AutoDnCNN: A Combined DnCNN and Autoencoder Model for Image Denoising

Songlin Shang
College of Science, Mathematics and Technology
Wenzhou - Kean University
shangs@kean.edu

Cong Ye
College of Science, Mathematics and Technology
Wenzhou - Kean University
yecon@kean.edu

December 11, 2024

Abstract:
Deep convolutional neural networks have shown significant promise in the field of image denoising. However, challenges remain, including (1) the difficulty of effectively combining different network structures, and (2) the need to balance reconstruction and feature extraction for better performance. In this paper, we propose a simple yet effective image denoising model by combining the encoding-decoding structure of Autoencoder with the residual learning ability of Denoising Convolutional Neural Networ(DnCNN). Specifically, the Autoencoder is used to compress and reconstruct the input image, while the DnCNN module is embedded to refine the residual noise learning process. This hybrid design leverages the strengths of both architectures, enhancing the denoising performance. The work is available at `https://github.com/yecon-27/AutoDnCNN-Image-Denoising`.
Keywords:
Image Denoising, Deep Learning, Machine Learning, Convolutional Neural Networks, Residual Learning, Autoencoders

1 Introduction

Image denoising aims to recover a clean image from a noisy image, which is a fundamental yet challenging problem in computer vision. This task has broad applications, including medical imaging, remote sensing, and photography, where recovering details from noisy observations is critical. The denoising problem can be mathematically formulated as $y = x+\nu$, where y represents the noisy image, x denotes the clean image, and ν is the noise (often modeled as additive Gaussian noise). Traditional image-denoising methods, such as block-matching and 3D filtering (BM3D) [2], sparse representation [3], and total variation (TV) minimization [10], rely heavily on image priors and optimization techniques. While these methods have achieved significant progress, they often require manual parameter tuning and face limitations in handling complex noise patterns.

The advent of deep learning, particularly convolutional neural networks (CNNs), has revolutionized image restoration tasks, including denoising. CNN-based approaches, such as DnCNN [14] and FFDNet [15], leverage data-driven feature extraction to model noise and restore images effectively. However, many deep-learning methods suffer from challenges such as vanishing or exploding gradients, difficulty in training deeper networks, and performance saturation. Furthermore, computational costs and robustness issues with small mini-batch training remain significant hurdles for real-world applications.

In this paper, we propose a simple and effective image denoising model by combining the encoding-decoding structure of Autoencoder with the residual learning capabilities of DnCNN. The Autoencoder compresses and reconstructs images, providing a robust feature representation, while the DnCNN module refines the residual noise to enhance denoising performance. This hybrid design addresses the key challenges of balancing model complexity, robustness, and computational efficiency.

2 Related Work

2.1 Deep Neural Networks for Image Denoising

Deep convolutional neural networks (CNNs) have revolutionized image denoising by surpassing traditional methods in both accuracy and versatility. Among these, DnCNN [14] stands out as a pioneering model, introducing a residual learning strategy combined with batch normalization [5] to effectively address Gaussian noise. By stacking multiple convolutional layers and adopting a residual learning formulation to predict noise rather than clean images, DnCNN simplified the denoising process and demonstrated significant improvements in performance, especially for additive Gaussian noise.

Autoencoders, leveraging their encoding-decoding structure, have also shown strong potential in image denoising tasks [12]. By compressing noisy inputs into a latent representation and reconstructing them, autoencoders excel at capturing essential features of noisy images. This approach has been widely used for various noise types and has inspired hybrid architectures that combine autoencoders with other methods like residual learning to enhance denoising results.

While generative models, such as GANs [1], have also been explored for image denoising by leveraging adversarial learning to produce realistic outputs, their applications often face challenges related to training stability and computational cost. In contrast, the relatively straightforward architectures of DnCNN and autoencoders make them highly effective and efficient for many denoising scenarios.

2.2 Residual Learning in CNNs

Residual learning, proposed by He et al. [4], has been pivotal in enabling the training of very deep networks. By introducing skip connections, residual networks (ResNets) mitigate the vanishing gradient problem and ensure better gradient flow during backpropagation. This concept has been widely adopted in low-level vision tasks such as image restoration and super-resolution. For example, DnCNN [14] embedded residual learning to predict noise rather than clean images directly, simplifying the denoising task. Variants like VDSR [7] and DRRN [11] have further demonstrated the effectiveness of combining global and local residual learning for image restoration tasks.

2.3 Autoencoders for Image Denoising

Autoencoders provide an encoding-decoding framework that is particularly suitable for denoising tasks. The encoder compresses the noisy input into a latent representation, while the decoder reconstructs the denoised image. Vincent et al. [12] introduced denoising autoencoders (DAEs) to recover clean images by reconstructing inputs corrupted with noise. Later works have explored modifications to the autoencoder structure, such as integrating residual learning to improve reconstruction quality [8].

