

## Research Article

# Convolutional Neural Network Combined with Half-Quadratic Splitting Method for Image Restoration

Ran Li, Lin Luo, and Yu Zhang 

School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

Correspondence should be addressed to Yu Zhang; zhang.yuer@163.com

Received 14 March 2020; Revised 11 July 2020; Accepted 18 July 2020; Published 28 August 2020

Academic Editor: Shahid Hussain

Copyright © 2020 Ran Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Generally, there are mainly two methods to solve the image restoration task in low-level computer vision, i.e., the model-based optimization method and the discriminative learning method. However, these two methods have clear advantages and disadvantages. For example, it is flexible for the model-based optimization method to handle different problems, but large quantity of computing time is required for better performance. The discriminative learning approach has high computing efficiency, but the application scope is seriously limited by the fixed training model. It would be better to combine the advantages of these two methods. Luckily, with the variable splitting techniques, we insert the trained convolutional neural network (CNN) for denoising as one model to the model-based optimization method to solve other image restoration problems (e.g., deblurring and super-resolution). Final experimental results show that our denoising network is able to provide strong prior information for image restoration tasks. The image restoration effects can reach or approximate the most advanced algorithm in such three tasks as denoising, deblurring, and super-resolution. Moreover, the algorithm proposed in this paper is also the most competitive in terms of computational efficiency.

## 1. Introduction

Image restoration is one interesting issue in low-level computer vision [1–3]. Generally, image restoration is to restore the potential clean images from the degraded observation images. Loading of different degradation matrices for the clean images forms the image restoration issue to be solved. Since image restoration is one ill-posed problem with numerous solutions, a prior (or regularization) method is required to restrain the solution space [4, 5]. The maximum a posteriori (MAP) approach fully considers the image prior knowledge, and it is based on Bayesian perspective to convert the original issue to the issue of optimal solution  $\hat{x}$ :

$$\hat{x} = \arg \max_x \log p(y|x) + \log p(x), \quad (1)$$

where  $p(y|x)$  represents likelihood probability and  $p(x)$  represents the prior probability of the clear image and is

irrelevant to the degraded image. The Equation (1) can be further corrected to

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - Hx\|_2^2 + \lambda \Phi(x), \quad (2)$$

where  $(1/2)\|y - Hx\|_2^2$  is the fidelity term and  $H$  is the degraded matrix which describes the image degradation process.  $\Phi(x)$  is the regularization term, which represents the image prior information used to restrain the final solution and convert the ill-posed problem to a good state problem.  $\lambda$  is the trade-off parameter between the fidelity term and the regularization term.

Usually, image restoration is the solution Equation (2) and is generally divided into two categories: model-based optimization and discriminative learning. The model-based optimization approach is to solve directly Equation (2), but these algorithms require a large quantity of iterative computation and greatly reduce the computation efficiency. The discriminative learning approach keeps optimizing the

loss function in a number of training sets including degradation images to get the prior parameter  $\Theta$ . The final target is the network output, and the target distance is the minimum [6–10], i.e.,  $\min_{\Theta} \ell(\hat{x}, x)$  whose constraint condition is

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - Hx\|_2^2 + \lambda \Phi(x; \Theta), \quad (3)$$

As shown above, we can conclude that the most obvious difference between the model-based optimization approach and the discriminative learning approach is that the model-based optimization approach can deal with flexibly various kinds of image restoration tasks by loading different kinds of degradation matrices  $H$ , while the discriminative learning approach requires different degradation image training sets to finish the restoration tasks. For example, the NCSR (one model-based optimization method [11]) algorithm can freely perform the tasks of image denoising, deblurring, and super-resolution, whereas the discriminative learning approach must be designed separately from the three tasks above, like MLP [12], DCNN [13], and SRCNN [14] separately executing these tasks. As for the particular task like deblurring, the model-based optimization method (e.g., IDDBM3D [15] and NCSR [11]) can well handle the degraded images loading different fuzzy kernels while the discriminative learning approach (e.g., MLP [16]) must train different learning models for different fuzzy kernels.

