MULTI-NOISENET CONVOLUTIONAL NEURAL NETWORK MODEL FOR COLOUR IMAGE RESTORATION

Sundaresha M P
Assistant Professor, Kalpataru Institute of Technology, Tiptur-572201,Karnataka, India
sundresh.kit@gmail.com

Anandthirtha B. Gudi
Professor, Department of Electronics and Communication, Global Academy of Technology, Bangalore, India

Nandhish G S
Assistant Professor, Kalpataru Institute of Technology, Tiptur-572201,Karnataka, India
nandikit@gmail.com

Abstract

Artificial intelligence and deep learning have led to a rise in the use of CNN technology in several fields, including image noise reduction. If you want to further process the image for purposes like object segmentation, detection, tracking, etc. then you need to eliminate the noise from the image and restore a high quality image. But there are substantial discrepancies between the numerous deep learning approaches for picture denoising. In particular, deep learning-based discriminative learning can efficiently address the problem of Gaussian noise. Pretrained Convolutional Neural Network (CNN) model for picture denoising is presented in this paper. An advantage of using this CNN model over traditional image denoising methods like Wiener and median filtering is that it can be fine-tuned during the filtering process, whereas in old-style denoising, the limits of these algorithms are secure and cannot be changed, which is called lack of adaptivity. This study uses four different datasets and a variety of performance indicators to demonstrate the general five tasks of picture denoising. This suggested model (Multi-noiseNet) directly estimates the latent clean picture and removes noise by utilising a residual learning technique. Data from Set14 dataset showed that the suggested model had a peak signal-to-noise ratio (PSNR) of 41.17 percent and an structural similarity index measure (SSIM) of 0.972 when salt and pepper noise was 10%.

Keywords: - Convolutional Neural Network; Deep Learning; Gaussian Noise; Image denoising; Median Filtering; Salt and Pepper noise;

Introduction

It is not possible to avoid noise contamination in images during the primary stages of image amplification, compression, and transmission. Which leads to visual distortion and information loss. The presence of noise in an image has a negative impact on all of the above-mentioned processing steps. Consequently, with today's digital imaging systems, picture

denoising is vital to the process. An original image can be restored by doing image denoising, which is the removal of noise from an image. Since high frequency components such as noise, edge, and texture are difficult to identify in the denoising process, certain details may be lost in the denoised images. In general, obtaining high-quality photos by removing noise from noisy photographs is a pressing issue in today's world. Image denoising is a well-known issue that has been researched extensively for many years. Even so, it's still a difficult and open-ended undertaking. To put it another way, image denoising can be thought of as an inverse issue with several solutions.

There are various downsides to most of the preceding approaches for denoising images, including the requirement to use optimization methods for the test phase and manual parameter adjustment, as well as a single denoising model for each individual picture denoising operation. Deep learning algorithms have recently been able to overcome these disadvantages as architectures have become more versatile [1]. In the 1980s, Zhou et al. employed deep learning to denoise images, which was the first usage of the technology. A feedforward network was designed to balance denoising efficiency and performance in order to lower the high computational expenses. The researchers who conducted this study also found that the mean squared error (MSE) was not limited to neural networks as a loss function [4]. To speed up the trained network's convergence and improve denoising performance, additional optimization algorithms were implemented [5-6].

It was also found that constructing a novel network architecture, either by increasing the depth or modifying the activation function, proved to be quite effective in eradicating the noise Despite the fact that this proposed method is capable of achieving good denoising outcomes, the settings of the templates must be manually adjusted. Denoising performance can be enhanced to a considerable extent using these advanced techniques. These networks, on the other hand, were unable to quickly accommodate new plug-in devices, which restricts their practical applicability [8]. Convolutional neural networks (CNNs) were planned for the reasons outlined above [9-10].

The work contributions are abridged as follows:

- A deep trainable end-to-end CNN for denoising is proposed in this paper. For the first time, a neural network is being used to eliminate latent clean images from noisy data.
- For CNN learning, we found that lingering learning and batch normalisation may considerably enhance the speed of training and improve denoising performance. Multi-noiseNet surpasses state-of-the-art approaches in terms of quantitative measures and visual quality for Gaussian denoising with an exact level of noise.
- Image denoising may be readily handled with our Multi-noiseNet. When trained for five various noise levels, the Multi-noiseNet model is superior than those trained for a single noise level. Furthermore, the Multi-noiseNet model shows promise in solving five general picture denoising tasks, namely on four datasets such as BSD68, CBSD68, Set14, and Set24.

