial convolution layer, followed by a series of convolution and batch normalization layers, and conclude with an output convolution layer. The depth of the intermediate convolution layers differs between the two models, with DnCNN-S consisting of 17 layers and DnCNN-B consisting of 20 layers. In both models, the spatial dimension of the input is preserved throughout the architecture.

During the training of both models, a batch size of 128 is used for 50 epochs. In the case of DnCNN-S, we added three different levels of noise to the images, with sigmas of 15, 25, and 50. On the other hand, for DnCNN-B, we added a random unknown noise of sigma in the range of [0, 55].

C. Results from our Experiments.

Figures 1 - 5 represent results from our implementation of the author's code (Paper code is available https://github.com/cszn/DnCNN). We implemented the paper with training on 50 epochs. Results are very good for single Noise variance model with $\sigma = 15, 25$ (figure 1 and 2). However as the noise increases, the effect of model starts depreciating as observed from the figure 3 and 4.

We created our model and trained on subset of images. The model has same architecture as detailed in section 4(A) and 4(B). However, due to lack of computation resources, we have not applied various transformations and patches creation on the input data set. Model training converges with smooth loss vs epochs plot (figure 8). However, since the model is underfitting for the lack of training, results are likely to improve and be close to the paper's implementation or even better with the variant of original model. Results from our model is visualized in Figures 5-7. Our model is giving competitive performance in terms of Blind Noise Denoising on grayscale image with PSNR (34.00) and single variance noisy image on grayscale (PSNR 28.03). Results are to be improved on three channel images (figure 6) with more training using various data transformations.

5. Models created and Experiments

Apart from recreating the DnCNN models and reproducing the denoising results based on this, we created three models on top of DnCNN taking ideas from Deep Learning concepts and tested our implementation with respect to the denoising results produced by DnCNN architecture. The architectures implemented are :

(a) **DnCNN++ (with increased depth of feature maps)**.

In this experimental model, we have used a similar architecture as the original DnCNN. We used 21 convolution layers. Layers 1-5 has a depth of 64. We them increased it to 128 for the layers 6-15 followed by a set of layers with depth 64 from 16-20. Last convolution layer converts 64 layered featuremap to 3 or 1 layers noise representation based on the input type. We called this **Increased Depth DnCNN model**.

With the same architecture, with dilated features, another model is trained. We called that **Increased Depth DnCNN with Dialated filters model**

Each of these models are trained for known and blind noises. The output results of which are presented in Figure 9, 10, 11, 12 respectively.

(b) **Self-Attention based Denoiser (DnCNN with a self - attention layer)**.

In this experimental model, We have used the same architecture as original DnCNN of 17 layered, Conv + BN + ReLU, with the addition of a self-attention layer before the output layer.

This modification is based on the intuition that while training, DnCNN model is using standard convolution kernel of size 3, which is instrumental in exploiting spatial correlation in image. Although with backpropagation of loss with respect to noise, it may still be capturing pixel correlations. With the addition of a self-attention layer before output and backpropagating loss with respect to noise, the model learns to forget image feature correlations and it should try to capture noise variance ignoring spatial image structure. We experimented with two self-attention layers, one in middle of DnCNN and one before output. However, in the initial epochs the model does not perform good, probably indication the capture of image features.

The performance of Single layered self-attention model is competitive to DnCNN even with training on just 10 epochs, it provides validation PSNR more than 31. The architechure of this model is visualized in Fig 9. PSNR curve for Blind Denoising with 10 epochs (varying with different batches) is plotted in Fig 10. Plot of single denoising performance is depicted in Figure 11 (PSNR 35.56), which is reconstructed at PSNR more than 35. The effect of this self attention model is shown through various images (Figure 13), their noisy version (Figure 14) and reconstructed image (Figure 15). Pre-trained weights and training file is submitted with the project.

Results with these experiments are tabulated in Table 1.

6. Study of other latest CNN models

. Latest trends in image denoising have been on use of diffusion based models (Unet) and GAN based generative models, which even reconstruct missing patches from images. However, as part of our project, we studied models focussing specifically on denoising and not generation. We studied following models apart from DnCNN.

A. ADF Net [12].

This denoising model uses the idea of adaptive designing of convolution kernel which better preserves spatial context information as compared to previous Deep CNN models for denoising. This is done by using adaptive dynamic filtering network (ADF Net).

B. Denoising Autoencoder. [17]

Autoencoders have multiple convolutional and deconvolutional layers. Denoising Autoencoders learn mappings from noisy images to original clean images.The learned mapping can then be used to remove noise from new, unseen input signals. Variational Autoencoders use use a probabilistic approach to generate clean outputs.Generative Adversial Networks use a discriminative model to learn the noise distribution in the input data.

C. SwinIR [9]

SwinIR is a Swin Transformer based approach for Image Reconstruction. This method consists of three modules namely shallow feature extraction, deep feature extraction and high-quality image reconstruction. In the shallow feature extraction module a 3×3 convolutional layer is used to extract shallow features. These features mostly contain the low frequencies. In deep feature extraction module 'k' residual Swin Transformer blocks (RSTB) and a 3×3 convolutional layer are used. Each RSTB in turn consists of several Swin Transformer layers for local attention and cross-window interaction. In the reconstruction module, the high-quality image is reconstructed by aggregating shallow and deep features. For image denoising and JPEG compression tasks, a single convolution layer is used for reconstruction while for superresolution a sub-pixel convolution layer is used for upsampling. Further, the authors of this paper have concluded that SwinIR outperforms state-of-the-art methods on different tasks by up to 0.14 ~0.45dB.