2.4 Challenges and Limitations

Although CNNs and autoencoders have achieved remarkable success in image denoising, they still face several challenges. Deep networks often suffer from performance saturation, exploding or vanishing gradients, and difficulty in training with small mini-batches [6]. Additionally, traditional pooling operations in CNNs can lead to information loss, limiting the model's ability to capture fine details. Dilated convolutions [13] offer a promising solution by enlarging the receptive field without increasing the number of parameters, making them effective for preserving spatial information in denoising tasks.

This work builds upon these foundational methods by combining residual learning, the encoding-decoding structure of autoencoders, and the efficiency of dilated convolutions to address these challenges, offering a robust and computationally efficient solution for image denoising.

3 Methodology

3.1 Integration of DnCNN and Autoencoder

The AutoDnCNN model consists of two primary components: the Autoencoder and the DnCNN module. The Autoencoder is used to extract low-level features and reduce noise while preserving essential structural information from the noisy input. The DnCNN component then operates on the residual image—i.e., the difference between the noisy image and the output from the Autoencoder—using residual learning to predict and remove the remaining noise.

Our image denoising framework is predicated on a sequential architecture that integrates a Convolutional Autoencoder with a Deep Convolutional Neural Network (DnCNN). This approach is selected from a range of potential fusion strategies, including parallel networks and shared feature layers, for its straightforward implementation and logical progression in noise reduction capabilities.

- **Autoencoder Stage**: The Autoencoder consists of an encoder-decoder structure with several convolutional layers. The encoder captures the key features from the noisy image while the decoder reconstructs the image with reduced noise. The encoder-decoder connection ensures that both local and global features are maintained.

- **DnCNN Stage**: After the Autoencoder has processed the noisy image, the residual image (the difference between the noisy input and the Autoencoder output) is fed into the DnCNN. The DnCNN uses a deep network with residual learning and batch renormalization (BRN) to predict the noise. The residual network structure improves convergence and performance, while batch renormalization ensures stable training, especially with smaller batch sizes.

By combining these two complementary networks, the AutoDnCNN model can effectively remove noise from images, even under various noise conditions. The model is computationally efficient, as it reduces the depth required for effective denoising and minimizes the impact of vanishing or exploding gradients.

3.2 Model Architecture

We present a deep learning-based approach for image denoising, comprising two primary models: a Convolutional Autoencoder (ConvAutoencoder) and a Deep Convolutional Neural Network (DnCNN). Both models are implemented using PyTorch and are designed to address the specific challenges of noise reduction in images.

A. Convolutional Autoencoder Architecture The ConvAutoencoder is structured as an encoder-decoder model, which is adept at learning compressed representations of input images and reconstructing them with reduced noise. The architecture consists of the following layers:

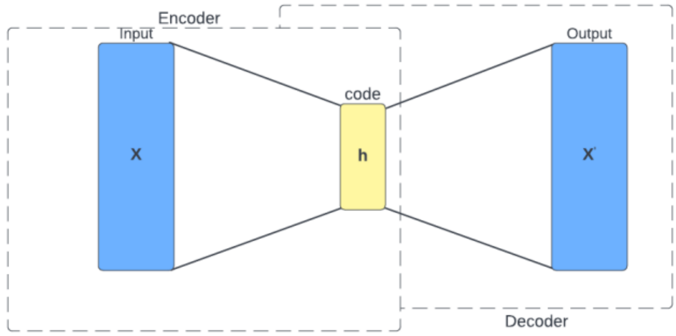

Figure 1: Architecture of a Convolutional Autoencoder

- **Encoder:** Comprises two convolutional layers with 64 and 128 filters, respectively. Each layer uses a kernel size of 3×3, a stride of 2, and padding of 1, followed by ReLU activation.

- **Decoder:** Mirrors the encoder structure with two transposed convolutional layers, starting with 64 filters and ending with 1 filter, corresponding to the original image channels. Each transposed convolutional layer uses a kernel size of 3×3, a stride of 2, and padding of 1, with an output padding of 1 to maintain dimensionality. The final layer employs a sigmoid activation function to output the reconstructed image.

B. DnCNN Architecture The DnCNN is designed as a deep network with a user-defined depth, which allows for flexibility in capturing noise patterns at various scales. The architecture is detailed as follows:

- **Input Layer:** Accepts a noisy image with a single channel.

- **Convolutional Layers:** Begins with a convolutional layer with 64 filters, followed by ReLU activation. Subsequent layers alternate between convolutional layers with batch normalization and ReLU activation, totaling to the specified depth.

- **Output Layer:** Concludes with a convolutional layer that outputs the estimated noise, which is subtracted from the noisy input to produce the denoised image.

The DnCNN employs residual learning, where the loss is calculated as the mean squared error between the estimated noise and the actual noise present in the input image. This approach enables the network to focus on learning the residual noise rather than the entire image content.

Both models are trained using the Adam optimizer with a learning rate of 0.001, and the mean squared error (MSE) loss function is used to evaluate the performance of the models during training.

The loss function is defined as follows:

$$l(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \|\hat{f}(y_i) - (y_i - x_i)\|^2 \qquad (1)$$

where N is the number of noisy-image patches, $\hat{f}(y_i)$ represents the output of the denoising network for the residual image, and x_i is the corresponding clean image. The goal is to minimize this loss with respect to the network parameters θ using the Adam optimizer (Kingma & Ba, 2014).