The rest of the paper has following information: which includes the study of existing techniques provides in Section 2. The brief explanation of proposed model with five different

noise is presented in Section 3. The validation of proposed Multi-noiseNet with existing techniques with various performance metrics is given in Section 4. Finally, the conclusion of the research study with future work is portrayed in Section 5.

2. Related Works

Additionally, for picture denoising, Jifara et al. [11] provided residual learning into a deeper CNN model. Because it relied on a deeper rather than a shallow one, the deeper CNN technique had a dependency issue over time. To deal with this issue, a number of signal-based approaches were put forth. A residual network using multi-scale cross-path was proposed by Lian et al. [12] as a way to reduce noise. These solutions all depended on enhanced CNNs to handle the noise. Image denoising relies on network architecture design.

For picture denoising, Chen et al. [13] combined Euclidean and perceptual loss. To boost denoising performance, the third method increased the receptive field size by increasing the network depth or width. Activation function, a fully connected layer, and pooling operations are among the plug-ins used in the fourth technique to improve the CNN's expressiveness. To supplement information in the deep layer of a CNN, the fifth technique used skip connections or cascade processes.

A three-phase denoising approach was proposed by Guo et al. To create noisy photos, the initial step was to apply Gaussian noise and an in-camera processing pipeline. In order to better portray genuine noisy images, the synthetic and real noise images were mixed together An asymmetric and complete variation loss loss network was employed in the second phase to estimate the noise in real noisy images. Using the initial noisy image and an estimate of noise, we were able to retrieve the latent clean image in the third step. The semi-supervised combination of CNN and prior knowledge was designed to address the issue of unpaired noisy images [15]. In order to build a dataset, researchers utilised a cluster method to first classify each patient's CT scan into numerous categories. Then they collected images from different patients that fell into those similar categories and fed the dataset into an HD-GAN. For picture denoising and classification, the GAN was utilised to cope with the dataset obtained. Denoising images with deep learning and the CNN model is presented by Ghose et al. [16]. Adding 1% to 10% Gaussian white noise to an image and then using the CNN model to denoise it is how this analysis is done. This is where edge factor, texture, uniformity and non-uniformity of colour and shape, smoothness and object structure come into play in qualitative analysis. PSNR, SSIM, and MSE are three metrics that are used in the quantitative analysis to compare the outcomes of the CNN-based method with those of the traditional or standard methods of image denoising. Only one noise is taken into account for the validation procedure, which shows that the CNN model is capable of effectively removing a large amount of Gaussian noise and restoring the image details and data. By incorporating information from different modalities, he and his colleagues [17] hope to increase the quality of the brief frames. Deep learning-based joint filtering is used to incorporate information from PET and high-resolution MR images simultaneously into the short frames. The training label is a composite of all dynamic frames, down-sampled to 1/10th of the counts, and utilised as the training input in each sample. During training, the loss function is composed of L1-norm and two regularizations depending on gradients. It has been demonstrated in simulations, however, that the suggested strategy reduces statistical noise while maintaining picture detail and provides

quantitative improvements over Gaussian, guided filter, and CNN trained using the mean squared error.

The authors Cao et al. [18] propose a spatial-spectral global reasoning network that takes into account both local and global information when removing noise from hyper-spectral images (HSIs). Global relational information is modelled and analysed using two new modules. Global spatial linkages between pixels in feature maps and across channels are being modelled by one and the same method. Using several representations, such as hierarchical local features, global spatial coherence, cross-channel correlation, and multiscale abstract depiction, the newly built global reasoning network can assist combat complex noise.

3. Proposed Methodology

As shown in Figure 1, the proposed model uses a pre-processing block to resize the input noisy image. Multi-noiseNet, a denoising CNN model that has been developed for generic picture denoising, is fed these scaled images. There are two ways in which a deep CNN model can be trained for a specific task: Designing a network and building a model are two different but intertwined tasks. By modifying the VGG network [19], we can make it more suited for picture denoising, and we can also establish the network's depth based on the real patch sizes used in the most advanced denoising methods. Using a residual learning formulation and batch normalisation, we can train models quickly and enhance denoising performance at the same time.