D. Attention Guided CNN for Image Denoising [2].

The Attention guided CNN contains a sparse block which improves performance by using dilated convolutional layers to remove the noise. Other than the sparse block, this network contains a feature enhancement block, an attention block and a reconstruction block. Efficiency of the model is improved by integrating the feature enhancement block and the attention block. The reconstruction block constructs the clean image by using the given noisy image and the noisy mapping obtained from the model.

7. Comparative Study

Following is the comaparative study of all the models with respect to their output PSNRs during testing

Model Type	**PSNR**
DnCNN-S	26.09
DnCNN-B	34.00
Increased Depth DnCNN-S	30.03
Increased Depth DnCNN-B	29.91
Increased Depth Dialated DnCNN-S	27.44
Increased Depth Dialated DnCNN-B	26.02
Self-Attention DnCNN	32.28

Table 1. PSNR values for the models.

8. Conclusion

In this course project, we studied non local means based traditional analytical methods of Image Denoising, BM3D, TNRD and then one CNN based denoising model DnCNN. DnCNN is a lightweight CNN model which still gives good denoising performance, if we ignore diffusion or transformer based denoising cum Super-Resolution problem solving models.

We went about implementing DnCNN model and trained it for different variants (Blind denoising with varying noise levels and Single Noise variant with Noise variance 15, 25). We modified the original model with three variants and analysed their performances on denoising task. All three variants of our model, i.e. with increased depth, with Dilated Kernel and with self-attention gives competitive performance as compared to DnCNN. It will be interesting to see the performance of self-attention model once it is trained for 50-60 epochs, which is the number on which DnCNN model with trained with.

9. Further scope of work.

In the future, we plan to study techniques like Self-supervised learning, Diffusion based learning. We plan to focus on reducing the computation cost while achieving better denoising performance. We also plan to employ sparsity driven techniques such as Dictionary Learning, Sparse networks,Wavelet Inspired Networks. We can try combining learning based techniques and wavelet based techniques to achieve better results.

References

[1] J. M. Morel A. Buades, B. Coll. Image denoising by non-local averaging. (1415332), 2005. 2

[2] Zuoyong Li Wangmeng Zuo Lunke Fei Hong Liu Chunwei Tian, Yong Xu. Attention-guided cnn for image denoising, 2020. 5

[3] Stefan Harmeling Harold C. Burger, Christian J. Schuler. Image denoising: Can plain neural networks compete with bm3d? *in IEEE Conferenceon Computer Vision and Pattern Recognition*, page 2392–2399, 2012. 2

[4] J. Ponce G. Sapiro J. Mairal, F. Bach and A. Zisserman. Non-local sparse models for image restoration. *IEEE International Conference on Computer Vision*, page 2272–2279, 2009. 1

[5] V. Katkovnik K. Dabov, A. Foi and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1

[6] M. Stinchcombe K. Hornik and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw*, 3(5):551–560, 1990. 1

[7] Yunjin Chen Deyu Meng Lei Zhang Kai Zhang, Wangmeng Zuo. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *in IEEE transactions on Image Processing*, pages 3142–3155, 2017. 1

[8] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *in IEEE Conferenceon Computer Vision and Pattern Recognition*, (7780459):770–778, 2016. 2

[9] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *arXiv preprint arXiv:2108.10257*, 2021. 4

[10] Iliadis M. Molina R. Katsaggelos A. K. Lucas, A. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018. 1

[11] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covari-

ate shift. *in International Conference on Machine Learning*, pages 448–456, 2015. 2

[12] Hao Shen, Zhong-Qiu Zhao, and Wandi Zhang. Adaptive dynamic filtering network for image denoising. 2023. 3, 4

[13] Wangmeng Zuo Shuhang Gu, Lei Zhang and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. *IEEE International Conference on Computer Vision and Pattern Recognition*, page 2862–2869, 2014. 1

[14] J. M. Morel Zhongwei Tang Toni Buades, Yefei Lou. A note on multi-image denoising. (5278408), 2009. 2

[15] Y. Romano V. Papyan and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *arXiv preprint,arXiv:1607.08194*, 2016. 2

[16] Sebastian Seung Viren Jain. Natural image denoising with convolutional networks. 2008. 2

[17] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders, 2023. 4

[18] T.Pock Y.Chenand. Trainable non linear reaction diffusion: Aflexible framework for fast and effective image restoration. *in IEEE transactions on Pattern Analysis and Machine Intelligence*, (2596743):1256–1272, 2016. 2

[19] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 3

[20] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Transactions on Image Processing*, 2018. 3

[21] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. *IEEE International Conference on Computer Vision*, page 479–486, 2011. 1