The discriminative learning approach sacrifices the flexibility, but the trained network model can have higher computation efficiency. With the gradual increase of GPU performance, the computation time can be further reduced. These two image recovery methods have their own merits and demerits, and it would be promising to combine their merits. Luckily, latest research suggests that with variable splitting techniques, like the alternating direction method of multipliers (ADMM [17]) and half quadratic splitting (HQS [18]), can handle separately the fidelity term and regularization term [7], which makes it possible to combine the network model trained with discriminative learning approach and the model-based optimization method to better handle the image restoration task.

This paper is aimed at training two sets of rapid and effective CNN denoising models (to handle separately the gray image and colorful image), where the latest CNN technologies like ReLU activation function [19], batch normalization [20], residual learning [21] and dilated convolution [22] are also adopted to get better image recovery performance. The network model is combined with the HQS method so as to provide powerful image prior information before the model-based optimization method is adopted. Besides, two sets of CNN denoising models are taken as one model to be inserted to the model-based optimization method to solve more widespread image restoration problem (e.g., image deblurring and single image super-resolution). Final experimental results show that our algorithm performs well in both image restoration flexibility and computational efficiency,

and the restoration can reach or approximate the latest advanced algorithm level.

## 2. Related Work

In the past, there have been many attempts to apply the denoiser trained in the model-based optimization method to other image restoration tasks. Some researchers have proposed an iterative decoupling image deblurring method based on Nash equilibrium derivation (IDDBM3D) [15]. Some people also proposed a method for single-frame image super-resolution (SISR) with a priori of CBM3D denoiser [23]. Through iterative update back projection and CBM3D denoising steps, the PSNR of this method is better than SRCNN [14]; It is further proposed that the BM3D denoising algorithm can be regarded as a priori and then integrated into the image deblurring scheme by using the improved Lagrange method [24]. Later, a plug and play prior framework based on the ADMM method was proposed [25], which adopted an iterative scheme similar to that in Reference [15]. It can be noted here that before the above method was proposed, the similar concept of plug and play was proposed in Reference [5], in which the half-quadratic splitting method (HQS) can be used in a variety of subproblems of image restoration. Some researchers have also proposed a denoising autoencoder based on a multichannel model and applied it to the restoration task of single-channel gray-scale infrared image [26]. In addition, there are also methods to apply the half-quadratic splitting method to image super-resolution task and achieve good results [27]. Almost every method mentioned above indicates that fidelity and regularization can be decoupled, so that the existing denoising model can solve a wider range of image restoration tasks.

It can also be found that as long as the preconditions for the fidelity term and regularization term are well separated, then denoising priori can be inserted into the model-based iterative optimization algorithm through certain mathematical methods. For this reason, these iterative methods are usually decomposed into a denoising problem and other subproblems. In the next chapter, we give the variable separation algorithm of this paper in detail, that is, the half-quadratic splitting method (HQS). Although HQS can be regarded as a general way to deal with different image restoration problems, the denoiser can also be applied to different image restoration tasks in advance.

## 3. HQS Method

The variable splitting technology can combine the advantages of two kinds of image restoration algorithms. This technology separates the fidelity and regularization terms, and the separated regularization term only corresponds to the subproblem of image denoising [28–33].

With the HQS method in variable splitting technology, after one auxiliary variable  $z$  is introduced, the Equation (2) can be rewritten to

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - Hx\|_2^2 + \lambda \Phi(z). \quad (4)$$

The constraint condition is  $z = x$ ; however, the original HQS approach is to solve the following problem:

$$\zeta_{\mu}(x, z) = \frac{1}{2} \|y - Hx\|_2^2 + \lambda\Phi(z) + \frac{\mu}{2} \|z - x\|_2^2, \quad (5)$$

where  $\mu$  is the penalty parameter in the regularization term, which keeps decreasing in the iteration of solution. As for the solution of Equation (5),  $z$  can be taken as constant, then

$$x_{k+1} = \arg \min_x \|y - Hx\|_2^2 + \mu \|x - z_k\|_2^2, \quad (6)$$

similarly, when  $x$  is taken as constant:

$$z_{k+1} = \arg \min_z \frac{\mu}{2} \|z - x_{k+1}\|_2^2 + \lambda\Phi(z). \quad (7)$$

