Enhancing Image Denoising Through Real Image Denoising with Feature Attention

LIENG HONGKY(202388530)
정보융합공학과 AI전공

Abstract

Image denoising is crucial for enhancing visual data quality across applications like medical imaging and surveillance. Classical methods face challenges in handling diverse noise types and demanding computational resources. Deep learning approaches show promise but often sacrifice details and require substantial computational power for very deep networks. This paper introduces the adoption of Real Image Denoising with Feature Attention (RIDNet)[4] to address these issues. This method efficiently preserves intricate details while reducing computational overhead. Through experiments, we validate its effectiveness, contributing to improved image denoising across various noise patterns. The code:

-For preprocessing and training model is available at https://github.com/lienghongky/RIDNet_Denoisor.git

-For deploying model on web based tools is available at https://github.com/lienghongky/PicyShake

1. Introduction:

Classical image denoising methods often require substantial computational resources and struggle to address various noise types. Deep learning-based approaches, while promising better results, often compromise image details as networks deepen, accompanied by escalating computational requirements to train very deep networks.

The Real Image Denoising with Feature Attention (RIDNet)[4] method has demonstrated two key advantages. Firstly, the incorporation of feature attention preserves intricate details during denoising. Secondly, when compared to other transformer-based or diffusion-based methods, this approach significantly reduces computational overhead, leading to faster training and inferencing times. However, RIDNet[4] conducted experiments on a limited number and types of real image datasets.

In response, this paper leverages the unique combination and architecture of RIDNet[4] by incorporating larger datasets, a variety of noise patterns, and adjusted sensitivity in the combined loss function (L1, L2). This paper reimplementing the architecture of original RIDNet[4] using **Tensorflow** instead of **Pytorch** for better training, Flexibility and Robustness. This approach proves to be an efficient and effective solution for image denoising. Through comprehensive experiments, we validate its effectiveness in achieving commendable denoising results across different noise patterns while maintaining computational efficiency. This contribution enhances the practicality and performance of image denoising techniques.

2. Related Works:

Image denoising techniques have evolved through classical methods, deep learning approaches, and recent advancements.

Figure 1. Show how the model performs on diverse types of noisy images.

2.1 Classical Methods:

Early methods like filtering and adaptive smoothing were effective but struggled with diverse noise types and high computational costs ([1]; [2]; [3]).

2.2 Real Image Denoising with Feature Attention:

RIDNet is a real image denoising method introduced by Anwar [4]. Its methodology hinges on two key components: feature attention and a carefully designed architecture. Feature attention enables the model to selectively emphasize important image features during the denoising process, also Long skip connection(LSC) and short skip connection(SSC), provide multiple paths for information flow and to leverage low-level features that capture fundamental characteristics of the input image. These low-level features can represent basic patterns and textures that are essential for accurate reconstruction.

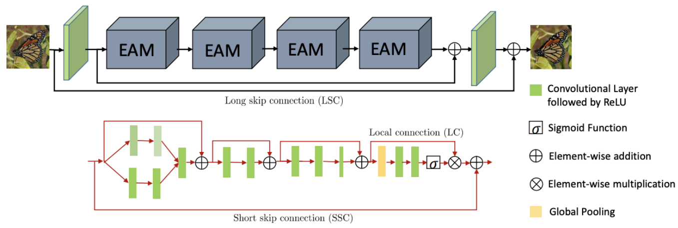


Figure 2. The architecture of the proposed network. Different green colors of the conv layers denote different dilations while the smaller size of the conv layer means the kernel is 1 × 1. The second row shows the architecture of each EAM. ([Anwar, ICCV-Oral, 2019])

3. Methods:

3.1 Data Augmentation

In this experiment, the imgaug[5] image augmentation library was employed to augment the dataset. Imgaug facilitates the application of a variety of diverse synthetic noise patterns to the images during the augmentation process. This stands in contrast to RIDNet[4], which is limited to real noise images and may have a constrained set of noise types and patterns. This approach broadens the dataset's representation of real-world scenarios, enhancing the model's ability to generalize and effectively denoise images with various noise characteristics.