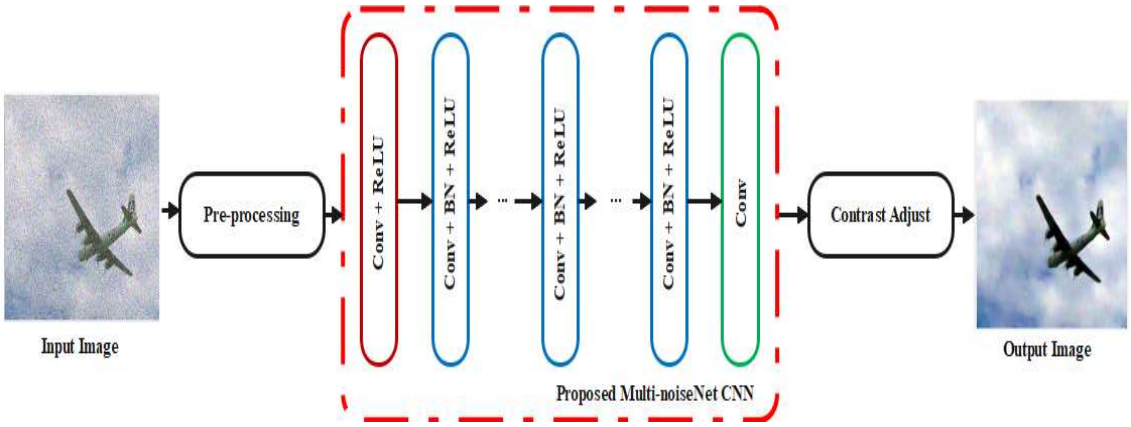


Fig.1. Proposed system Architecture.

3.1. Dataset description

In entirely image denoising means work on noisy images under three dissimilar noise variances $\sigma \in [30, 50, 75]$. For the test images, we use different datasets for a full evaluation: BSD68, CBSD68, Set14 and Set24. The total dataset contains of 174 images from the separate test set of the 74 images. Some of the sample dataset images are shown in Fig. 2. Those are broadly used testing images. All the dataset images are evaluated in two different sizes 256×256 and 512×512.

Fig. 2. Some of the widely used testing images.

3.2. Multi-noiseNet Model

Our Multi-noiseNet takes as input a noisy observation with the formula y=x+v. Denoising models like MLP and CSF are trained on a mapping function F(y)=x so they can learn how to anticipate the latent clean picture. R(y)v for Multi-noiseNet can be trained using the residual learning formulation, as shown in the following figure (y). The mean squared difference between the desired residual images and the estimated ones based on noisy inputs.

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

can be modified as the loss function to trainable parameters Θ in Multi-noiseNet. Here $\{(y_i, x_i)\}_{i=1}^{N}$represents N noisy-clean training image pairs. In the following, the architecture of the proposed Multi-noiseNet is defined.

3.2.1. Multi-noiseNet Architecture

There are three sorts of layers for the Multi-noiseNet with D depth: First, 64 3x3xc filters are employed to build 64 feature maps, then rectified linear units (ReLU,max(0,•)) are used for nonlinearity. A grey image has one channel, a colour image has three channels (c =1 for grey image, c =3 for colour). Batch normalisation [21] is introduced between convolution and ReLU for layers 2(D-1) and 64 filters of size 3364 are utilised. (iii) Conv: the output is reconstructed using c filters of size 3364 for the final layer. R(y) is learned using a residual learning formulation, and batch normalisation speeds up training while also improving denoising performance, in our Multi-noiseNet model. Multiple NoiseNet's convolution with ReLU allows it to gradually remove the image structure from the noise over time. A similar technique is used in methods such as EPLL and WNNM to remove noise iteratively, but our Multi-noiseNet is trained end-to-end. We'll talk more about why batch normalisation and residual learning go hand in hand in the future.

