

Image Denoising based on Deep Learning

Zeyu Li *

Department of Computer Science, McGill University, Montreal, Canada

* Corresponding author email: zeyu.li2@mail.mcgill.ca

Abstract. Image denoising has always been one of the research hotspots in the field of image processing, which aims to remove the noise from the imaging device or external noise environment and other interfering factors in the image to restore the noisy image to the original clean and noise-free image. Mature algorithms and machine learning techniques have been developed previously for different application situations and specific computer vision works. Image denoising based on deep learning can adaptively learn image content and is suitable for image denoising tasks in high-noise environments. This paper builds a model based on the representative image denoising algorithm DnCNN, and discusses the performance difference with other denoising algorithms. All the results show the new methods are usually more efficient than traditional ones which can process pictures that are under more complex conditions.

Keywords: Image Denoising; Batch Normalization; Residual Learning.

1. Introduction

With the continuous improvement of image quality requirements in many scientific fields such as computer vision, image denoising has always been one of the hot research topics in the field of image processing. The denoising is usually based on some images with certain local defects, which mainly come from the interference of imaging equipment or external noise environment and other factors. The main problem of image denoising research is how to restore a noisy image to the original clean and noise-free image. In fact, due to the irreversible process of adding noise, even if the various data of the noise model are known, most of the time it is still impossible to perfectly restore the real image. Therefore, image denoising, especially in environments with high noise intensity, is still a very challenging task.

Over the past few decades, many efforts have been invested to improve denoising performance [1-2]. Early image denoising algorithms were devoted to mining image features from different perspectives, and representative algorithms mainly include the following three categories. (1) Image denoising based on local smoothing. On the basis of noise reduction, this kind of technology concentrates on local smoothing of noisy pictures while preserving as many images' information as feasible. Common smoothing methods include mean filtering, median filtering, bilinear filtering, and Gaussian filtering. Although fast and easy to implement, the denoising effect of such methods often cannot meet the actual application requirements. (2) Image denoising based on patch similarity. This type of method takes advantage of the fact that different picture blocks in a picture are likely to be similar to each other, and makes full use of local smoothness and global self-similarity to denoise from the global perspective of the picture. Representative algorithms include Non-Local and Block-matching and 3D filtering (BM3D), etc. (3) Statistically based image denoising. The traditional approach is frequency domain filtering, and this kind of algorithm is based on the statistical properties of natural pictures to denoise. Frequency-domain filtering separates noise from useful information by altering the space, assuming that the high-frequency portion of the frequency domain picture is more likely to be noise and the low-frequency portion is more likely to be useful information. Wavelet transform and discrete cosine transform, etc.

As deep learning has gradually become a research hotspot in the area of artificial intelligence and machine learning, the successful application of deep convolutional neural networks in the fields of image feature extraction and recognition provides new ideas for solving image denoising problems, especially when image has high level of natural optical noise or noises are entered manually by testers.

Existing image denoising algorithms based on deep learning can be divided into methods based on convolutional neural network and methods based on autoencoder. Thanks to the design of the local receptive field, the number of parameters of the convolutional neural network is greatly reduced compared with the ordinary multi-layer perceptron. Denoising autoencoders are a special class of neural networks that perform denoising via unsupervised learning, using learned hidden layer units to represent image features. Compared with the traditional image denoising method, the deep convolutional neural network has a stronger learning ability. By using a large number of noisy image sample data for training, it can effectively improve the adaptability of the network model to different standard noises, and make it has stronger generalization ability.

Although the previous works are different in the design of training objectives, selection of training features and size of training set, they can significantly improve the denoising effect. However, limited by the extremely complex actual denoising scenarios, how to choose the most suitable denoising algorithm for different noise scenarios is still an open issue. In this paper, we build a denoising model based on Feed forward denoising convolutional neural networks (DnCNN) and compare its results with other representative denoising algorithms. DnCNN is a denoising algorithm based on deep learning proposed by Zhang et al [3], which uses a single denoising model to achieve the task of image denoising and usually performs better when the data are designed to fill with various unknown levels of noises. For our method, we first extract the features of the input noise image and reconstruct the image features extracted from the previous step. Finally, by combining residual learning with batch normalisation to produce a residual image that is exactly as the same size as what we input, we can effectively separate the photo from the noise.