3.1 Combination of L1 and L2 loss

In our denoising methodology, we employed a combination of L1 and L2 loss functions. This dual-loss approach involves utilizing both the L1 loss, which emphasizes preserving intricate **details** and **sharpness** in the denoised images, and the L2 loss, which promotes overall **smoothness**. The combined loss function is represented as follows:

$$Combined\ Loss\ = \lambda_1 * L1\ +\ \lambda_2 * L2$$

λ_1, λ_2 are weights that determine the influence of each component. This combination aims to strike a balance between preserving fine details and achieving a globally smooth denoised output, contributing to the overall efficiency and effectiveness of the denoising process.

4. Experiment:

All of the experiments were conducted within a Kubernetes pod, providing a containerized and scalable environment for all the required dependency to train the model with the same configuration as following:

CPU	: **Intel(R) Xeon(R)**
GPUs	: **NVIDIA A100-SXM4-40GB**
Framework	:**TensorFlow version: 2.14.0-gpu**
Duration	: **20 epochs.**
Optimizer	: **Adam optimizer.**
Learning rate	: **0.001 (1e-03).**

4.1 Experiment I And Results :

4.1.1 Dataset:

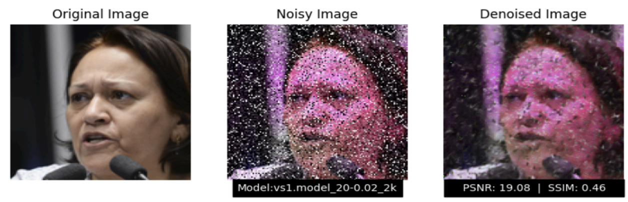
We began with a set of 2000 clean images from SIDD[6] and noise pattern applied image:

Noise Pattern: the following noise is applied sequentially in this order

1. *BlendAlphaFrequencyNoise*: Max pixel 32 and upscale by "linear"
2. *AdditiveGaussianNoise*: **10%**
3. *SaltNoise*: 1%

Loss: Mean Squared Error (MSE)

Training loss : 0.0010
Validation loss : 0.0187
Denoising capability : **PSNR: 19.08, SSIM: 0.46**



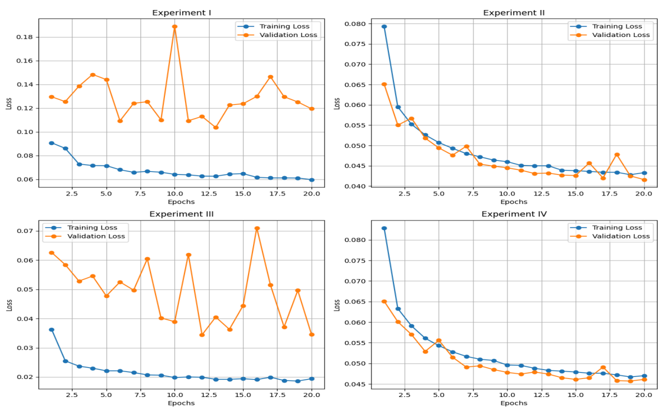


Figure 3. Show train and validation loss over epoch in all 4 experiments

4.2 Experiment II And Results:

4.2.1 Dataset:

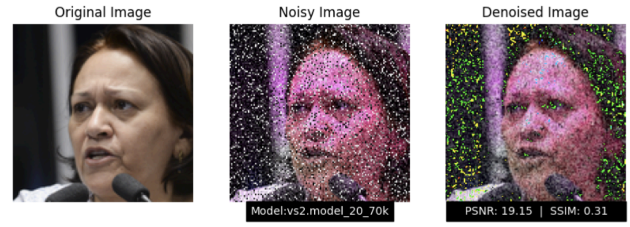
The 2000 clean images from the previous dataset, combined with an additional 68,000 clean images sourced from the FFHQ dataset [7], amount to a total of 70,000 clean images used in Experiment II.

Noise Pattern: randomly select one noise type and applied to an image

1. *AdditiveGaussianNoise*: 30%
2. *SaltNoise*: 10%
3. *SaltAndPaper*: 10%
4. *PaperNoise*: 10%
5. *BlendAlphaFrequencyNoise*: Max pixel 128 and upscale by "linear"
6. *JpegCompression*(compression=(70, 99))
7. *AdditivePoissonNoise*(40)

Loss: Mean Squared Error (MSE)

Training loss : 0.0428
Validation loss : 0.0425
Denoising capability : **PSNR: 19.15, SSIM: 0.31**



4.3 Experiment III And Results:

4.3.1 Dataset:

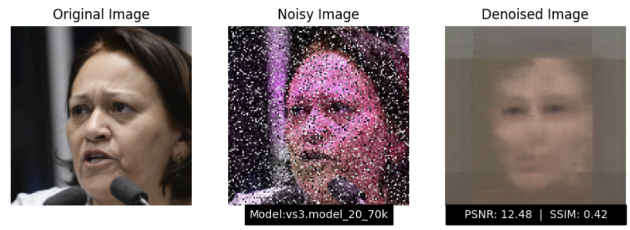
The 70,000 clean images from the previous dataset, are used in Experiment III.