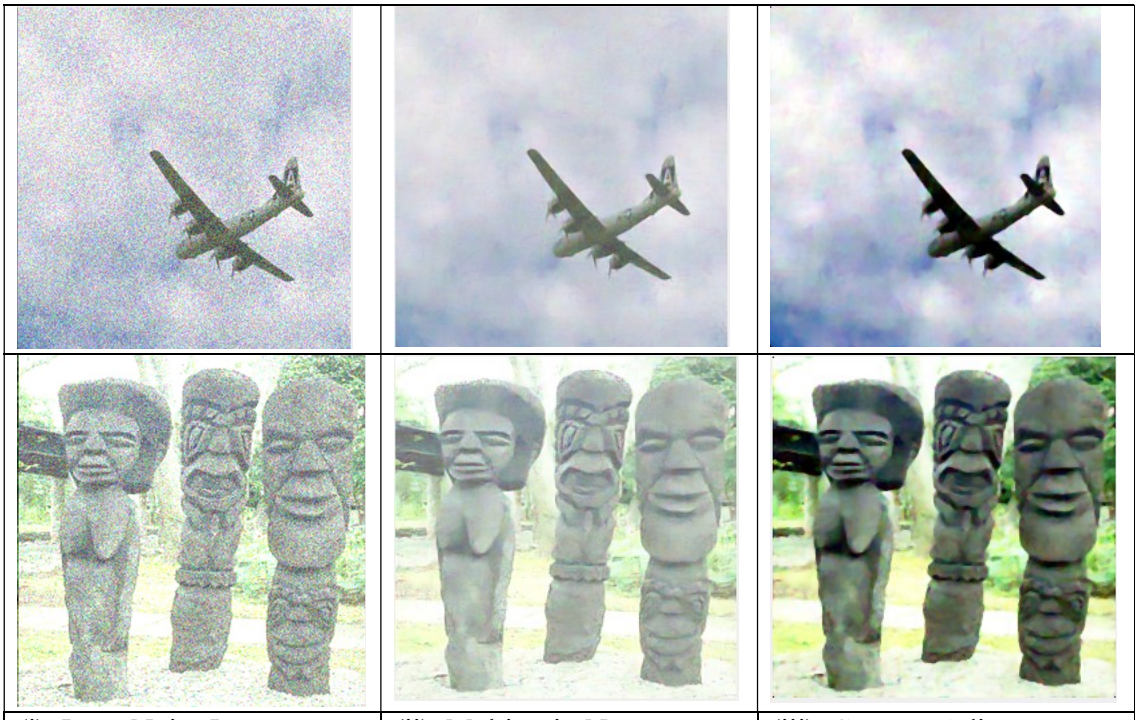
Boundary Reducing: In many low-level vision applications, the output picture size is frequently required to be the same as the input image size. Artifacts at the boundary could result from this. In CSF and TNRD, the same padding approach is used before each stage, but the noisy input picture border is symmetrically padded in MLP during the preprocessing stage [22]. We explicitly pad zeros before convolution to ensure that each feature map of the intermediate layers has the same size as the input image, unlike the previous techniques. There are no boundary artefacts when using the simple zero padding strategy. That the Multi-noiseNet has such a potent ability is presumably the reason for this good property.

Parameter Setting and Network Training: Learn R(y) for predicting residual v using the loss function in Eqn (1), which is used. For the initialization of the weights, we utilise the approach described in [23], with a weight decay of 0.0001, an initial momentum of 0,9, and a batch size

of 128 for SGD. Our Multi-noiseNet models are trained over 50 iterations. For each of the 50 epochs, the learning rate decreased exponentially from 1e1 to 1e4.

Proposed Contrast Adjustment

The output image of the Multi-noiseNet is transfer to the next Contrast Adjustment model, which consist of two basic image enhancement parts such as image intensity Adjustment and median filter. (i) In image intensity Adjustment, the intensity values in the image I to new values in J. By saturates the bottom 1% and the top 1% of all pixel values, this operation increases the contrast of the output image J. (ii) In median filter, performs median operation of the image I in two dimensions. Both this image intensity Adjustment and median filter operations improves the visual quality of the output image. The sample of before process and outputs is defined in the below figure.3.



(i). Input Noise Image	(ii). Multi-noiseNet	(iii). Contrast Adjustment

Fig.3. Sample output images.