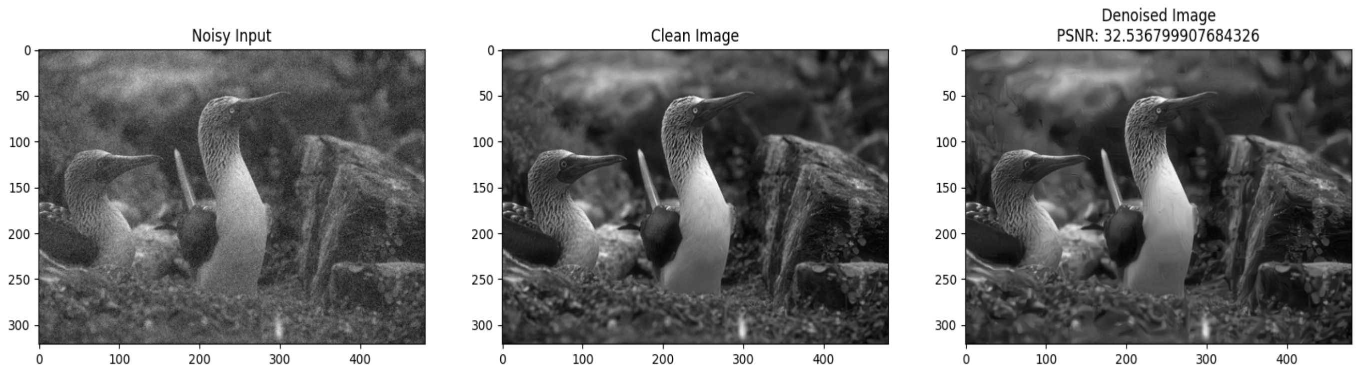

Figure 1. DnCNN output for Gassian Noise with sigma=15

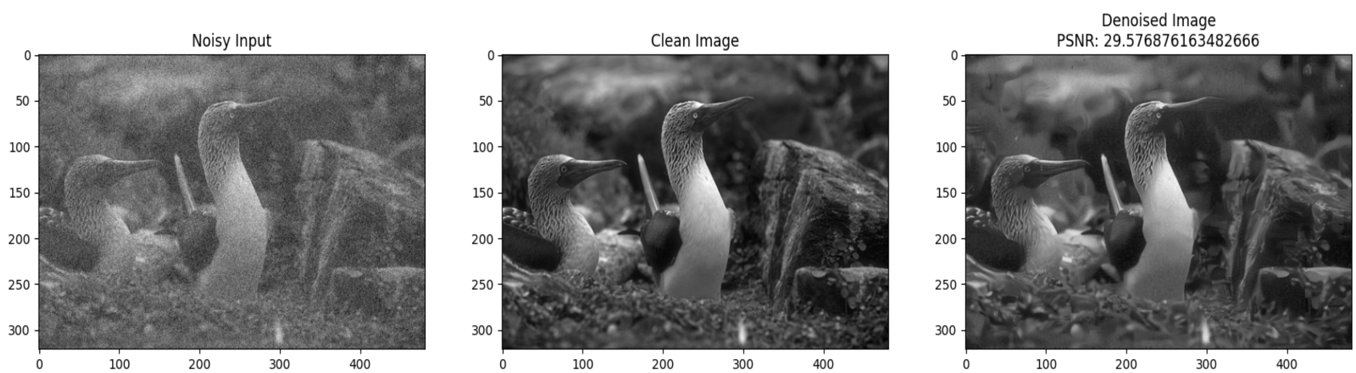

Figure 2. DnCNN output for Gassian Noise with sigma=25

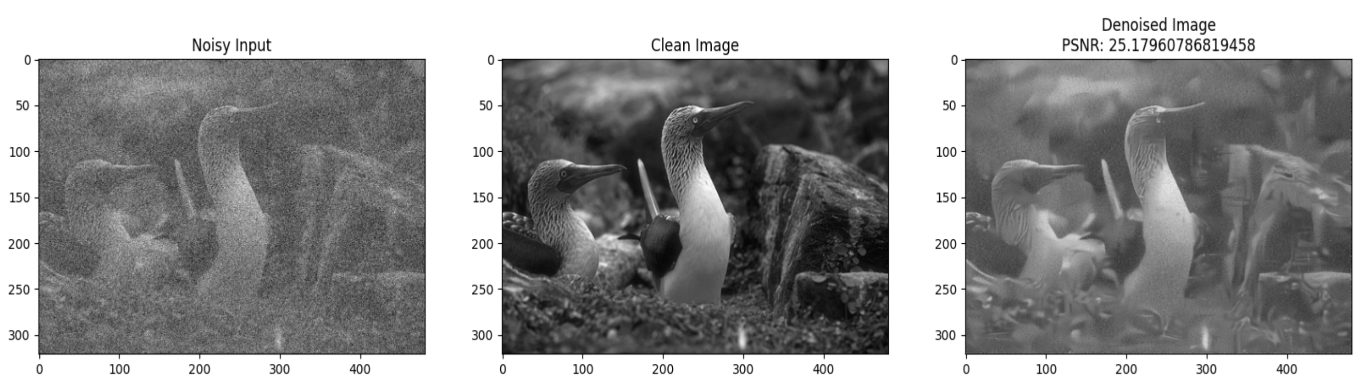

Figure 3. DnCNN output for Gassian Noise with sigma=50