Focusing on the above aspects, the arrangement of this paper is introduced here. In the second section, we first survey the most representative denoising algorithms. In Section 3, we will introduce the basic theory, key steps and implementation details of our method in detail. We then report the results of our approach in and discuss issues and future directions for image denoising in Section 4.

2. Related Work

In this section, we first introduce related representative denoising methods. Block Matching and 3D Filtering (BM3D) [4] is a well-known denoising algorithm, which fully exploits the similarities existing in natural images rather than relying on probabilistic image priors. In BM3D, many image blocks that are similar to each other perform collaborative filtering at the same time, which makes each image block provide context information reference for denoising of other image blocks. Weighted kernel norm minimization (WNNM) [5] generally has good performance for removing non-sparse noise such as Gaussian noise but struggles with mixed noise. Better performance can be achieved using adaptive median filtering. IrCNN [6] is an approach quite similar to DnCNN that we mainly discuss, which can also be used for deblurring and simple image super-resolution. It consists of seven layers and three different blocks, where the first layer is a dilated convolution + ReLU block, and then in the middle layer is the same block for batch normalization, and the last is a dilated convolution block. The computational load grows when the filter size is increased, adding additional parameters. ResNet[7-8] introduces a skip connection to fit the input of the previous layer to the next layer without modifying the input. ResNet is also a deep convolutional neural network composed of residual blocks. At the same time, high precision and good computational efficiency are the key features of ResNet. Previous deep learning methods mainly focus on denoising images with Gaussian or Poisson corruption, ResNet here can deal with denoising images with more practical Poisson and additive Gaussian noises [9]. ResNet is considered and proved to be more efficient, has better performance than popular variants of the famous BM3D algorithm.

3. Methods

3.1 Basic Idea of Image Denoising

The primary goal is to restore a clear photo from a chaotic input using an image degradation model, which is defined as:

$$Y = X + V \quad (1)$$

Here, X is indicating the original image without noise, and v is the image that has noise with a distribution of $N(0, \sigma^2)$. σ is referring to the noise standard deviation, and so Y is indicating the noisy image. There used to be a lot of models with high denoising quality and good performance, but at the same time most of the existing denoising methods still have two main drawbacks to deal with. The testing portion of those methods typically involves a challenging optimization problem, which adds time to the denoising process. As a result, the majority of approaches hardly ever manage to attain high performance without compromising computational efficiency. Second, there is some room to improve denoising performance because most models are non-convex models and involves some hand-chosen criteria. To overcome the disadvantages listed above, some models are driven from the previous ones which have improvements in some of the cases. The continuous optimization technique in the testing stage can be eliminated by the generated models. This kind of training code approaches image denoising as a more straightforward differing learning issue, which entails removing the noise from a noisy picture using feed-forward, instead of developing a discriminative model with a clear picture prior.

To be more specific, we would like to mention three most popular reasons for using the CNN network instead of other similar models and algorithms. First of all, Convolutional Neural Network is known for its deep architecture, which helps to increase the capacity and flexibility for both finding and identifying the characteristics of certain image or picture. Secondly, there are already multiple advanced achievements in regularization the network and optimizing the learning methods for training CNN. The ones we might be familiar with would be batch normalization [10] and residual learning [9], which have already been proved to have good performance. Both methods introduced above can accelerate the training process and at the same time improve the overall performance of the convolutional neural network. Third, CNNs are capable for parallel computation on modern improved powerful GPUs, which means the running time can be improved through adding more graphical computing units to increase the computation power during the training process. We will call it as DnCNN, which refers to Denoising Convolutional Neural Network. The network is invented to find a potential existing image, which refers to the discrepancy between the picture with noise and the underneath relatively pure image, which means it does not directly output the clean image we expected. With operations in the buried layers, the latent clean image is eliminated. The training performance is then stabilised using the batch normalisation procedure. This model can also be extended for handling other general and commonly known image denoising tasks, such as 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 can be used 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 described in the preceding paragraph, are briefly reviewed below.

