Research on an Image Denoising Algorithm based on Deep Network Learning

Jian Liang1, *, Pengxu Chen2, Mian Wu3

1College of Engineering and College of Science, China Agricultural University, Beijing, 100083
2Merchant Marine College, Shanghai Maritime University, Shanghai, 201306
3Department of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning, 116024

*Corresponding author: Jian_liang@cau.edu.cn

Abstract. In recent years, the learning method based on depth convolution neural network has achieved unprecedented results in image denoising, and it has become a research hotspot to obtain better image denoising effect by adjusting network structure and parameters. Noise reduction convolution neural network adopts residual learning method in depth neural network, which not only improves the effect of noise reduction, but also solves the problem of blind noise reduction to a certain extent. Its disadvantage is that the algorithm has a long convergence time. In this paper, the structure of noise reduction convolution neural network is further improved, and an image denoising algorithm based on deconvolutional neural network is proposed. The main features of this paper are as follows: 1) in the original network structure, deconvolution neural network is introduced to optimize the residual learning method; 2) A new loss function calculation method is proposed. BSD68 and SET12 test data sets are used to verify the proposed method. The experimental results show that the noise reduction performance of this algorithm is compared with the noise reduction convolution neural network algorithm. Under the same noise reduction effect, the convergence time of this algorithm is shortened by 120%. 138%. At the same time, compared with the traditional deep learning image denoising algorithm, the denoising effect and running efficiency of this method are also improved.

Keywords: Image denoising deep learning residual learning deconvolutional neural network.

1. Introduction
Noise reduction is an important application in image processing and a classic inverse problem in image processing. The purpose of image noise reduction is to separate the noise signal from the contaminated image as much as possible, remove the noise signal, retain the real image information, and bring better visual effects. Given an additive white Gaussian noise model, it can be expressed as \( y = x + v \) (\( x \) represents the original image, \( v \) represents noise). Image noise reduction is to restore a clean image \( x \) from a noisy image \( y \). Although many image noise reduction algorithms have been proposed...
in the past few decades, the current deep convolutional neural network (DCNN) method has a better noise reduction effect than traditional noise reduction methods. Therefore, algorithms based on deep learning have become one of the hotspots in current research on image noise reduction [1].

DCNN method treats image denoising as a supervised regression problem and makes effective use of external prior information. Through a large number of paired clean-noise images, a high complexity multi-layer network structure constraint function is obtained, which pushes the way of learning external prior conditions to the limit. The advantages of DCNN image denoising methods are mainly reflected in two aspects: on the one hand, DCNN has a deep hierarchical structure, which can effectively improve the ability and flexibility of image feature mining; on the other hand, in the process of deep learning, regularization method, RELU activation function, batch standardization processing and other methods are adopted to speed up the training efficiency and improve the performance of noise reduction.

In this paper, a deconvolution noise reduction neural network called DnDNN is proposed. In the network structure constructed in this paper, each hidden layer includes convolution operation and deconvolution operation, the convolution operation is to extract the feature information of the image, including edges, lines and other features, and the deconvolution operation is to restore the image to the size of the image before convolution, and the last layer is fully connected as the output layer. In this paper, the method of image reconstruction by deconvolution network is used to reduce noise, and residual learning is introduced into the process of model training. The pure noise image is obtained after residual learning, and the pure noise image contains less information than the clean image, the training is simpler and the effect is better, and the image denoising method using deconvolution network needs less network layers. It can also accelerate the convergence speed of the network and improve the noise reduction performance.

2. DnDNN denoising method

Generally speaking, training a deep convolutional neural network model for a specific task is divided into two steps: 1) design the network structure; 2) learn the model from the training data. In the network structure design, in order to make it more suitable for image noise reduction, this paper introduces deconvolution on the basis of DnCNN and improves the original loss function.

2.1. Proposed DnDNN structure

![Figure 1 DnDNN network structure](image)

The network structure designed in this paper is shown in figure 1, and the designed network depth is 10. Each layer contains a set of convolution operations and a set of deconvolution operations, convolution operation is used to extract the feature information of the image, including edges, lines and other features, deconvolution operation is used to restore the image, deconvolution in the reconstruction process, remove part of the noise information that can not be reconstructed; the last layer is a fully connected layer, which is composed of a set of convolution operations to output the image. In the above structure, each convolution operation (or deconvolution operation) is followed by batch normalization (BatchNormalization,BN) and RELU, to speed up the convergence speed and improve the training effect.
2.2. Algorithm steps

1. The noisy original image is subjected to convolution calculation to obtain a feature map, and then the feature map is deconvolved to obtain a reconstructed image; 2. The noisy original image is subtracted from the reconstructed image to obtain a residual image; 3. The original image, subtracting the noise image and the corresponding label image, and then making the difference with the residual image in step 2, and performing the minimization calculation; 4. Set the number of iterations, when the number of iterations is reached, the training stops, otherwise repeat 1~3.

