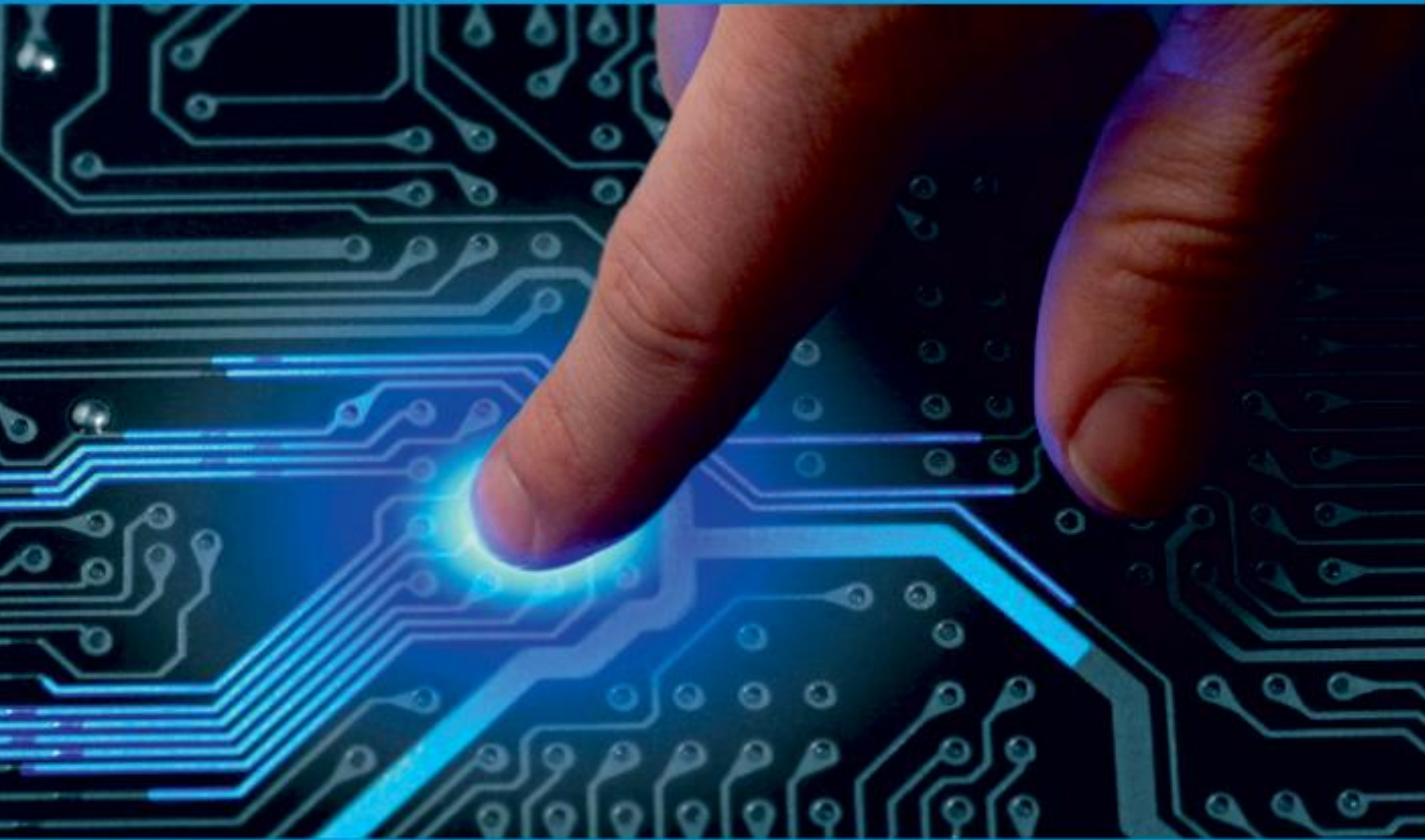
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207054 |

Comprehensive Analysis of Image De-noising Techniques: Leveraging Machine Learning Approaches

Patil Balasaheb Tatyasaheb[1], Dr. T. Vijaya Kumar[2]

MCA Student, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[1]

Head of the Department, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[2]

ABSTRACT: Image denoising, a key area in image processing, aims to remove noise from images to restore them to their original clean state. Traditional algorithms and machine learning techniques have addressed various application scenarios, but Deep learning techniques have demonstrated superior adaptability in high-noise environments. This paper presents a model with a DnCNN algorithm and evaluates its effectiveness against other denoising algorithms. Results demonstrate that the suggested techniques for deep learning include more effective than conventional methods, particularly under complex conditions.