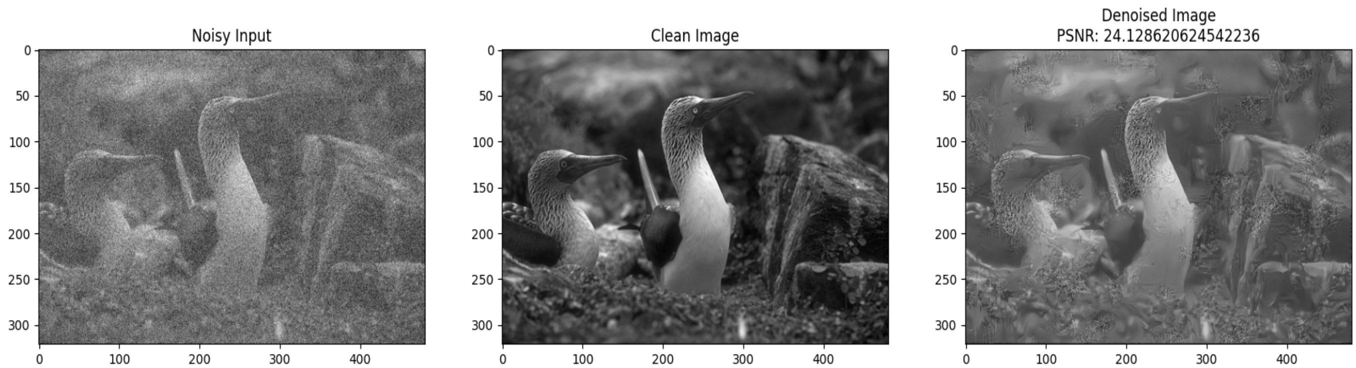

Figure 4. DnCNN output for Blind Noise

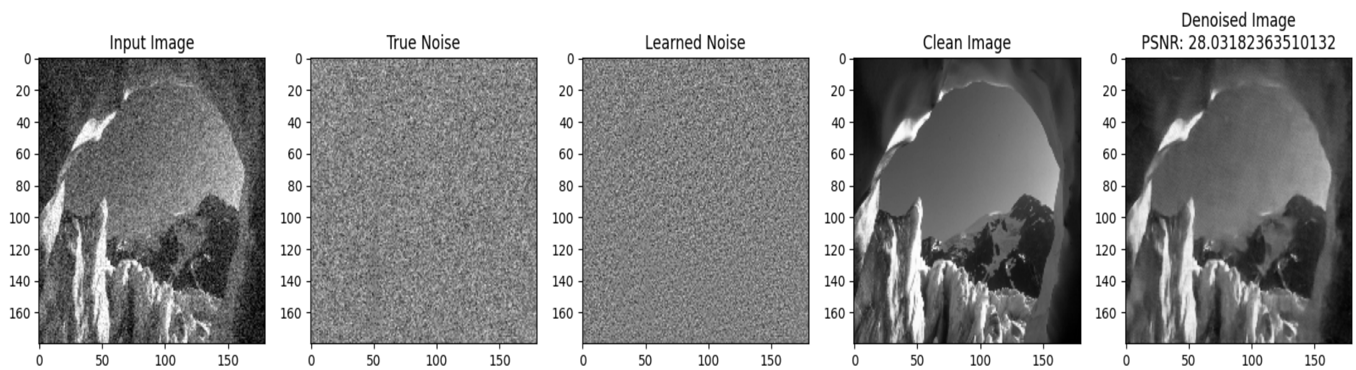

Figure 5. DnCNN-S output of Our Implementation for Grayscale Images

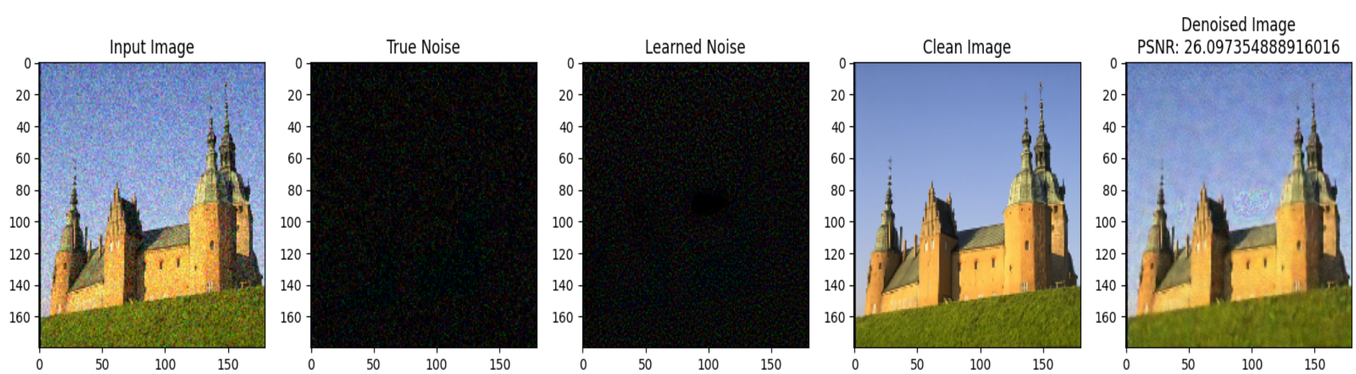

Figure 6. DnCNN-S output of Our Implementation for Colour Images