As shown in Equations (6) and (7), the HQS method has successfully separated the fidelity term  $(1/2)\|y - Hx\|_2^2$  and regularization term  $\Phi(x)$ , which divides the original big problem into two small individual problems. As for Equation (6), the method of solution derivative equal to 0 can be used.  $x_{k+1}$  is also the solution of the following

$$\frac{\partial(\|y - Hx\|_2^2 + \mu\|x - z_k\|_2^2)}{\partial x} = 0. \quad (8)$$

It is easy to get:

$$x_{k+1} = (H^T H + \mu I)^{-1} (H^T y + \mu z_k). \quad (9)$$

As for Equation (7), it can be changed to the form of Equation (10) as follows:

$$z_{k+1} = \arg \min_z \frac{1}{2(\sqrt{\lambda/\mu})^2} \|x_{k+1} - z\|_2^2 + \Phi(z). \quad (10)$$

According to the Bayesian probability, the Equation (10) can be explained as  $z_{k+1}$  is the result of denoising of Gaussian denoiser with the noise level of  $\sqrt{\lambda/\mu}$  for the image  $x_{k+1}$  with the noise. According to this theory, a series of Gaussian denoisers after CNN training can utilize the conclusions of this chapter to make the image restoration in other directions. In order to represent this point more vividly, the Equation (10) can be changed to the form as follows:

$$z_{k+1} = \text{denoiser}(x_{k+1}, \sqrt{\lambda/\mu}). \quad (11)$$

The Equations (9) and (11) suggest that the fidelity term and regularization term have been separated successfully and the regularization term corresponds only to the subproblem of image denoising. The result is such that we can integrate the two trained sets of denoisers to the model-based optimization method to solve different kinds of image restoration problems.

## 4. CNN Denoising Model

**4.1. Key Technology in CNN.** Due to strong ability of CNN to excavate the image characteristics, we have reason to believe that better results will be achieved when CNN is used to remove the image noise. However, there are many problems when directly classic CNN structure (e.g., LeNet, AlexNet, and ZF-net) is used to make image denoising: firstly, how to select the activation function; secondly, if the pooling layer is added to the network structure, the image after the network can be compressed into very small and lot of information will be lost, which leads to more complexity of the image restoration, so how to increase the receptive field while not changing the image size; thirdly, there are many parameters in the whole network; it takes a large amount of time to train so many parameters normally, so to speed up the whole training process. In the following text, we will give details about the network design.

**4.1.1. Selection of Activation Function.** The introduction of activation function is to add the nonlinear factor. Although the sigmoid function has been successfully applied to many network structures, it also has its own drawbacks as found in recent years, which results in gradual disappearing of gradient flowing to this layer of network and greatly reduces the training speed. The other kind is called ReLU activation function (i.e., rectifier linear units), and it just solves this problem, whose representation is as follows:

$$f(x) = \max(0, x) \quad (12)$$

After the input is less than or equal to 0, the output is 0, which is equivalent to the building of one sparse matrix. This feature can remove the redundancy in the data and retain the characteristics of the data. In the continuous network computation process, it changes as to how it tries to represent the data characteristics with one matrix most of 0. Due to the sparsity, this method runs fast and effectively.

**4.1.2. Dilated Convolution.** The existence of the pooling layer leads to image shrinking and loss of lot of information. The convolution method of dilated convolution is introduced. The basic idea of the dilated convolution is that while keeping the image size unchanged, the receptive field does not become smaller, and huge computation amount is not increased. Specifically, the convolution kernels originally densely arranged become somewhat fluffy, the number of points to be calculated in the convolution kernels is not changed, and the spare positions are all filled with 0. The receptive field can keep increasing while the part of the convolution kernels that really needs to be calculated remains unchanged, and it is always  $3 \times 3$ .