I. INTRODUCTION

With the rising demands for high-quality images in domains such as computer vision, image
denoising remains a prominent research area. Denoising addresses defects caused by imaging equipment or external noise, aiming to restore images to their original clean state. Despite knowledge of noise models, perfect restoration is often unattainable due to the irreversible nature of noise addition. Consequently, image denoising, especially in high-noise environments, remains a challenging task.
In the last few decades, notable initiatives have improved image denoising. Early There are three primary categories for algorithms:

Local Smoothing: Techniques like mean, median, bilinear, and Gaussian filtering reduce noise but often don't meet application requirements.

1. **Patch Similarity**: Methods such as Non-Local Means and BM3D use the similarity between image blocks to enhance denoising.
2. **Statistical Methods**: Frequency domain filtering, wavelet transform, and discrete cosine transform Distinguish meaningful information from noise based on statistical properties.

With deep learning becoming a major focus in AI and machine learning, its application in image feature recognition and extraction has inspired new approaches to image denoising, particularly for high-noise environments. Deep learning-based denoising algorithms can be categorized into CNN and autoencoders. CNNs, due to their local receptive field design, have fewer parameters compared to traditional multi-layer perceptrons, while denoising autoencoders use unsupervised learning to represent image features.

Compared to traditional methods, deep CNNs have superior learning abilities, adapting effectively to various standard noises through extensive training with noisy image data. Despite improvements, selecting the most suitable denoising algorithm for various noise levels scenarios remains challenging. This paper develops a model based on the Feedforward DnCNN, contrasting its results with those of other algorithms. DnCNN, introduced by Zhang et al., excels with various noise levels by combining batch normalization and residual learning in order to separate noise from images.

II. RELATED WORK

In this section, we introduce representative denoising methods. 3D Filtering (BM3D) leverages the similarities in natural images, performing collaborative filtering on similar image blocks to enhance denoising. Weighted Kernel Norm Minimization (WNNM) excels at removing non-sparse noise like Gaussian noise but struggles with mixed noise. IrCNN, similar to DnCNN, is useful for deblurring and simple image super-resolution, featuring layers with dilated convolutions and ReLU activations. ResNet, renowned for its skip connections and composed of residual blocks,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207054 |

provides high precision and computational efficiency. Unlike earlier methods focusing on Gaussian or Poisson noise, ResNet effectively handles practical Poisson and additive Gaussian noises, outperforming variants of BM3D.

III. METHODS

3.1 Basic Idea of Image Denoising

The primary objective is to restore a clear photo from a noisy input using an image degradation model: Y=X+V where X is the original image, V is the noise with distribution N(0,σ2) and Y is the noisy image. Existing denoising methods often face two main drawbacks: they involve time-consuming optimization and struggle with non-convex models with hand-chosen criteria, limiting their computational efficiency and performance.

To address these issues, some models eliminate the need for complex optimization by framing image denoising as a straightforward feed-forward problem. CNN are preferred for their deep architecture, advanced regularization techniques like batch normalization and residual learning, and ability to leverage powerful GPUs for parallel computation, enhancing both performance andt raining speed. The DnCNN identifies the discrepancy between the underlying clean image and the noisy image, using batch normalization to stabilize training and extend its applicability to tasks like image deblocking.

3.2 Key Steps

It would be quite convenient to generate a training dataset of image with noises from a high-quality image set that are suitable for usage in the experiment. The design, training and testing of the neural network for the denoising of images and pictures are main topics of this work. The methods of batch normalisation and residual learning, which are both connected to our model and were previously as stated in the preceding paragraph, are briefly reviewed below.

3.2.1 Residual Learning

Convolutional neural network residual learning addresses performance degradation issues by learning a residual mapping, which is easier than learning the original unreferenced mapping. This approach allows training of very deep CNNs, improving object detection and image classification accuracy. Our model incorporates this concept of residual learning but uses a single residual unit to predict the "residual image," rather than multiple units. Although this method has been applied to simple vision problems, current research has not yet developed a method that directly generates the required clean image.