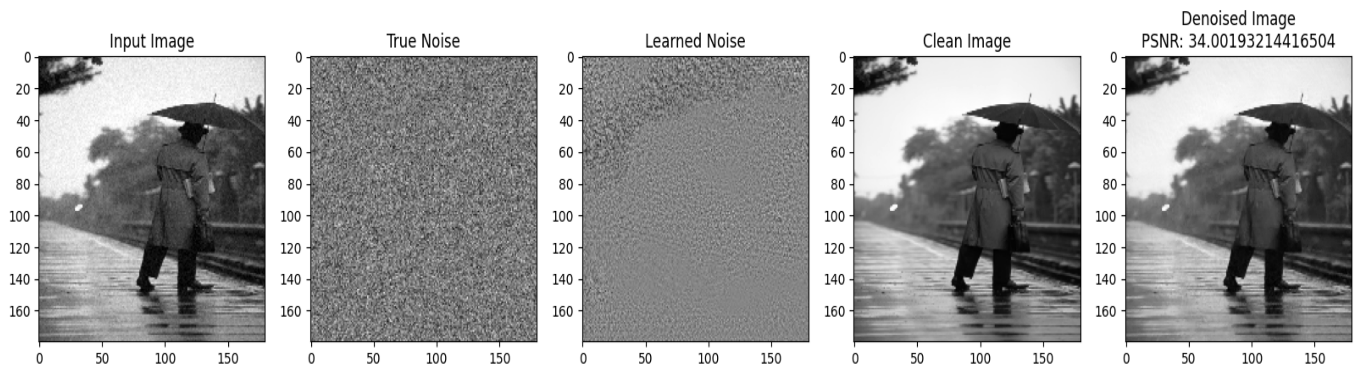

Figure 7. DnCNN-B output of Our Implementation with Blind Noise

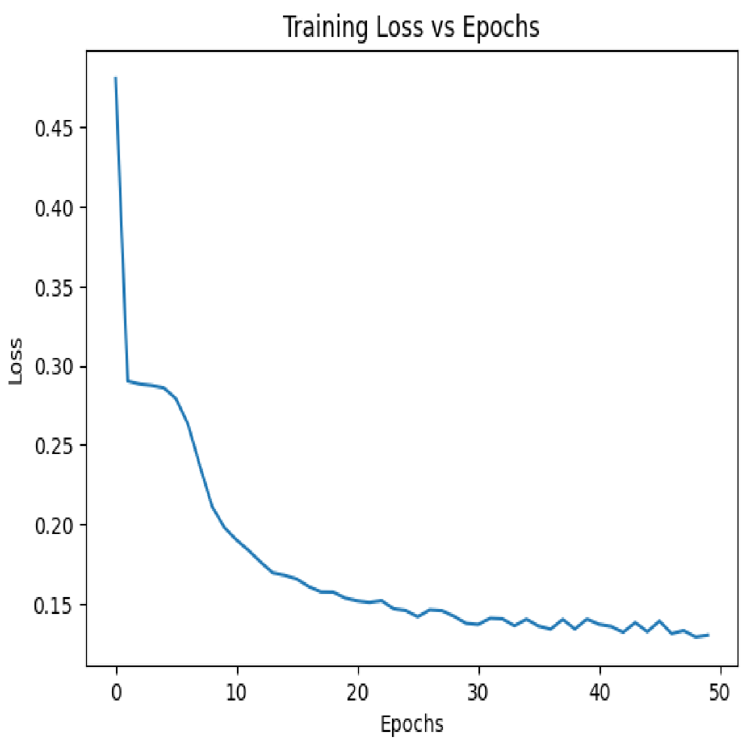

Figure 8. DnCNN loss vs epochs plot

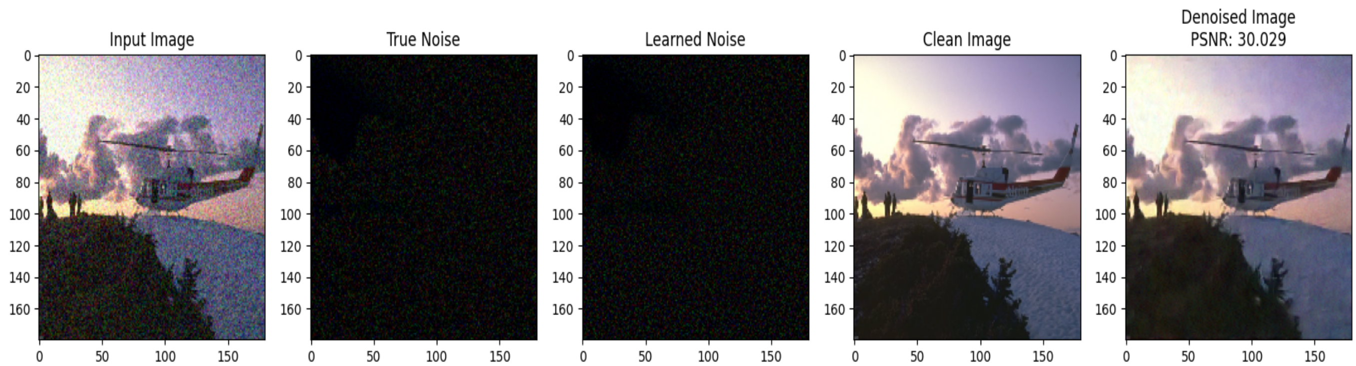

Figure 9. Increased Depth DnCNN for Gaussian noise (sigma=25)