3.2.1 Residual Learning

Convolutional neural network residual learning is intended to address the well-known performance degradation issue without taking into account the fact that training accuracy decreases as network depth increases. The residual network intentionally learns a residual mapping for a few stacked layers since it is easier to learn than the original unreferenced mapping. It is simple to train exceptionally

deep CNN using such a residual learning approach, and better object detection and picture classification accuracy have been attained. The suggested model incorporates the concept for residual learning as well. In contrast to the residual network, which employs several residual units, the model we're talking about here uses a single residual unit to anticipate the "residual picture." The residual image prediction technique has been applied in the past to several simple vision issues. But for now, in the current stage of our research and studies, there is still no work which can in overall generate the image we required directly using any method.

3.2.2 Optimization Using Batch Normalization

By adding a normalisation step, a scale step, and a shift step before the nonlinearity in each of the layer, it is suggested to use batch normalisation to lessen internal covariate shift. Only two parameters are added for each activation during batch normalisation, and back-propagation can be used to update these parameters. The benefits of batch normalisation include quick training, improved performance, and little initialization sensitivity. The input of the network is a set to be a picture with noise, indicated by $Y = X + V$. The denoising models are usually designed to get a mapping function in order to inversely predict the clean image hidden under the noise. The residual learning formulation can then be used to train and get a residual graph, indicated as $R(Y) \approx V$. For the result, we can get the ideal clean image by using the difference between the two, which is $X = Y - R(Y)$. The average MSE calculated using the desired residual image and the estimated image we get from a data fed in with noise can be modified to get the loss function to update and modify the trainable parameters.

3.3 Network Architecture

Given denoising model involves three different kinds of layers: Conv+ReLU, the previous layer with batch normalization, and a single Conv layer, which we will introduce in the following subsections.

3.3.1 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 $3 \times 3 \times C$, 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 $3 \times 3 \times 64$ are used, and batch normalization is added. In total there are 64 such kind of filters. And so, for the last layer, $3 \times 3 \times 64$ 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.

3.3.2 Reducing Boundary Artifacts

The output picture size must typically remain constant with the input image size in many low-level vision applications. Boundary artefacts could result from this. The noisy input picture's border is symmetrically padded at the MLP preprocessing step, but this padding method is used at every level in CSF [11] and TNRD. To make the feature maps indicated before of the middle layers that should have the same height and width as the data we fed in, we immediately pad zeros before convolution, in contrast to the approaches mentioned before. We discover that there are no boundary artefacts when using the straightforward zero padding method. This beneficial quality is possibly due to the network's strong capabilities.

3.4 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 number of samples in each training batch, and h represents the height, w represents the width of the sample, respectively.

4. 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 $m \times n$ 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 defined as:

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

Where MAX_I^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 used 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 μ_x and μ_y are the pixel sample mean of x and y . σ_x and σ_y are the standard deviations of x and y .

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 number of epochs increases with number of iterations, up to 20000.

4.4 Performance Analysis

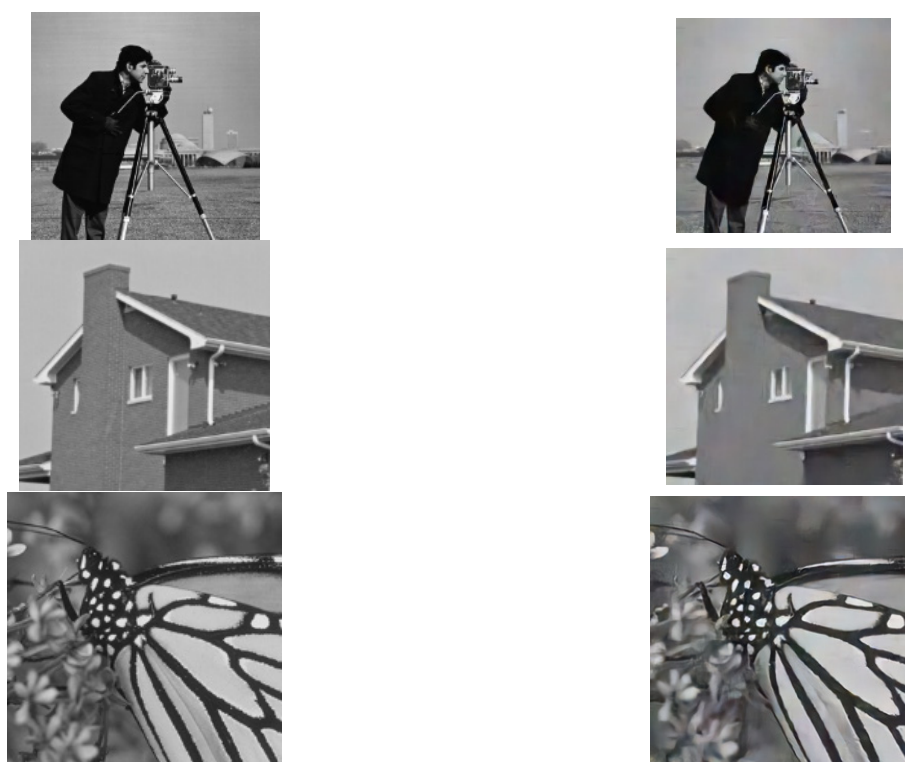
To verify the effectiveness of model training, we first report the variation of model loss under different iterations, and the results are shown in Table 1. From the results generated we can see that the reported learning rate remains constant and G_loss decreases as the number of iterations increases.