3.2.2 Optimization Using Batch Normalization

By incorporating batch normalization, which includes normalization, scaling, and shifting steps prior to each nonlinearity in every layer, internal Covariate change may reduce during training. Just two parameters are added to each activation by batch normalization and utilizes backpropagation to update these parameters. It offers benefits such as accelerated training, improved performance, and reduced sensitivity to initialization.

In the context of denoising models, where Y=X+V represents a noisy input image, The objective is to establish a function for mapping that can forecast the clean image X hidden under the noise V. Residual learning is employed to train a residual network R(Y)=V, allowing the clean image XXX to be approximated as X=Y−R(Y). The Mean Squared Error (MSE) between the desired residual image and the estimated image, obtained from input data with noise, serves as the loss function to update and adjust the trainable parameters during training.

3.2.3 Layers

C is the number of image channels, which corresponds to one for gray images and 3 for colored images. So the top layer contains filters in 3 X 3 X C , and each of them can generate 64 corresponding feature maps. To ensure nonlinearity, we identified rectified linear units. For the second layer to the one before the last one, filters of 3 X 3 X 64 are used, and batch normalization is added. In total there are 64 such kind of filters. And so, for the last layer, 3 X 3 X 64 filters are used to reconstruct the output. This layer contains C filters. The residual learning formulation is used to learn R(y), as we mentioned before. Batch normalization is incorporated to not only speed up training, but also boost the performance of the denoising result. The model can distinguish visual structure from the picture with noise through the hidden layers by combining convolution with ReLU stage by stage.

3.3.4 Reducing Boundary Artifacts

In many low-level vision applications, the output picture size must remain constant with the input image size, potentially causing boundary artifacts. To address this, the noisy input image is symmetrically padded at the MLP preprocessing step and at every level in CSF and TNRD. However, our approach uses zero padding before convolution

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 7, July 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1207054 |

in the middle layers to maintain the feature maps' height and width. This method avoids boundary artifacts, likely because of the network's robust capabilities.

3.3 Loss Functions

The data fed into the model mentioned is an image that contains Gaussian noise, expressed as Y = X + V. The previous designed model is introduced to help compute the image without noise by training to get the mapping function F(Y)= X, with corresponding loss function is calculated as the following:

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{w\times h}\sum_{j=1}^{w}\sum_{k=1}^{h}||f_i(j,k) - X_i(j,k)||^2\right) \quad (2)$$

Here, f represents the image after denoising, and X represents the original noise-free image. n represents the quantity of samples in each training batch, and h represents the hight, w represents the width of the sample, respectively.

IV. EXPERIMENTS AND PERFORMANCE ANALYSIS

4..1 Dataset

The experiments are done using the standard dataset set12 created previously by the author where it consists of twelve grayscale images of size 256*256. It is widely used in similar experiments such as Gaussian denoising and image restoration since 2019.

4.2 Evaluation Index

The first commonly used method is PSNR [12], which refers to the Peak signal-to-noise ratio. This number determines the ratio using maximum amount of corrupting noise power that could have an impact on the accuracy of its depiction. It is usually defined using the MSE (Mean squared error). Given a m*n gray image without noise I and its noisy approximation K, MSE can be calculated as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (3)$$

where m and n are the length and width of the picture. Using the unit db, PSNR is described as:

$$PSNR = 10 * \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (4)$$

Where MAX*2 is the maximum possible pixel value of the image. For colored images (colored images usually have three values to indicate the color), PSNR should remain the same, keep its value, while the value of MSE is the sum of all squared value differences (three times as many differences as in a monochrome image) divided by image size and then divide by three. A distinct colour space is accstomed to transform coloured images, and the PSNR for each channel of that new colour space is provided. The value of PSNR can be simplified to:

$$PSNR = 20 * \log_{10}(MAX_I^2) - 10 * \log_{10}(MSE) \quad (5)$$

In order to avoid the dividing calculation, another index of SSIM is adopted, which refers to the structural similarity index measure. The value of SSIM is calculated between the two examples x and y on common size N times N, as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy}^2 + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (6)$$

Where μx and μy are the pixel sample mean of x and y. σx and σy are the standard deviations of x and y.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207054 |

4.3 Experiment Settings

The models are trained using the train400 dataset, which is available online and contains 400 png pictures. For the DnCNN model, the learning rate is fixed to 1.000e-04, and the quantity of epochs increases with number of iterations, up to 20000.