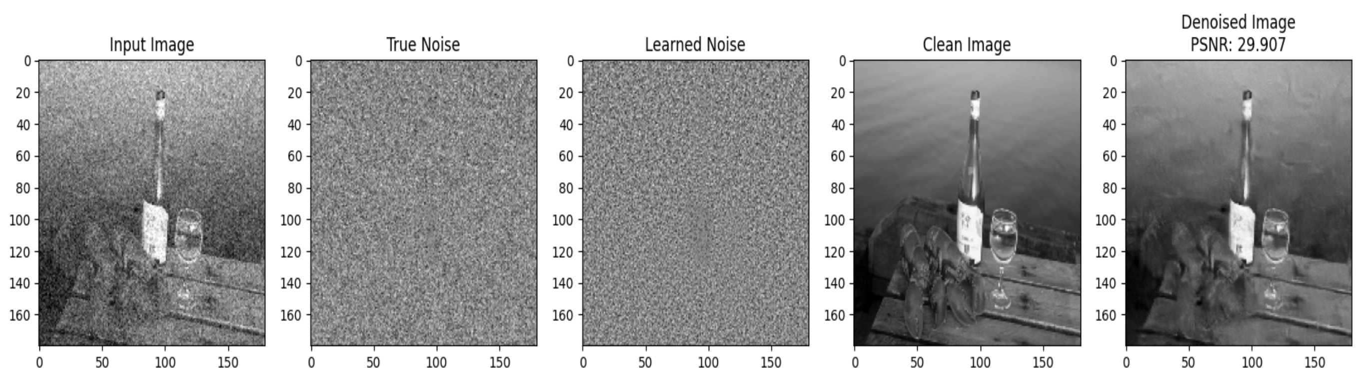

Figure 10. Increased Depth DnCNN for blind noise

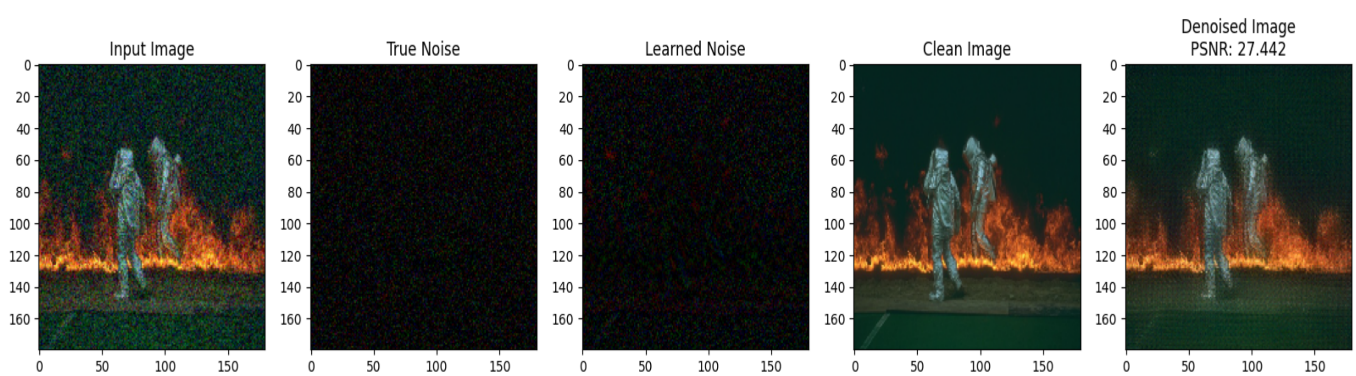

Figure 11. Increased Depth DnCNN with dialataed filters for Gaussian noise (sigma=25)

