Low Cost and Complexity Image Denoising Methods

Aryaman Trivedi - 2021022, Ieshaan Awasthy - 2021054

Problem Statement

Given a noisy image x',$x' = x + \Delta m$ where x is a clean image and Δ is a noise distribution, create a model f that $x = f(x')$. Further, to create f to achieve efficient and accurate denoising for real-world images utilizing low complexity models and maximizing dataset efficiency.

Background

Image denoising is a classical computer vision problem with various real-world applications, from photography to medicine to graphics design. Effective denoising is especially desirable in domains where it may be costly to capture clean data.

Challenges in Denoising. Denoising is challenging due to several factors:

- **Complexity of Denoising:** Denoising requires the model to ignore high-frequency details (noise) and maintain large-scale patterns while preserving small-scale characteristics such as edges and corners.

- **Unknown Noise Distribution:** Variation in noise patterns between images, wherein the noise distribution is unknown.

- **Cost of Capturing Clean Data:** The cost of capturing clean data for target labels can be significant; for example, 3D rendering a 'clean' sample may take thousands of compute hours.

Various models have been proposed, especially in recent years, and have achieved good results at the task. However, most remain computationally intensive and require large datasets to train. We intend to tackle both issues by analyzing the exceptions to this rule - Noise2Noise and NAFNet.

Selected Papers

Simple Baselines for Image Restoration. The authors proposed a then state-of-the-art model: NAFNet(Nonlinear Activation Free Network) for denoising and deblurring images. The authors carried out an extensive ablation analysis of various state-of-the-art methods, and utilized their core components to create a new, lightweight, efficient, and effective model which does not require any nonlinear activation functions. This model serves as the base of our project.

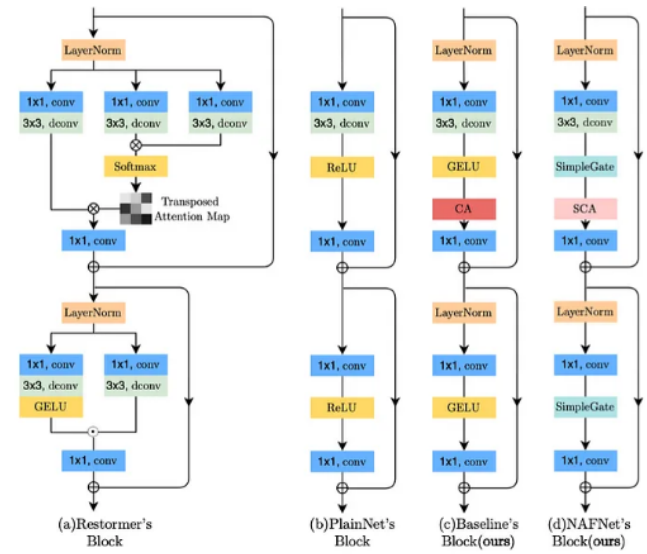

Fig. 3: Intra-block structure comparison. ⊗:matrix multiplication, ⊙/⊕:element-wise multiplication/addition. dconv: Depthwise convolution. Nonlinear activation functions are represented by yellow boxes. (a) Restormer's block[39], some details are omitted for simplicity, e.g. reshaping the feature maps. (b) PlainNet's block, which contains the most common components. (c) Our proposed baseline. Compares to (b), Channel Attention (CA) and LayerNorm are adopted. Besides, ReLU is replaced by GELU. (d) Our proposed Nonlinear Activation Free Network's block. It replaces CA/GELU with Simplified Channel Attention(SCA) and SimpleGate respectively. The details of these components are shown in Fig 4

Fig. 1. NAFNet Model

Noise2Noise: Learning Image Restoration without Clean Data. A simple statistical concept underlies Noise2Noise: the result of L2 Minimization remains the same even if the variables involved are replaced with others with the same expected values. This allows us to replace the clean targets with their noisy representations, which will, given infinite samples, yield the same result. This method yields good results even with limited or no clean samples. In its essence, this is an augmentation method.