4. Results and Discussion

As of right now, there aren't any mathematical or particular procedures for evaluating visual analysis. Artifacts, edge protection, and texture preservation are the three main requirements for visual analysis in general. Several performance measurements, including as PSNR and the structural similarity index measurement (SSIM) [24], are used to assess the correctness of image denoising algorithms. An Intel Core i5–4570 CPU 3.30 GHz with an Nvidia Titan X GPU is used for all of our research. All of the BM3D techniques use MATLAB, with the exception of the core, which is implemented using a compiled C++ mexfunction that runs in the background in parallel (R2019b). Proposals on the GPU can be trained in roughly six hours, one day, and three days.

4.1. Metrics of denoising performance

Image denoising algorithms are evaluated using PSNR and SSIM as quantitative measures of performance. A picture's PSNR can be calculated by using a ground truth image (x) as a starting point.

$$PSNR(x,\hat{x}) = 10.log_{10}\left(\frac{255^2}{||x-\hat{x}||_2^2}\right) \quad (2)$$

In addition, the SSIM index is intended by

$$SSIM(x,\hat{x}) = \frac{(2\mu_x\mu_{\hat{x}}+C_1)(2\sigma_{x\hat{x}}+C_2)}{(\mu_x^2+\mu_{\hat{x}}^2+C_1)(\sigma_x^2+\sigma_{\hat{x}}^2+C_2)} \quad (3)$$

Here, we have the means and variances of each of the two variables (x) and their covariance (xx) as well as two constants (C 1 and C 2) to prevent instability. The visual quality of a group of photos must be compared, despite the fact that quantitative measurements cannot accurately reflect this. For evaluating a denoising algorithm, edge and texture preservation is critical.

4.2. Performances evaluation of proposed model

In this section, the proposed network is validated using five different noises on various datasets in terms of PSNR and SSIM, which is provided from Table.1 to Table.5. All the mentioned results are averaged results.

Table.1. PSNR/SSIM evaluation of the Gaussian noise.

Dataset	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
BSD68	33.59/0.928	29.90/0.856	27.83/0.794	26.36/0.740	25.15/0.690	24.14/0.655	22.14/0.537	18.92/0.363
CBSD68	33.63/0.933	30.03/0.881	27.73/0.832	25.99/0.785	24.99/0.742	23.36/0.74	21.64/0.602	19.00/0.455
Set14	33.70/0.914	30.29/0.854	28.09/0.801	26.62/0.752	25.33/0.706	24.19/0.670	22.27/0.557	19.08/0.388
Set24	35.81/0.948	31.84/0.912	29.19/0.878	27.15/0.848	25.45/0.822	23.97/0.795	22.13/0.892	19.53/0.556

For the analysis on BSD68 dataset, the proposed model achieved high PSNR (i.e.33.59%) and high SSIM (i.e.0.928) only, when the Gaussian noise is 10%. When the percentage of noises are increased, the performance of model degrades for this dataset analysis. For instance, when it is 50%, the model achieved 25.15% of PSNR and 0.690 of SSIM, where the same model achieved only 18.92% of PSNR and 0.363 of SSIM, when the noise level is 80%. While comparing with all datasets, the proposed model achieved high PSNR values on Set24 dataset. For example, the PSNR is 35.81% on Set24 and the PSNR is 33.63% on CBSD68, when the noise level is 10%. The SSIM is 0.822 on Set24 dataset and SSIM is nearly 0.742 on CBSD68 and Set14 datasets, when the noise level is 50%.

Table.2. PSNR/SSIM evaluation of the salt and pepper.

Dataset	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
BSD6	40.19/0.	35.75/0.	33.26/0.	31.52/0.	30.18/0.	29.07/0.	28.09/0.	27.18/0.