In this experiment, the picture block size is 30×30, the step size is 6, and 64 convolution kernels with a size of 3×3 are set in each layer. In order to ensure that the image input and output sizes are the same, zero padding is used. Except for the last fully connected layer, each layer processing includes convolution and deconvolution, and no pooling layer is used.

2.3. Proposed loss function calculation method

The training data is a noisy image y=x+v, and the mapping function f(y)=x is found by learning the pattern of the training data. The purpose is to predict the potential clean image. This paper uses residual learning to obtain the mapping $R(y) \approx v$, and then obtains $x = y - R(y)$, so the calculation formula of the loss function is as follows:

$$l(\Theta) = \| R(y_i; \Theta) - (y_i - x_i) \|^2_F$$

In the above formula, $x_i - y_i$ is the sample residual value (that is, the noise image value minus the clean image value)

3. Experimental results and analysis

For Gaussian noise reduction with known or unknown noise levels, the training set is 400 grayscale images with 180 / 180 pixels. The method in this paper is trained under the training set of $\sigma \in [0,55]$ noise level. In order to verify the noise reduction effect of this method on the known noise level, the Gaussian noise images with noise intensity of $\sigma = 15, 25$ and 50 are selected in this paper. The size of each patch block is 30 to 30 pixels. In order to expand the data sample and reduce the influence of edge effect, each image in 400pixel 180x180 image is clipped twice according to its size (1, 0.9), and then $R(y_i; \Theta) y$ is used to regularize the expected residual, $l(\Theta)$ by L2 to obtain the loss function value of each patch block. In DnCNN, the final loss function is the accumulation of each patch block loss function, and then the loss function as small as possible is obtained by optimization. Because the mean square error function (MSE) is sensitive to the change of gradient, the robustness of loss function in the process of convergence is poor, and the convergence speed is slow.

In order to shorten the convergence time of the model and ensure the noise reduction effect of the model, this paper combines the respective advantages of L1 regularization and L2 regularization to optimize the above loss function. The formula is as follows:

$$L(t) = \frac{1}{2N} \sum_{i=1}^{N} t^2(\sqrt{1 + l(\Theta)} / t - 1)$$

In the above formula, t is a hyperparameter, which is used to adjust the gradient of the loss function $L(t)$. Compared with the square loss, the above equation is not sensitive to outliers, but it also maintains differentiable properties. It is based on the absolute error (MAE) but becomes MSE when the error is small. We can use the super parameter t to adjust the gradient. When t tends to 0, it degenerates to MAE, and when t tends to infinity, it degenerates to MSE. The loss function makes use of the respective advantages of MAE and MSE. It can not only maintain its continuous derivative, but also make use of the characteristic of MSE gradient with error reduction to obtain a more accurate
minimum value, but also has better robustness to abnormal values. Through the experimental analysis, when the value of $t$ is set to 0.25, it can not only ensure the training speed, but also accelerate the convergence speed. Use smaller steps than DnCNN to retain more edge information of the image. In order to verify the noise reduction effect of this method, BSD68 and SET12 data sets are used to test.

### 3.1. Parameter setting and network training

In order to take into account the effect of noise reduction and convergence efficiency, the depth of DnDNN network is set to 10. Each of the first nine layers contains convolution and deconvolution, and the last layer is a fully connected layer. Using the improved loss function of section 3.2, the number of image blocks per batch of small batch training is 128 and the number of iterative cycles is 50. The initial learning rate is 0.01, and the learning rate is reduced to 1/10 per 10 cycles of iteration. The running environment of this experiment is AMD Ryzen 5 2600X core processor @ 3.60GHz, NVIDIA GeForce GTX 2070 GPU and 16G memory space. Using BSD68 and SET12 test data sets, under three different noise intensities, this method is compared with several classical noise reduction methods, including two noise reduction methods based on non-local similarity, BM3D[2] and WNNM[3], one generation method, EPLL[4], and two discriminant training-based methods (i.e., MLP[5], DnCNN-B[6]). The experimental results are shown in Table 1 and Table 2. Figure 2 shows a comparison of the denoising results obtained from several common image denoising algorithms with a noise level of $\sigma=50$ on the BSD68 test set. From the value of PSNR, the denoising result of DnDNN method is similar to that of DnCNN-B and superior to other methods. As can be seen from the picture, DnDNN can not only restore sharp edges, clear textures and fine details, but also produce visually pleasant effects in smooth areas.