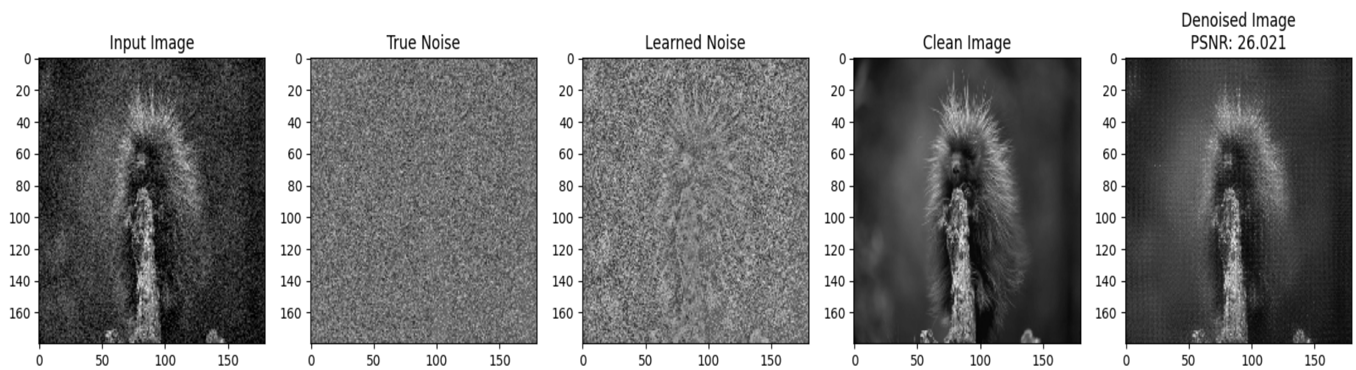

Figure 12. Increased Depth DnCNN with dialataed filters for Blind Noise

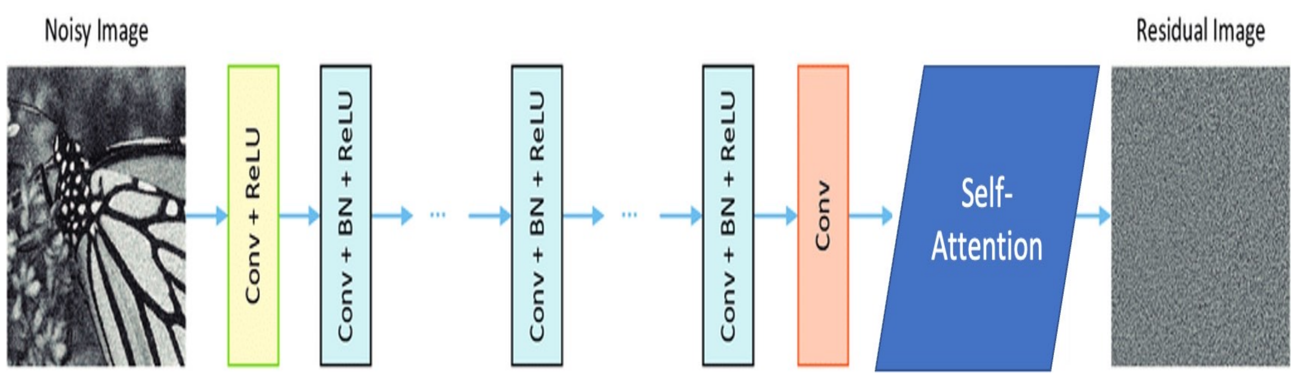

Figure 13. DnCNN with Self Attention - Architecture

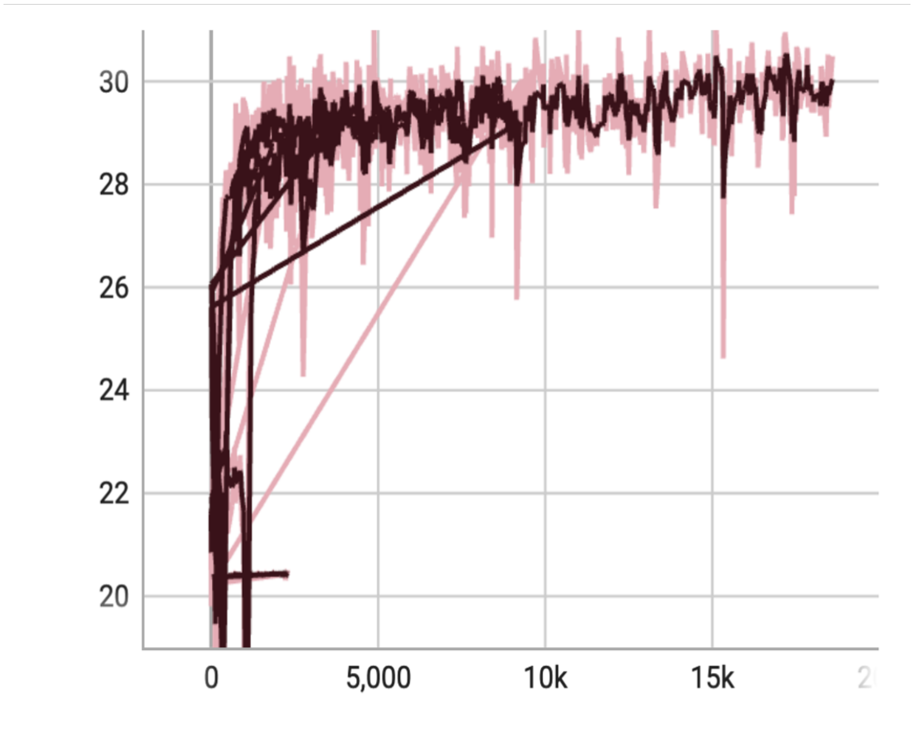

Figure 14. Attention DnCNN:Training PSNR plot (at 50 batch interval)

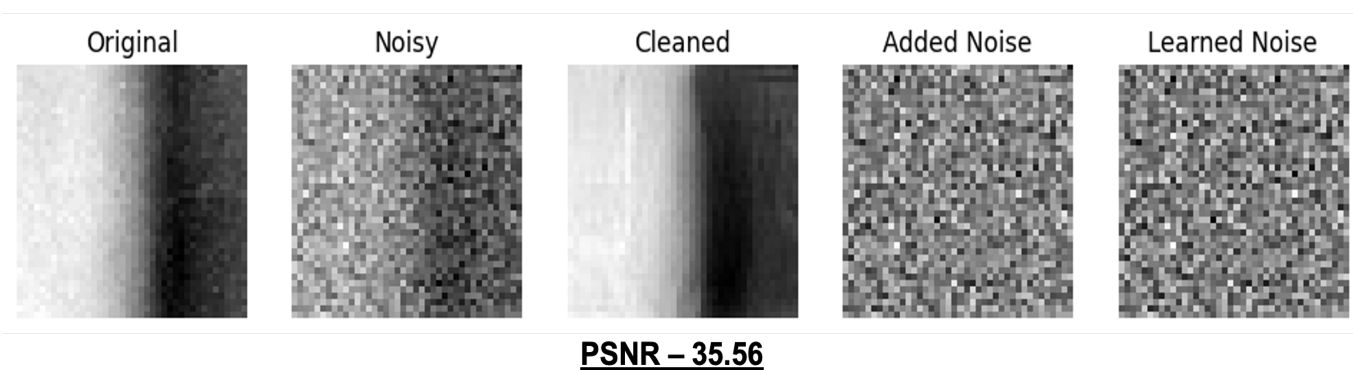

Figure 15. Effect on Validation Image Self-Attention(S = 25)