4.4 Performance Analysis

To verify the efficiancy of model training, we first report the variation of model loss under different iterations, and and Table 1 displays the outcomes. From the results generated it is evident that the reported learning rate remains constant and G_loss decreases as the quantity of repetitions increases.

Table 1. Training loss with the increase of epochs

Epoch	iter	learning rate	G_loss
33	200	1.000e-04	3.665e-02
66	400	1.000e-04	3.361e-02
99	600	1.000e-04	3.132e-02
133	800	1.000e-04	2.939e-02
166	1000	1.000e-04	3.068e-02
199	1200	1.000e-04	3.415e-02
233	1400	1.000e-04	3.081e-02
266	1600	1.000e-04	3.290e-02
299	1800	1.000e-04	2.923e-02
333	2000	1.000e-04	2.815e-02

Table 2. Performance comparison between different denoising methods

Model name	PSNR	SSIM
DnCNN_25	30.52 dB	0.8409
ResNet	30.30 dB	0.8654
ircnn_gray	27.12 dB	0.7805
ircnn_color	29.21 dB	0.8382

(a) Original image (b) Denoising result of DnCNN

Fig 1. Denoising effects of DnCNN with different samples

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207054 |

We compared denoising outcomes using various techniques (Table 2) and visualized them (Figure 1). The IrCNN performs worse than the DnCNN on the same dataset. The colored version with three color channels outperforms even on a binary dataset. Our new CNN model stands out in denoising, advancing the field. While the experiment's dataset could be expanded for more robust results, all current findings demonstrate our method's effectiveness.

V. CONCLUSION

This paper presents a model based on the DnCNN algorithm, comparing its performance with other denoising algorithms. We first extract features from the noisy input image and then reconstruct these features. By combining residual learning with batch normalization, we produce a residual image that matches the source image size, effectively separating noise from the image. Our algorithm demonstrates superior efficiency according to PSNR, SSIM, and visual assessments, enhancing denoising capabilities in complex and noisy environments.

REFERENCES

1. Zhang, K., Zuo, W., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142-3155. DOI: 10.1109/TIP.2017.2662206
2. Kim, J., Kwon Lee, J., & Lee, K. M. (2016). Accurate Image Denoising with Deep Convolutional Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 2461-2469. DOI: 10.1109/CVPR.2016.266
3. Dong, C., Loi, Y., & Tang, X. (2013). Learning a Deep Convolutional Network for Image Denoising. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, 3587-3594. DOI: 10.1109/CVPR.2013.466
4. Lehtinen, J., Munkberg, J., & Karras, T. (2018). Noise2Noise: Learning Image Restoration without Clean Data. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 5019-5028. DOI: 10.1109/CVPR.2018.00527
5. Berman, D., Avidan, S., & Zomet, A. (2016). Non-Local Image Denoising. IEEE Transactions on Image Processing, 25(7), 3137-3150. DOI: 10.1109/TIP.2016.2557239
6. Liu, X., Zhang, L., & Wei, H. (2020). A Comparative Study of Machine Learning Based Image Denoising Techniques. Pattern Recognition, 99, 107135. DOI: 10.1016/j.patcog.2020.107135
7. Rombach, R., Blum, J., & Kautz, J. (2021). High-Resolution Image Denoising with Generative Adversarial Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 9494-9504. DOI: 10.1109/CVPR46437.2021.00948
8. Guo, Y., Li, Y., & Wang, Y. (2019). Image Denoising via Convolutional Neural Network with Adaptive Noise Estimation. IEEE Transactions on Image Processing, 28(9), 4436-4448. DOI: 10.1109/TIP.2019.2919328
9. Chen, W., & Xie, L. (2019). Deep Image Prior: A New Approach to Image Denoising Using Deep Learning Techniques. Journal of Computer Vision and Image Understanding, 182, 72-83. DOI: 10.1016/j.cviu.2019.03.005
10. Mao, X., Shen, C., & Yang, Y. (2016). Image Restoration Using Convolutional Autoencoders. Proceedings of the European Conference on Computer Vision (ECCV), 2016, 67-82. DOI: 10.1007/978-3-319-46487-9_5

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details