### Table 1

Average PSNR (DB) results of different methods on the BSD68 data set

| $\sigma$ | BM3D    | WNNM    | EPLL    | MLP     | DnCNN-B | ours    |
|----------|---------|---------|---------|---------|---------|---------|
| $\sigma=15$ | 31.07   | 31.37   | 31.21   | ___     | 31.61   | 31.66   |
| $\sigma=25$ | 28.57   | 28.83   | 28.68   | 28.96   | 29.16   | 29.17   |
| $\sigma=50$ | 25.62   | 25.87   | 25.67   | 26.03   | 26.18   | 26.14   |

### Table 2

The average PSNR (DB) results of different methods on the SET12 data set

| $\sigma$ | BM3D    | WNNM    | EPLL    | MLP     | DnCNN-B | ours    |
|----------|---------|---------|---------|---------|---------|---------|
| $\sigma=15$ | 32.37   | 32.7    | 32.14   | ___     | 32.68   | 32.74   |
| $\sigma=25$ | 29.97   | 30.26   | 29.69   | 30.03   | 30.36   | 30.38   |
| $\sigma=50$ | 26.72   | 27.05   | 26.47   | 26.78   | 27.15   | 27.13   |

Table 2 The noise reduction result of a picture with a noise level of 50 from BSD68 using different noise reduction methods.
Figure 3 describes the change of the PSNR value of the model training with the iteration cycle when the noise intensity is $\sigma = 25$ based on the gradient optimization algorithm SGD, the residual learning method and the improved loss function of the DnDNN in this paper. It can be seen from Figure 4 that when the DnDNN method is used for training, when the iteration cycle reaches about 18 times, the PSNR value tends to be stable at 30.38dB, its convergence speed is very fast, and the training effect is also very good.

![Denoising Results of DnDNN](image)

**Figure 3** Convergence effect diagram of $\sigma = 25$ Gaussian noise reduction training under DnDNN model

### 3.2 Running time

In addition to the noise reduction effect, the running efficiency and convergence time of the model are also important indicators to evaluate the performance of the model. Table 3 shows the running time of some classical image denoising algorithms when the noise level is 25 and the image size is 256x256, 512x512 and 1024 × 1024 respectively. It can be seen from the data in the table that the running efficiency of DnDNN method BMD and TNRD method is slightly lower than that of DnCNN-B method. Considering that DnDNN method has the advantages of blind denoising, the DnDNN method proposed in this paper still has high computational efficiency. Considering the convergence time of the model, because the DnDNN method has blind noise reduction performance, when comparing the convergence time of the model, the DnDNN method is mainly compared with the DnCNN-B method. As can be seen from figure 5, the training result of the DnDNN algorithm proposed in this paper tends to be stable when iterating for 20 cycles, and the PSNR value is 30.38dB. The method proposed in this paper takes 317 seconds for each iteration, which is slightly longer than that for DnCNN-B (301 seconds for DnCNN-B). However, the convergence time of this method is much less than that of DnCNN-B. The convergence time of DnCNN-B is about 50 iterations, the convergence time is about 14932 seconds, and the convergence time of DnDNN is only about 6263 seconds. Based on the statistical analysis of a large number of images, the convergence time of this method is 120% less than that of DnCNN-B.

**table 3** Different noise reduction methods have a noise level of 25 and an image size of 256×256, 512×512, and 1024×1024, respectively, running time (seconds) under the CPU and GPU, for CSF, TNRD, DnCNN and the DnDNN proposed in this article. We give the running time of CPU (left) and GPU (right)

| $\sigma$ | BM3D | WNNM | EPLL | MLP | DnCNN-B | ours   |
|---------|------|------|------|-----|---------|-------|
| $\sigma = 15$ | 0.65 | 203.1 | 25.4 | 1.42 | 0.90/0.016 | 0.93/0.017 |
| $\sigma = 25$ | 2.85 | 773.2 | 45.4 | 5.51 | 4.11/0.060 | 4.30/0.063 |
| $\sigma = 50$ | 11.89 | 2536.4 | 42.1 | 19.4 | 14.1/0.235 | 14.82/0.242 |
Figure 3 describes the change of the PSNR value of the model training with the iteration period when the noise intensity is $\sigma=25$ based on the gradient optimization algorithm SGD, the residual learning method and the improved loss function of the DnDNN in this paper. It can be seen from Figure 4 that when the DnDNN method is used for training, when the iteration cycle reaches about 18 times, the PSNR value tends to be stable, at 30.38dB, its convergence speed is very fast, and the training effect is also very good.