Noise Pattern: the following noise is applied sequentially in this order

1. *AdditiveGaussianNoise*: 30%
2. *SaltNoise*: 10%
3. *SaltAndPaper*: 10%
4. *PaperNoise*: 10%
5. *BlendAlphaFrequencyNoise*: Max pixel 128 and upscale by "linear"
6. *JpegCompression*(compression=(70, 99))
7. *AdditivePoissonNoise*(40)

Combine Loss equal sensitivity: $L1\ +\ L2$

Training loss : 0.0194
Validation loss : 0.0346
Denoising capability : **PSNR: 12.48, SSIM: 0.42**



4.4 Experiment IV And Results:

4.4.1 Dataset:

The 70000 clean images from the previous dataset, combined with an additional 3,000 clean images sourced from the PPCD dataset [8], amount to a total of 100,000 clean images used in Experiment IV.

Noise Pattern: randomly select one sequence of noises out of two main sequences(A or B) and apply them in order of the selected sequence.

Sequence A

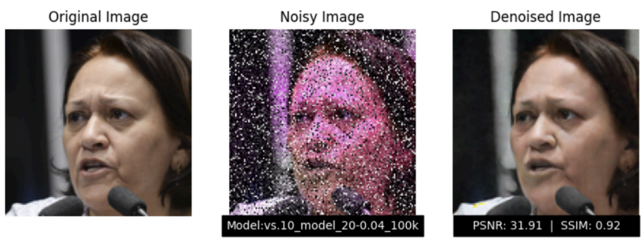
1. *BlendAlphaFrequencyNoise***:** Max pixel 32 and upscale by "linear"
2. *AdditiveGaussianNoise***: 10%**
3. *SaltAndPaper***: 10%**

Sequence B

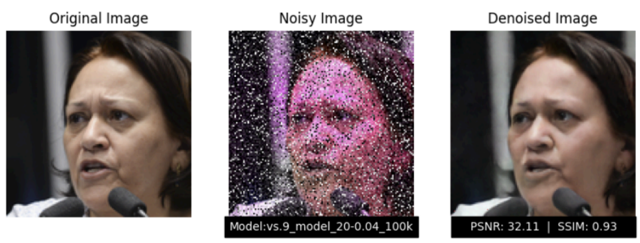
1. *JpegCompression*(compression=(70, 99))
2. *Randomly Select one from*
   a. *AdditiveGaussianNoise*: 10%(RGB)
   b. *AdditivePoissonNoise*: lam = 5
3. *AdditiveLaplaceNoise*: 10 %
4. Randomly *Select one from*
   a. *SaltNoise*: 10%
   b. *SaltAndPaper*: 10%
   c. *PaperNoise*: 10%
5. *BlendAlphaFrequencyNoise*: Max pixel 32 and upscale by "linear"

Combine Loss: $= \lambda_1 * L1 + \lambda_2 * L2$

Total Loss = (1.0*L1 + 0.8*L2)(1)
Training loss : 0.0470
Validation loss : 0.0461
Denoising capability : **PSNR: 31.91, SSIM: 0.92**



Total Loss = (0.7*L1 + 1.0* L2) (2)
Training loss : 0.0433
Validation loss : 0.0416
Denoising capability : **PSNR: 32.11, SSIM: 0.93**



5. Discussion:

This research endeavors to enhance the performance of denoising models through a series of carefully designed experiments. The key findings and conclusions drawn from these experiments are summarized as follows:

5.1 Minimal Noises Augmentation (Experiment I):

- Combination of Minimal noise types contribute to simplify network training but performance is still not good.
- Limited dataset leads to overfitting, impacting model adaptability.
- Emphasizing both augmentation diversity and dataset size for balanced denoising are needed.