8	979	950	919	888	859	830	801	771
CBSD68	40.05/0.979	35.92/0.953	33.57/0.927	31.93/0.903	30.66/0.877	29.61/0.853	28.71/0.830	27.92/0.808
Set14	41.17/0.972	35.08/0.969	33.049/0.965	31.74/0.873	29.82/0826	28.32/0.845.	27.89/0.823	26.58/0.738
Set24	39.75/0.978	35.74/0.953	33.46/0.928	31.86/0.903	30.61/0.879	29.59/0.855	28.70/0.833	27.92/0.812

The proposed model achieved nearly 31% to 35% of PSNR and achieved 0.89 to 0.95 of SSIM on all four different data set, when the Salt and pepper noise level is 20% to 40%. When the noise level is 10%, the PSNR values are 40.19%, 40.05%, 41.17% and 39.75% on BSD68, CBSD68, Set14 and Set24. While comparing with Gaussian noise, the proposed model achieved better performance on Salt and pepper noise, even at the highest level of noise percentage. For instance, the SSIM is 0.771, 0.808, 0.738 and 0.812, where the PSNR is 27.18%, 27.92%, 26.58% and 27.92% on all different datasets for the noise level at 80%. While comparing with all datasets, the proposed model achieved low performance on PSNR and SSIM on Set14 dataset for the noise level varies from 60% to 80%.

Table.3. PSNR/SSIM evaluation of the speckle.

Dataset	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
BSD68	29.18/0.802	28.69/0.798	27.35/0.763	22.44/0.495	18.19/0.316	15.73/0.220	14.10/0.166	12.90/0.132
CBSD68	33.61/0.926	29.65/0.838	27.170/0.754	25.31/0.682	23.75/0.616	22.29/0.546	19.42/0.352	16.67/0.233
Set14	33.59/0.928	29.90/0.856	27.83/0.794	26.36/0.740	25.25/0.690	24.14/0.655	22.14/0.537	18.92/0.363
Set24	33.58/0.926	29.91/0.852	27.80/0.787	26.30/0.731	25.06/0.680	23.97/0.637	22.28/0.536	19.63/0.374

When the percentage of noises are increased, the performance of model degrades for this dataset analysis. For instance, when it is 30%, the model achieved 27.35% of PSNR and 0.763 of SSIM, where the same model achieved only 15.73% of PSNR and 0.220 of SSIM, when the noise level is 60%. In the analysis of CBSD68, the proposed model achieved nearly 33% to 23% of PSNR and 0.9 to 0.7 of SSIM, when the noise level reaches from 10% to 50%. The BSD68 dataset provides low performance on PSNR and SSIM at various noise levels than other datasets. For instance, the model achieved 14.10% of PSNR and 0.166 of SSIM on BSD68, where the same

model achieved 22.14% of PSNR and 0.537 of SSIM on Set14 dataset, when the noise level is 70%.

Table.4. PSNR/SSIM evaluation of the poisson.

Dataset	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
BSD6	30.58/0.	29.68/0.	27.85/0.	22.55/0.	18.24/0.	15.82/0.	14.14/0.	12.90/0.

8	832	818	772	502	326	231	166	132
CBSD68	30.45/0.814	29.70/0.815	26.28/0.707	20.89/0.435	17.58/0.298	15.76/0.221	14.14/0.172	13.62/0.153
Set14	29.51/0.798	29.14/0.808	26.07/0.707	20.82/0.430	17.54/0.289	15.70/0.215	14.53/0.175	13.61/0.149
Set24	30.16/0.840	30.50/0.839	28.39/0.765	22.51/0.476	18.69/0.304	16.29/0.216	14.59/0.161	13.25/0.127

In the analysis of poisson attack, the proposed model achieved nearly 29% of PSNR on BSD68, CBSD68 and Set14, where Set24 has 30% of PSNR value, when the noise level is 20%. However, when the noise level is 50%, the PSNR on BSD68 and Set24 achieved nearly 18% and other two dataset has 17% of PSNR value. When the noise levels are 10%, 20% and 30%, the SSIM is nearly 0.8 to 0.7, where it is suddenly decreased to 0.2 to 0.1, when the noise level is 50% to 80%. From this analysis, it is clearly proving that the visual quality is low on poisson attack, which must be improved in the near future work.

Table.5. PSNR/SSIM evaluation of the localvar.