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

In addition, we also compare the denoising results using different kinds of methods, as shown in Table 2. Finally, we visualize the denoising results of different methods in Figure 1. It is easy to see that the IrCNN performs worse than the DnCNN we talked about on the same data set. The colored version with three color channels has a better performance even on a binary dataset. After the

experiments we introduced on different models and algorithms, the new CNN model performs as an outstanding one from the others where the functions and abilities it showed in the area of denoising bring the problem to a new stage. For the experiment itself, the quantity could still be expanded with more datasets with various features. The standard dataset is composed of clean images so the results might not be as optimal as expected. With a more detailed designed dataset corresponding to the weak points of each model, the result might differ much larger than what we have now. All the results shown in the experiment demonstrate the effectiveness of our method.

5. Conclusion

This paper builds a model based on the representative image denoising algorithm DnCNN, and discusses the performance difference with other denoising algorithms. In particular, we first extract the features of the input noise image and reconstruct the extracted image features. Then, by combining residual learning with batch normalisation to produce an image leftover that is the same size as the source image, we successfully separate the image from the noise. The algorithm in this work exhibits superior efficiency according to PNSR, SSIM, and good visual effects of all the experimental data, which can improve the denoising ability in an environment with more complicated destruction and noisy disturbance.

References

- [1] S. Ghose, N. Singh and P. Singh, "Image Denoising using Deep Learning: Convolutional Neural Network," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020.
- [2] D. Yang and J. Sun, "BM3D-Net: A Convolutional Neural Network for Transform-Domain Collaborative Filtering," in IEEE Signal Processing Letters, Jan. 2018.
- [3] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," in IEEE Transactions on Image Processing, July 2017.
- [4] D. Yang and J. Sun, "BM3D-Net: A Convolutional Neural Network for Transform-Domain Collaborative Filtering," in IEEE Signal Processing Letters, Jan. 2018.
- [5] S. Gu, L. Zhang, W. Zuo and X. Feng, "Weighted Nuclear Norm Minimization with Application to Image Denoising," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.
- [6] K. Zhang, W. Zuo, S. Gu, L. Zhang, "Learning Deep CNN Denoiser Prior for Image Restoration", Apr. 2017.
- [7] W. Wu et al., "Outside Box and Contactless Palm Vein Recognition Based on a Wavelet Denoising ResNet," in IEEE Access, 2021.
- [8] L. Marcos, F. Quint, P. Babyn and J. Alirezaie, "Dilated Convolution ResNet with Boosting Attention Modules and Combined Loss Functions for LDCT Image Denoising," 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2022.
- [9] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", Dec. 2015.
- [10] S. Ioffe, C. Szegedy, "Batch Normalization: Accepting Deep Network Trained by Reducing Internal Covariate Shift", Mar. 2015.
- [11] K. Wendel, N. G. Narra, M. Hannula, P. Kauppinen and J. Malmivuo, "The Influence of CSF on EEG Sensitivity Distributions of Multilayered Head Models," in IEEE Transactions on Biomedical Engineering, vol. 55, no. 4, pp. 1454-1456, April 2008.
- [12] K. Joshi, R. Yadav and S. Allwadhi, "PSNR and MSE based investigation of LSB," 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016.