**4.1.3. Batch Normalization.** After the ReLU activation function above is adopted, the problem of saturation gradient disappearing is solved, but many factors slowing the training speed still exist in real training. In the network training, a continuous change of different layers of parameters can cause change in each layer of input distribution of the network due to the backpropagation. On the other hand, the training has

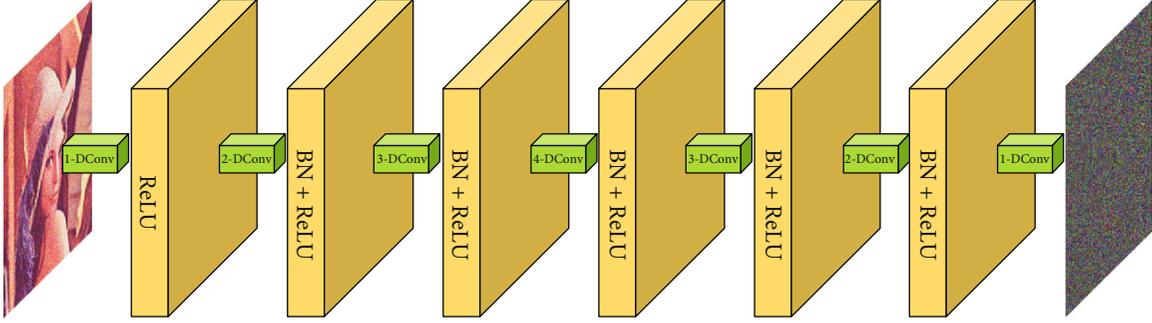


FIGURE 1: Denoising network structure, where  $s$ -DConv means expanded convolution,  $s = 1, 2, 3,$  and  $4$ , BN means batch normalization, and ReLU means activation function.

TABLE 1: Structure parameters of the network model.

	The layers of operation	Convolution kernel description	Step length	Expansion factor	Fill factor	Output size
The input layer	Degraded image					$35 \times 35$
Hidden layers	1	1-DConv+ReLU	$64 \times 1 \times 3 \times 3$	1	1	$64 \times 35 \times 35$
	2	2-DConv+BN+ReLU	$64 \times 64 \times 3 \times 3$	1	2	$64 \times 35 \times 35$
	3	3-DConv+BN+ReLU	$128 \times 64 \times 3 \times 3$	1	3	$128 \times 35 \times 35$
	4	4-DConv+BN+ReLU	$128 \times 128 \times 3 \times 3$	1	4	$128 \times 35 \times 35$
	5	3-DConv+BN+ReLU	$64 \times 128 \times 3 \times 3$	1	3	$64 \times 35 \times 35$
	6	2-DConv+BN+ReLU	$64 \times 64 \times 3 \times 3$	1	2	$64 \times 35 \times 35$
Output layer	1-DConv	$1 \times 64 \times 3 \times 3$	1	1	1	$35 \times 35$

to adapt to such change again, which reduces the training efficiency. The batch normalization (BN) method is used in this paper to solve the problem. After the mean value and variance of the whole data are obtained, the data are normalized so that the training of each layer of the network is no longer suitable for input change, which greatly improves the training efficiency.

**4.1.4. Residual Learning.** There are two kinds of learning methods in the neural network. One is to directly learn the mapping from the image  $y$  including noise to the potential clear image  $x$ , and the other is to firstly learn the noise in the image and then solve indirectly the potential clear image. The second learning method is called residual learning. If one mapping is close to the identity mapping, the use of residual network can make the optimization process easier. Clearly, the process of image denoising is closer to one identity mapping, particularly when the noise level is lower. So the residual learning is added to our network model, which can accelerate and stabilize the training process together with BN mentioned above and improve the denoising ability of the network model.

**4.2. Proposal of Network Model.** The practice proves that the CNN structure is more useful for image feature extraction and is powerful in performance. In particular, parallel computation can be based on GPU during the network training, which greatly improves the training efficiency. Based on such trend, we use CNN to restore the image. And ReLU activa-

tion function, dilated convolution, batch normalization, and residual learning above are applied to the network model to get better image restoration ability. The network model in this paper is shown in Figure 1.

As shown in Table 1, between the input layer and the output layer are collectively called hidden layers. The traditional  $3 \times 3$  model is used for single convolution kernel of each layer, and the convolution step is 1. Meanwhile, in order to solve the boundary effect, the zero-filling method is adopted. In order to improve the training efficiency, the practice in this paper is to cut the image to a size of  $35 \times 35$ , so both the input layer and the output layer have images whose size is  $35 \times 35$  individually.