5.2 Diverse Balancing Noise Types (Experiment II):

- Resolved overfitting, but noises in each channel appear as if the model did not learn the noise in the color channel.
- Model's selective performance on one type of noise instead of all the noise type.
- Combination of noise types for balanced denoising could contribute to model performance.

5.3 Excessive Augmentation (Experiment III):

- Excessive augmentation leads to increase in complexity and fails to generalize.
- Lack of adaptability and overfitting issues and failed to generalize the noise types.
- Indicate that the image augmentation should not be too much or too less.

5.4 Loss Function Sensitivity (Experiment IV):

- Demonstrates the effectiveness of the combination of noise types and diverse balance of noise sequences.
- Sensitivity adjustments in L1 and L2 result in nuanced performance variations. There are improvements when L2 is more sensitive than L1 as shown in Experiment IV.
- The experiment suggests that by adjusting the sensitivity of L1 and L2 could further fine tune the model performance.

6. Conclusions:

Method	PSNR	SSIM
RIDNet	39.23	0.9526
Our Version Experiment IV(2)	32.11	0.93
Our Version Experiment IV(1)	31.11	0.93
Our Version Experiment III	12.48	0.42
Our Version Experiment II	19.15	0.31
Our Version Experiment I	19.8	0.46

Figure 4. The comparison of PSNR and SSIM between ours and RIDNet

In conclusion, while our approach may not have surpassed the performance metrics of the original RIDNet[4], it has demonstrated notable strengths in addressing diverse noise types, including unconventional ones like rain and snow.

The performance, as indicated by both PSNR and SSIM metrics, may be slightly lower than the original version, with PSNR at **32.11** and SSIM at **0.93** compared to RIDNet[4]'s **39.23** and **0.9526**, respectively.

7. Limitation:

7.1 Border Artifacts:

Processing images in smaller patches may result in noticeable artifacts at the patch borders.

7.2 Challenges in Reconstruction:

Reconstructing the final image from processed patches poses challenges in maintaining visual coherence, especially in areas with complex textures.

Fixed patch size, like 256x256, may not align optimally with the structure of larger, real-world images, contributing to the line effect.

8. Literature Cited:

1. P. Saint-Marc, J.-S. Chen, and G. Medioni, Adaptive smoothing: A general tool for early vision, IEEE Transactions on Pattern Analysis & Machine Intelligence, 13 (1991), pp. 514–529
2. P. Huber, Robust statistics. new-york: John Wiley and sons, HuberRobust statistics 1981, (1981).
3. S. Roth and M. J. Black, Fields of experts: a framework for learning image priors, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2 (2005), pp. 860–867 vol. 2
4. Anwar, Saeed and Barnes, Nick, Real Image Denoising with Feature Attention: IEEE International Conference on Computer Vision (ICCV-Oral), 2019
5. Alexander Jung, imgaug,. https://imgaug.readthedocs.io/en/latest/index.html#
6. NVLab, FFHQ, Image dataset of human faces, https://github.com/NVlabs/ffhq-dataset
7. Abdelrahman Abdelhamed, Lin S., Brown M. S, SIDD, Smartphone Image Denoising Dataset https://www.eecs.yorku.ca/~kamel/sidd/
8. Siddesh Sambasivam, PPCD, Painting Vs Photograph Classification Dataset, https://www.kaggle.com/datasets/iiplutocrat45ii/painting-vs-photograph-classification-dataset/data

9. Appendices:

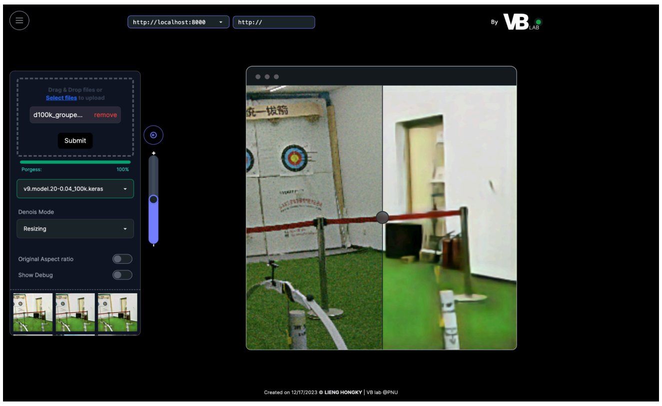


Figure 5. Web based Image denoise, using trained model.