Approach Description

Preprocessing

1. While SIDD contains noisy and clean pairs, BSD has only clean images. These are loaded and normalised (division by 255.0).

2. Noisy samples are creating by introducing random Gaussian noise centered at $\mu = 0$ with $\sigma = 0.05$.

3. Both datasets are divided into Train, Validation, and Test subsets with 60/20/20 distribution.

4. Images are resized to a standard (512, 512) size.

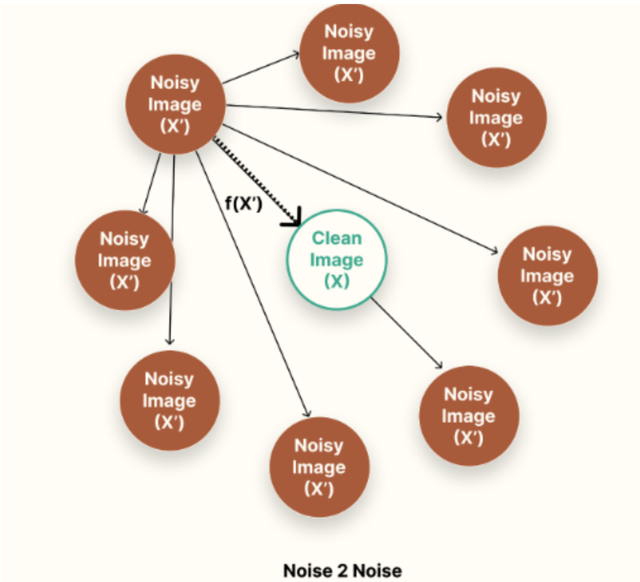

Fig. 2. Noise 2 Noise

5. **Noise to Noise** augmentation is implemented as follows:
   - num_create noisy samples are generated from each clean image.
   - All pairwise permutations of these samples, $(num\_create * (num\_create - 1))$ in number are saved. Finally, the num_create noisy samples are paired with the ground truth image.
   - Thus, the dataset size is multiplied $num\_create * num\_create$ times.

Training The NAFNet architecture is trained upon with the following hyperparameters using the Noise2Noise dataset:

- Inital Learning Rate = 0.009 with step reduction of 0.5 every 10 epochs.
- Batch size = 16
- Epochs = 200
- num_create = 2

We hypothesise that Noise 2 Noise augmentation should increase the performance of NAFNet modestly. The training takes roughly $3\frac{1}{2}$ hours to complete. If we make the assumption that the increase in time is linear with dataset size, training without Noise 2 Noise augmentation would take roughly an hour - we pay a heavy penalty in training time.

Datasets and Evaluation Metrics

Datasets.

SIDD (Smartphone Image Denoising Dataset). SIDD is an image-denoising dataset comprised of 30,000 noisy photos taken with five sample smartphone cameras from ten scenarios in various lighting situations. Ground truth images accompany the noisy photos. The dataset includes real-world noisy images, simulating the typical noise encountered in smartphone photography. Noise sources include sensor noise, compression artifacts, and other imperfections. Each noisy image is accompanied by its corresponding ground truth (clean) image. Ground truth images represent the ideal, noise-free version of the scene. This was originally used to test the performance of NAFNet.

BSD (Berkeley Segmentation Dataset). The BSD dataset was designed for image segmentation tasks. The photographs in the dataset range widely, from shots of natural scenery to images of specific objects like people, food, and plants. Targets were corrupted to be used as inputs for the model. This allows researchers to evaluate the ability of segmentation algorithms to recover clean boundaries from noisy or degraded inputs. This dataset was used to test RED30 within the Noise2Noise paper. Similar to them, we use this dataset by artificially creating noise.

Evaluation Metrics.