## 5. Experiment

**5.1. Image Denoising.** Since residual learning is added in the network, the residual function of the training model is

$$\ell(\Theta) = \frac{1}{2N} \sum_{i=1}^N \|f(y_i; \Theta) - (y_i - x_i)\|_F^2, \quad (13)$$

where  $\{(x_i, y_i)\}_{i=1}^N$  represents  $N$  pairs of clear and noise images;  $f$  is the output of network model, the difference between the predicated value and the actual value can be represented by  $f(y_i; \Theta) - (y_i - x_i)$ ; and  $\Theta$  is the parameter to be trained in the model.  $N$  is the number of minibatch input images.

TABLE 2: Average PSNR/(dB) on BSD68 [4] dataset (gray) of different denoising algorithms.

Noise level	BM3D	WNNM	EPLL	TNRD	MLP	DnCNN	FFDNet	Proposed
Sigma10	33.32	33.41	33.39	—	33.49	33.72	33.76	33.74
Sigma25	28.57	28.82	28.74	28.78	28.94	29.14	29.17	29.15
Sigma35	27.09	27.39	27.35	—	27.48	27.63	27.65	27.66
Sigma50	25.62	25.86	25.83	25.80	26.01	26.15	26.16	26.19

TABLE 3: Average PSNR/(dB) on CBSD68 dataset (colorful) of CBM3D and proposed.

Noise level	CBM3D	DnCNN	FFDNet	Proposed
Sigma10	35.90	36.03	36.10	36.06
Sigma20	31.89	32.25	32.29	32.27
Sigma30	29.72	30.19	30.21	30.22
Sigma40	28.08	28.78	28.82	28.85
Sigma50	27.37	27.75	27.79	27.86

After determining the loss function, it is the training network. The training data sets adopted in this paper are 400 images in Berkeley segmentation dataset [34], 400 images in ImageNet database [35], and 4744 images in Waterloo Exploration Database [36]. We cut all the images to pieces of  $35 \times 35$  and select randomly  $256 \times 4000$  pieces for training. The solver adopts Adam (adaptive moment estimation), uses the default hyperparameters, and the minibatch size is selected to be 256. In order to handle different levels of noise, we train a series of denoising network models (also called denoisers) targeting different noise levels. The noise level  $\sigma$  is from 0 to 50, the step is 2, and there are 25 denoisers in total. Our experiments have been implemented in MATLAB R2017b with MatConvNet package [37], running on PC with Intel Core i7-7700HQ CPU, 2.80 GHz, NVIDIA GeForce GTX 1060 GPU. The operating system is Windows 10. It takes about six days to train two groups of denoisers (used separately for gray and color images).

In order to better explain the algorithm performance in this paper, two model-based optimization methods (i.e., BM3D [38] and WNNM [39]) and five discriminative learning methods (i.e., TNRD [34], MLP [12], EPLL [5], DnCNN [40], and FFDNet [41]) are selected for comparison. On the experimental results of color image denoising, we also choose the CBM3D algorithm, DnCNN, and FFDNet models to compare with the method proposed in the text. The experimental data are shown in Tables 2 and 3; our method achieves the level of the best denoising model FFDNet in gray and color image denoising, and with the increase of noise level, the advantages of our method become more obvious. Moreover, the number of our network layer is far less than the FFDNet model, which has the advantage of easier network training and higher efficiency. In terms of computational efficiency, BM3D and WNNM algorithms use CPU operations, while TNRD, MLP, and our algorithms use GPU parallel computing. Since our CNN structure was originally designed for image denoising, the algorithm has

extremely high operational efficiency in the denoising sub-problem. Whether it is gray or color image denoising, usually the operation time is kept within 0.1 s. Generally speaking, both the gray and color image denoising of the algorithm proposed in this paper are excellent. In particular, we have showcased part of the experimental results in Figures 2 and 3 so as to feel vividly the performance of each algorithm. From the local enlarged image, we can find that our method does not make the denoised image become too smooth, and the restoration of detail texture is closest to the original image.