Dataset	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
BSD68	27.10/0.713	27.71/0.750	26.56/0.715	21.48/0.444	18.21/0.297	15.85/0.212	14.28/0.165	13.09/0.134
CBSD68	27.83/0.750	28.35/0.799	27.06/0.743	21.89/0.468	18.22/0.306	15.83/0.217	14.21/0.166	13.00/0.134
Set14	27.73/0.729	28.19/0.755	26.74/0.711	21.50/0.447	18.12/0.304	15.90/0.219	14.28/0.169	13.08/0.138
Set24	28.89/0.777	29.24/0.748	27.49/0.748	22.00/0.476	18.28/0.316	15.92/0.227	14.24/0.173	13.01/0.138

Same like poisson attack, the noise levels on localvar are 10%, 20% and 30%, the SSIM is nearly 0.7, where it is suddenly decreased to 0.4, 0.2 to 0.1, when the noise level is 40% to 80% on all datasets. The PSNR value is also low on all dataset, when the noise lvel reaches 50%. For instance, the PSNR is nearly 28% to 26% on 10% to 30% of noise level, where PSNR is 21% on 40% of noise level. But, all datasets have nearly 18% to 13% of PSNR, when the noise level is 50% to 80%. From this experiments, it is proving that the proposed Multi-noiseNet has low performance on poisson and this attack that needs to be improved for better visual quality of input images.

4.3. Comparative analysis

Table 6 displays the average PSNR values of several approaches on the BSD68 dataset. As can be shown, Multi-noiseNet outperforms the competition in terms of PSNR. In terms of PSNR, the MLP and TNRD approaches outperform the BM3D benchmark by a factor of roughly 0.35dB. Only a few approaches can beat BM3D by more than 0.03 dB on average, according to [25]. Multi-noiseNet outperforms BM3D on all three noise levels, with an average difference of 0.6%. Our DnCNN-B, in particular, outperforms rival methods that are trained for a known specific noise level even with a single model with unknown noise levels. Although

Multi-noiseNet outperforms BM3D by around 0.6dB at 50, this is extremely close to the estimated PSNR bound of [26], which is 0.7dB higher than the BM3D performance at 50.

Table.6. The average PSNR (db) results of dissimilar approaches on the BSD68 dataset.

Methods	**BM3D**	**WNNM**	MLP	**EPLL**	**CSF**	**TNRD**	**DnCNN**	**Multi-noiseNet**
σ = 15	31.07	31.37	-	31.21	31.24	31.42	31.73	31.61
σ = 25	28.57	28.83	28.96	28.68	28.74	28.92	29.23	29.16
σ = 50	25.62	25.87	26.03	25.67	-	25.97	26.23	26.23

An image restoration method's testing speed is just as crucial as the visual quality of the results. For denoising photos with a noise level of 25, the run times of several approaches are shown in Table 6. We also provide GPU run speeds for our CSF, TNRD, and Multi-noiseNet techniques, since they are all well-suited to parallel computing on GPU. Accelerate GPU calculation of suggested Multi-noiseNet by using Nvidia cuDNNv5 deep learning library.

Table.7. Processing speed evaluation.

Methods	**BM3D**	**EPLL**	WNNM	**MLP**	**TNRD**	CSF	**DnCNN**	**Multi-noiseNet**
256×256	0.65	25.4	203.1	1.42	0.45/ 0.010	2.11	0.74/ 0.014	0.90/0.016
512×512	2.85	45.5	773.2	5.51	1.33/ 0.032	5.67/0.92	3.41/ 0.051	4.11/0.060
1024×1024	11.89	422.1	2536.4	19.4	4.61/ 0.116	40.8/1.72	12.1/0.200	14.1/ 0.235

The time it takes for various ways to process photos with noise levels of 25, 256, 512, and 1024 1024 in seconds. CPU (left) and GPU (right) timings for CSF, TNRD and our suggested Multi-noiseNet are given (right). Also worth noting is that the GPU run time varies widely depending on the GPU and the GPU-accelerated library, so it is difficult to compare CSF, TNRD, and our suggested Multi-noiseNet to one other in terms of performance. In order to speed up CSF and tnrd on the GPU, we just replicate the run times from the original publications.

Conclusion

In this research work, a new deep learning method, namely Multi-noiseNet is developed and validated its effectiveness on general tasks of image denoising. Following the mathematical modeling of proposed model, Multi-noiseNet learns the information from training data, and learns a specific feature extraction for each image. The performance of denoising is improved by adopting the formulation of residual learning and integrated with batch normalization. Moreover, contrast adjustment such as median filter with image intensity adjustment is used for better visual quality of input images. The experiments are approved on gray scale images and color images from four various datasets by inducing five different attacks in terms of PSNR and SSIM. The results proved that the proposed model achieved 35.81% of PSNR and 0.948 SSIM on Set24 dataset, when the Gaussian noise level is 10%. However, the proposed model

achieved 13.01% of PSNR and 0.138 SSIM on Set24 dataset, when the localvar noise level is 80% that provides low visual quality. It must be improved by incorporating learning rate optimization the proposed model as future work.