Peak Signal-to-Noise Ratio (PSNR). It is an essential metric used to assess a reconstructed image's quality compared to the original image. PSNR quantifies the fidelity of a reconstructed image by comparing it to the original, noise-free image. It is usually expressed as a logarithmic quantity using the decibel (dB) scale. The mathematical formula is as follows:

$$PSNR_{(x,y)} = \frac{10\log_{10}[\max(\max(x),\max(y))]^2}{|x-y|^2}$$

- $PSNR(x,y)$ represents the Peak Signal-to-Noise Ratio between images x and y.
- x and y are the two images being compared.
- $max(x)$ and $max(y)$ denotes the maximum pixel values in images x and y, respectively.
- |x-y| represents the absolute difference between images x and y.

Structural Similarity Index Measure(SSIM). SSIM predicts the perceived quality of digital images, including digital television, cinematic pictures, and other types of visual content, and it also measures the similarity between the two images. SSIM focuses on structural information. SSIM accounts for how our eyes perceive changes in image structure. It incorporates luminance masking (where distortions are less visible in bright regions) and contrast masking (less visible in textured areas). By comparing the processed image with the ground truth, SSIM provides a robust measure of image fidelity, making it widely adopted in image processing.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad \textbf{(1)}$$

- x is the reference images(noise-free), and y is the processed images (denoised)

- (μ_x) and (μ_y) are the mean pixel intensities of the images x and y
- (σ_{xy}) represents the covariance between the pixel intensities of the images x and y
- C_1 and C_2 are regularization constants to stabilize the SSIM calculation and they prevent division by zero and ensure stability in the presence of small variances.

Observations and Results

Fig. 3. BSD Sample

Table 1. Average PSNR values

	BSD(avg PSNR)	SIDD(avg PSNR)
NAFNet	29.18	37.74
NAFNet + N2N	29.42	38.03

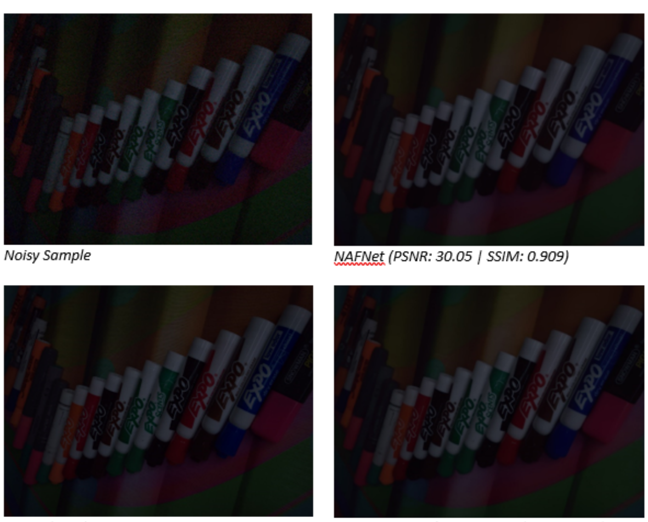

Fig. 4. SIDD Sample

Table 2. Average SSIM values

	BSD(avg SSIM)	SIDD(avg SSIM)
NAFNet	0.90	0.92
NAFNet + N2N	0.91	0.94

Examining the results, we make the following observations:

- Noise 2 Noise augmentation consistently outperforms vanilla NAFNet.
- Although modest, the change is disproportionate to the extra time of training involved.
- Noise2Noise benefits SIDD more than BSD.
- Our results are slightly worse than the authors'.

Conclusion

Utilising Noise2Noise augmentation improves the performance of the model a moderate amount. We hypothesize the following reasons to explain our observations:

- Noise 2 Noise allows the model to generalise better with more diversity.
- While helpful, not enough noisy samples were generated to capture the transform from noisy to clean images.
- SIDD's slightly greater improvement may be attributed to its limited number of samples. It therefore benefits more greatly from the increase in dataset size.
- The difference in performance may be due to our choice of parameters, lack of compute required to train for a longer duration, or our choice of resizing rather than random cropping.