**5.2. Image Deblurring.** Generally, the production of blurred image can be modeled as one blurred kernel applied to the original clear image and added with white Gaussian noise with the level of  $\sigma$ . In order to evaluate the method proposed in this paper, three kinds of blurred kernels are adopted in the experiment: one is the normal Gaussian blurred kernel with the standard deviation of 1.6 and size of  $25 \times 25$ ; the other two are the first two kinds in eight kinds of actual blurred kernels proposed in the literature [42] (the first kernel and the second kernel size are  $19 \times 19$  and  $17 \times 17$ , respectively). In addition, the normal three levels of white Gaussian noises are added, which are  $\sigma = 2$ ,  $\sigma = 2.55$ , and  $\sigma = 7.65$ , respectively.

According to the Gaussian denoiser trained in the last section, we design the following image deblurring experiment process as shown in Algorithm 1.

During the iterative process above, the noise level of the denoiser is one gradual decreasing process. The number of iteration times set by us is 30. One geometric progression of 30 numbers with the scope of 50 to 0 (in descending order) is formed, and then, 25 denoisers are mapped to 30 according to the principle of proximity (the denoisers have repetition), so the new model is not loaded in each time of iteration. Besides, since there is inverse to solve matrix in Equation (9), we use FFT to speed up the solution.

The method proposed is compared with four other kinds of algorithms. IDDBM3D, NCSR, and EPLL [5] are model-based optimization methods, and MLP is a discriminative learning method. The test data sets are Set3G and Set3C (including gray and color images separately). As shown in Table 4, the method proposed in any blurring condition performs well, particularly in deblurring of color images. Among them, IDDBM3D, NCSR, and MLP algorithms tend to make the image edge smoother and prone to color artifacts. In contrast, our method can better restore the image clarity and naturalness. As shown in Figures 4 and 5, the deblurring effect of each algorithm can be seen directly. At the same time, we also