References

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

[2]. Fukushima, K., Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and Cooperation in Neural Nets. Springer, pp. 267–285.

[3]. Zhou, Y., Chellappa, R., Jenkins, B., 1987. A novel approach to image restoration based on a neural network. In: Proceedings of the International Conference on Neural Networks, San Diego, California.

[4]. Greenhill, D., Davies, E., 1994. Relative effectiveness of neural networks for image noise suppression. In: Machine Intelligence and Pattern Recognition. Vol. 16. Elsevier, pp. 367–378.

[5]. Bedini, L., Tonazzini, A., 1992. Image restoration preserving discontinuities: the bayesian approach and neural networks. Image and Vision Computing 10 (2), 108–118.

[6]. Gardner, E., Wallace, D., Stroud, N., 1989. Training with noise and the storage of correlated patterns in a neural network model. Journal of Physics A: Mathematical and General 22 (12), 2019.

[7]. Sivakumar, K., Desai, U. B., 1993. Image restoration using a multilayer perceptron with a multilevel sigmoidal function. IEEE Transactions on Signal Processing 41 (5), 2018–2022.

[8]. Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36 (4), 193–202.

[9]. Lo, S.-C., Lou, S.-L., Lin, J.-S., Freedman, M. T., Chien, M. V., Mun, S. K., 1995. Artificial convolution neural network techniques and applications for lung nodule detection. IEEE Transactions on Medical Imaging 14 (4), 711–718.

[10]. Ren, W., Pan, J., Zhang, H., Cao, X., Yang, M.-H., 2020. Single image dehazing via multi-scale convolutional neural networks with holistic edges. International Journal of Computer Vision 128 (1), 240–259.

[11]. Jifara, W., Jiang, F., Rho, S., Cheng, M., Liu, S., 2019. Medical image denoising using convolutional neural network: a residual learning approach. The Journal of Supercomputing 75 (2), 704–718.

[12]. Su, Y., Lian, Q., Zhang, X., Shi, B., Fan, X., 2019. Multi-scale cross-path concatenation residual network for poisson denoising. IET Image Processing.

[13]. Chen, X., Zhan, S., Ji, D., Xu, L., Wu, C., Li, X., 2018. Image denoising via deep network based on edge enhancement. Journal of Ambient Intelligence and Humanized Computing, 1–11.

[14]. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L., 2019. Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1712–1722.

[15]. Meng, M., Li, S., Yao, L., Li, D., Zhu, M., Gao, Q., Xie, Q., Zhao, Q., Bian, Z., Huang, J., et al., 2020. Semisupervised learned sinogram restoration network for low-dose ct image reconstruction. In: Medical Imaging 2020: Physics of Medical Imaging. Vol. 11312. International Society for Optics and Photonics, p. 113120B.

[16]. Ghose, S., Singh, N. and Singh, P., 2020, January. Image denoising using deep learning: Convolutional neural network. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 511-517). IEEE.

[17]. He, Y., Cao, S., Zhang, H., Sun, H., Wang, F., Zhu, H., Lv, W. and Lu, L., 2021. Dynamic PET image denoising with deep learning-based joint filtering. IEEE Access, 9, pp.41998-42012.

[18]. Cao, X., Fu, X., Xu, C. and Meng, D., 2021. Deep spatial-spectral global reasoning network for hyperspectral image denoising. IEEE Transactions on Geoscience and Remote Sensing, 60, pp.1-14.

[19]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference for Learning Representations, 2015.

[20]. H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2392–2399.

[21]. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015, pp. 448–456.

[22]. Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," to appear in IEEE transactions on Pattern Analysis and Machine Intelligence, 2016.

[23]. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in IEEE International Conference on Computer Vision, 2015, pp. 1026– 1034.

[24]. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612.

[25]. A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2833–2840.

[26]. A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *European Conference on Computer Vision*, 2012, pp. 73–86.