Memberwise Analyses

Aryaman Trivedi

- Our experiment aims combine the high efficiency of NAFNet with the low cost of Noise 2 Noise (since the latter doesn't require the generation of clean images).
- To achieve the same, we compare the performance of vanilla NAFNet with NAFNet trained on two N2N augmented datasets.
- N2N's core idea can be visualised as a vector drawn from the noisy sample to the clean ground truth. The model can be trained to transform the noisy sample to other noisy samples generated from the same ground truth. The aggregate vector of these approximates the actual transform from the noisy to the clean image.
- While our results were not quite up to par with those published by the authors, we got reasonably close to achieving the same.

- N2N helped a modest amount in improving performance, but not enough to justify its cost of training time. Hypothetically, this wouldn't be as much of a problem with more powerful computational resources.
- Better performance of N2N on SIDD dataset is probably simply due to SIDD being so limited in number.
- N2N's lacklustre performance can be attributed to the fact that to truly approximate the noisy to clean transformation, one needs an infinite number of samples. We gave the model only $num_c reate = 2$, 3 including the clean image. In the future, I'd love to explore the impact of a greater number of samples on performance.
- Another limitation of our implementation was the use of only Gaussian noise, other noise generation methods (such as Poisson's RV) can also be explored.

Ieshaan Awasthy

- After reviewing both papers and looking at each example specifically, we could say that both techniques had their own advantages.
- NAFNet was a complete end-to-end procedure that significantly reduced the noise by not considering the non-linear activation functions like ReLU, Softmax, etc. It mainly worked by computing removing these functions or replacing them with multiplication. Due to this, there was a significant drop in the complexity of the whole procedure, and it was much more effective as it significantly reduced the computational power at the same time, not losing out on the accuracy of the model.
- Noise2Noise is a data augmentation technique that helped in a significant increase in image restoration tasks. Since it can restore the denoised images just by using the statistical approach without any other information, we can comment that the image processed had a reduced amount of noise in the images.
- When we combine both of these methods, it gives us a better output than only using NAFNet.We can see that in the results table, NAFNet with N2N had better results.
- Drawback in this review was that the amount of on which we had tested was less than what we had expected to use initially. With larger samples, we could have provided better insights about the comparisons.
- We could have explored other types of noise reduction techniques to have a better overall view of the different combinations of noise reduction techniques.

Bibliography

1. Fan, L., Zhang, F., Fan, H., Zhang, C. (2019, July 8). Brief review of image denoising techniques. Visual Computing for Industry, Biomedicine, and Art, 2(1). https://doi.org/10.1186/s42492-019-0016-7
2. Ilesanmi, A. E., Ilesanmi, T. O. (2021, June 10). Methods for image denoising using convolutional neural network: a review. Complex Intelligent Systems, 7(5), 2179–2198. https://doi.org/10.1007/s40747-021-00428-4
3. (2020, May 5). Ray Tracing Essentials Part 7: Denoising for Ray Tracing [Video]. YouTube. Retrieved April 23, 2024, from https://youtu.be/6O2B9BZiZjQ
4. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T. (2018, March 12). Noise2Noise: Learning Image Restoration without Clean Data. arXiv.org. https://arxiv.org/abs/1803.04189
5. Chen, L., Chu, X., Zhang, X., Sun, J. (2022, April 10). Simple Baselines for Image Restoration. arXiv.org. https://doi.org/10.48550/arXiv.2204.04676
6. A High-Quality Denoising Dataset for Smartphone Cameras. (2018, June 1). IEEE Conference Publication | IEEE Xplore. https://doi.org/10.1109/CVPR.2018.00182
7. Mao, X. J., Shen, C., Yang, Y. B. (2016, March 30). Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. arXiv.org. https://doi.org/10.48550/arXiv.1603.09056
8. Mansour, Y., Heckel, R. (2023, March 20). Zero-Shot Noise2Noise: Efficient Image Denoising without any Data. arXiv.org. https://doi.org/10.48550/arXiv.2303.11253
9. S. (n.d.). What is the relationship between GMACs and GFLOPs? · Issue 16 · sovrasov/flops-counter.pytorch. GitHub. https://github.com/sovrasov/flops-counter.pytorch/issues/16