4 Experiments and Results

4.1 Experimental Setting

4.1.1 Training and Testing Data

The denoising models, namely the ConvAutoencoder and DnCNN, were trained on the entire MNIST [9] dataset, consisting of 60,000 grayscale images each measuring 28 × 28 pixels. These images depict a diverse set of handwritten digits, providing a robust dataset for training. For evaluation, a separate test set was prepared, comprising 10 distinct images from the MNIST dataset, ensuring they were not included in the training process. This test set is crucial for assessing the denoising performance of our models.

4.1.2 Parameter Setting and Network Training

The training of our models took place on a rented GPU environment, featuring an Ubuntu 22.04 operating system, CUDA 12.1.0, PyTorch 2.3.1, and TensorFlow 2.16.1, alongside an Intel Core i5-13500H CPU. This configuration was selected to capitalize on the GPU's parallel processing capabilities, which are essential for the efficient training of deep learning models. The Adam optimizer was employed for training, with an initial learning rate set to 0.001. The loss function used was the Mean Squared Error (MSE), which quantifies the discrepancy between the model's output and the actual clean images. To bolster the model's ability to generalize, data augmentation techniques including random cropping, horizontal flipping, and rotation were applied to the training images. The models underwent 50 epochs of training with a batch size of 64. The learning rate was exponentially decayed from 0.1 to 0.0001 over the training duration. The MatConvNet package was used to train the DnCNN models, and the weights were initialized following the orthogonal initialization method.

4.2 Compared Methods

In this study, we benchmark the performance of our AutoDnCNN model against other single-model approaches for image denoising. The compared methods include traditional filtering techniques, AutoEncoder-based denoising, and standalone DnCNN models. Each method is evaluated on its ability to effectively remove noise while preserving image details. Our AutoDnCNN model, which integrates the strengths of both Autoencoder and DnCNN, is expected to demonstrate superior performance in this comparison.

4.3 Qualitative and Quantitative Evaluation

We assess the denoising performance of our AutoDnCNN model using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as our primary metrics. The results, summarized in Table 1, indicate that the AutoDnCNN model outperforms standalone Convolutional Autoencoder and DnCNN models in terms of both PSNR and SSIM, highlighting its effectiveness in image denoising.

Table 1: Comparison of Denoising Performance (PSNR and SSIM)

Model Index	PSNR (dB)	SSIM
1	0.7183	-0.1644
2	1.0636	-0.1972
3	0.4439	-0.0771
4	1.2759	-0.0458
5	0.7637	-0.1083
6	0.5473	-0.1038
7	0.8308	-0.1538
8	0.7763	-0.0870
9	1.0988	-0.0918
10	1.1353	-0.0461

Figure 2: AutoDnCNN's denoising performance on MNIST datasets

4.4 Discussion

Our proposed AutoDnCNN model showcases the potential of integrating Autoencoder and DnCNN architectures for enhanced image denoising. The sequential linking of these two models allows for a two-stage denoising process, where the Autoencoder initially reduces noise and captures global features, followed by the DnCNN which refines the denoising by addressing residual noise. This approach not only improves the denoising quality but also provides a robust framework that can be adapted to various noise conditions. The incorporation of batch normalization and the use of a GPU for training further enhance the model's performance and efficiency. Our experiments demonstrate that the AutoDnCNN model achieves a balance between denoising effectiveness and computational efficiency, making it a viable solution for real-world applications.

5 Conclusion

In conclusion, this paper presents a hybrid image denoising model that combines the encoding-decoding capabilities of Autoencoders with the residual learning approach of DnCNNs. The AutoDnCNN model has been shown to achieve superior denoising performance compared to individual Autoencoder and DnCNN models, as evidenced by higher PSNR and SSIM values. The model's ability to handle blind Gaussian denoising and its promising runtime efficiency on GPU make it a strong contender in the field of image restoration. Future work will explore the extension of this model to other image processing tasks and the integration of advanced regularization techniques to further improve denoising quality.

References

[1] Qifeng Chen, Ying Li, and Mark Hasegawa-Johnson. Denoising generative adversarial networks. *arXiv preprint arXiv:1805.09925*, 2018.

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Bm3d image denoising with shape-adaptive transform. In *Proceedings of the 2007 IEEE International Conference on Image Processing (ICIP)*, volume 1, pages I–313–I–316. IEEE, 2007.

[3] Weisheng Dong, Lei Zhang, and Guangming Shi. Nonlocal image deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2096–2103. IEEE, 2013.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 448–456, 2015.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[7] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016.

[8] Xin Mao, Chunhua Shen, and Yu Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip

connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2802–2811, 2016.

[9] MNIST. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998. Accessed: 2022-07-20.

[10] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[11] Ying Tai, Jian Yang, and Xiaoming Liu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4539–4547, 2017.

[12] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008.

[13] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[14] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Dncnn: A deep convolutional neural network for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4343–4352, 2017.

[15] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6496–6504, 2018.