3.3. Empirical analysis

In order to compare the image noise reduction effect between the method in this paper and the DnCNN-B method, this paper analyzes the factors that affect the training convergence speed through multiple sets of experiments. Network parameters related to convergence speed include learning rate, batch block, step size, number of convolutional layers, and loss function.

Figure 4 Comparison of the convergence speed of Gaussian noise reduction training for several different denoising methods under $\sigma=25$
By setting different values for the factors that affect the training convergence speed, this paper conducted 5 sets of experiments on image denoising, and the results are shown in Figure 4.

The first set of experiments is the DnCNN-B method, which uses L2 regularization as the loss function, the batch size is 40×40, the step size is 10, the initial learning rate is 0.001, and the number of convolutional layers is 17. After 30 iterations, the learning rate is reduced to 1/10 of the original.

The network structure of the second group of experiments and the fourth group of experiments is the same as that of DnCNN-B, except that its parameter settings have been adjusted; the third group of experiments is based on the DnCNN-B method and sets an inverse after each convolutional layer. Convolutional layer, the rest of the structure is the same as DnCNN-B, the specific settings are as follows:

The second set of experiments is the Test1 method. The batch size is set to 40×40, the step size is 10, the initial learning rate is 0.001, and the number of convolutional layers is 17. After every 10 iterations, the learning rate is reduced to 1/10 of the original.

The third set of experiments is the Test2 method, using L2 regularization as the loss function, setting the batch size to 30×30, the step size to 6, the initial learning rate is 0.001, and the number of convolution and deconvolution junction layers is 10, when each iteration After 10 times, the learning rate is reduced to 1/10 of the original.

The fourth set of experiments is the Test3 method, using L2 regularization as the loss function, setting the batch size to 30×30, the step size to 6, the initial learning rate to 0.001, and the number of convolutional layers to 17. After 30 iterations, the learning rate is reduced to 1/10 of the original.

The fifth set of experiments is the DnDNN method proposed in this paper. On the basis of adding an improved loss function and deconvolution layer to the network structure, the batch size is set to 30×30, the step size is 6, the initial learning rate is 0.001, and the number of layers is 10. After every 10 iterations, the learning rate is reduced to 1/10 of the original.

As can be seen from figure 4, the DnDNN denoising method has the best denoising effect in five groups of experiments, and tends to be stable when iterated for 20 cycles. Compared with the DnCNN-B method, the main contribution to the improvement of the training convergence speed of this method is to add the deconvolution layer to the original network structure, and the improved loss function is used to calculate the residual value of the image.

4. Conclusion
In this paper, in the original deep residual learning network structure, by introducing a deconvolution layer and improving the loss function, an image denoising algorithm based on deconvolution denoising neural network is proposed. The introduction of deconvolution improves the quality and efficiency of noise image residual learning, and the optimization of the loss function further accelerates the training convergence speed and improves the noise reduction effect. Different from the discriminant model trained for specific image noise levels, the DnDNN algorithm proposed in this paper has a strong ability to deal with unknown noise levels. Compared with the DnCNN-B algorithm, the convergence time of the algorithm in this paper is greatly reduced while ensuring the noise reduction effect and operating efficiency.

References
[1] Lefkimmiatis S. Universal Denoising Networks: A Novel CNN Architecture for Image Denoising [J]. 2017.
[2] Dabov, Kostadin, Foi,. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. [J]. IEEE Transactions on Image Processing, 2007.
[3] Simonyan, Karen, Zisserman, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.
[4] Zoran D, Weiss Y. From learning models of natural image patches to whole image restoration [J]. 2011:479-486.
[5] Burger H C, Schuler C J , Harmeling S . Image denoising: Can plain neural networks
compete with BM3D?[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2012.

[6] Zhang K, Zuo W , Chen Y , et al. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising [J]. IEEE Transactions on Image Processing, 2016, 26(7):3142-3155.