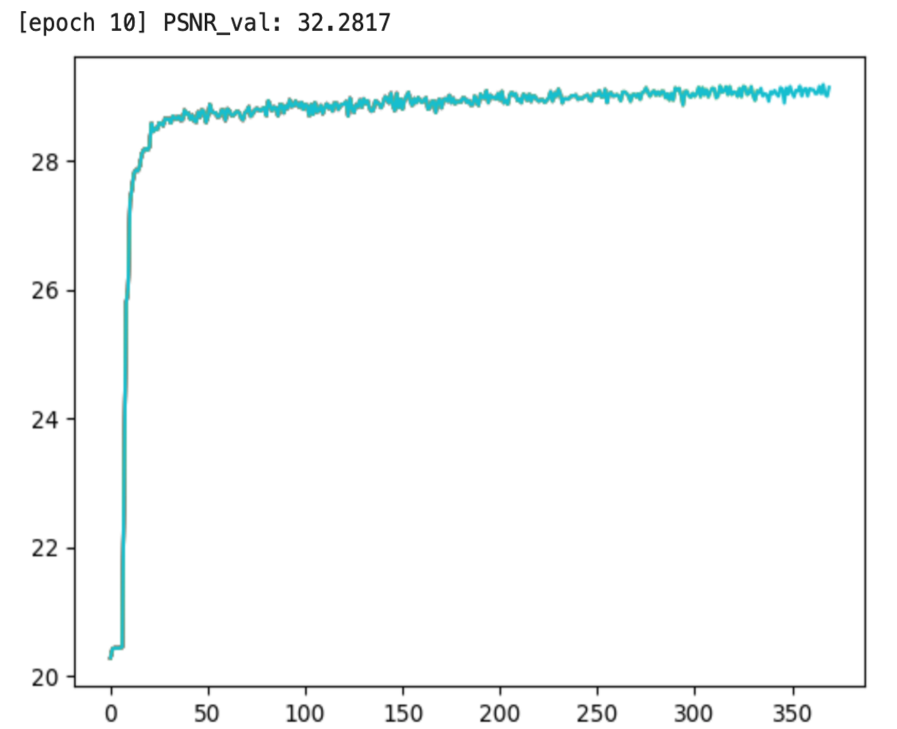

Figure 16. PSNR On Validation Data Self-Attention(S = 25), at 50 Batch Intervals

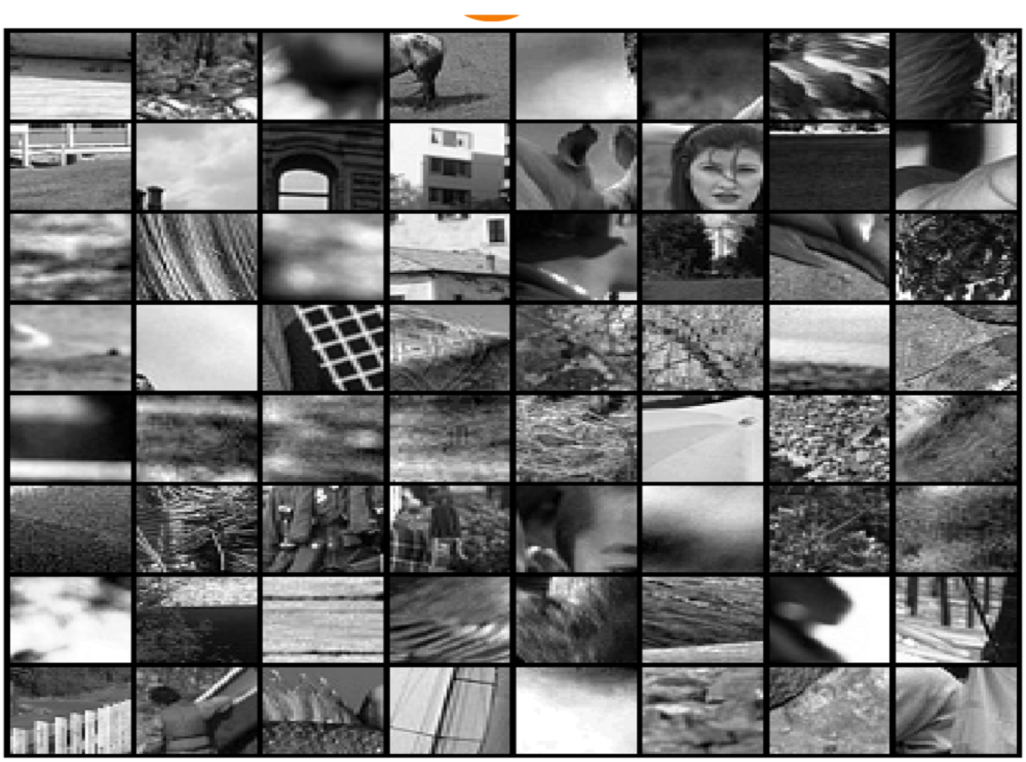

Figure 17. Original Images

Figure 18. Noisy Images

Figure 19. Reconstructed with Self-Attention Model