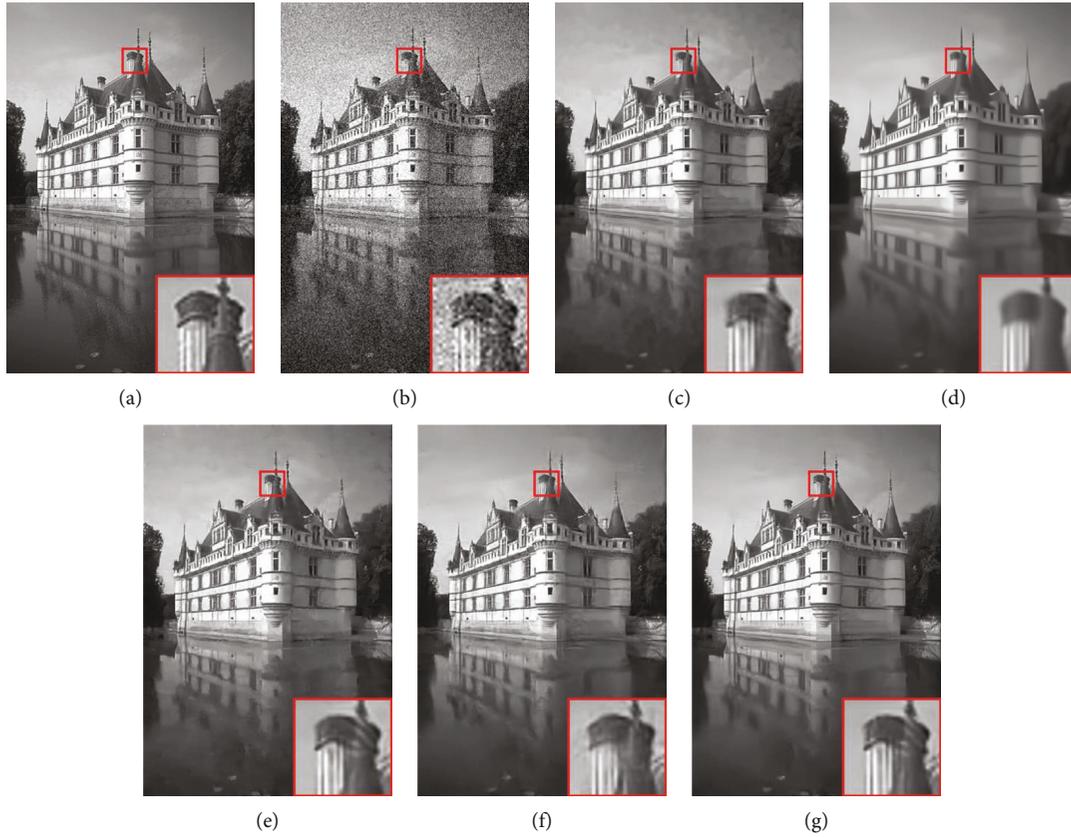


FIGURE 2: Restoration of different algorithms for the gray image denoising ( $\sigma = 25$ ). (a) Original image. (b) Noise image. (c) BM3D, PSNR = 29.53 dB, CPU: 3.95 s. (d) WNNM, PSNR = 28.91 dB, CPU: 648.72 s. (e) TNRD, PSNR = 29.63 dB, GPU: 4.67 s. (f) MLP, PSNR = 29.73 dB, GPU: 13.38 s. (g) Proposed, PSNR = 30.07 dB, GPU: 0.02 s.

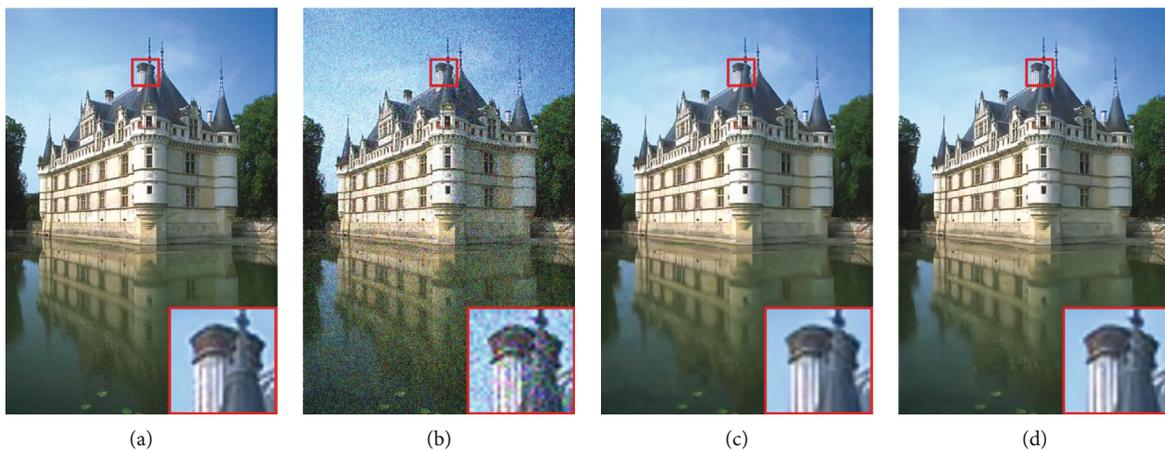


FIGURE 3: Restoration of CBM3D and proposed for the color image denoising ( $\sigma = 25$ ). (a) Original image. (b) Noise image. (c) CBM3D, PSNR = 31.69 dB, CPU: 5.53 s. (d) Proposed, PSNR = 32.08 dB, GPU: 0.05 s.

recorded the running time of different algorithms. It can be seen that the algorithm proposed is the most competitive in terms of computational efficiency.

**5.3. Single Image Super-Resolution.** In reconstruction of single image super-resolution (SISR), it can be modeled as fuzzy

treatment of a high-resolution image and then low-resolution image is obtained after downsampling. Based on the obtained model of the low-resolution image, the super-resolution reconstruction of a single image can be divided into two subproblems [23, 43], i.e., iterative update back projection solution and denoising. Combined with the HQS

1. Select one particular blurred kernel to apply to the original clear image  $x$ , and add the Gaussian noise of particular level  $\sigma$  to produce the blurred image  $y$
2. Determine the parameter  $\lambda$  according to the noise level  $\sigma$
3. for  $i=1:n$  ( $n$  is the set iteration times)
4. Determine the relevant level of denoiser according to the iterative value so as to determine the value of parameter  $\mu$
5. Solve the Eqn. (9) with FFT and get the result  $z$
6. Load relevant level of denoiser and take the  $z$  obtained from last step as the model input to get the model output residual, get the difference value  $z = z - \text{residual}$  and update the value  $z$
- 7.end
- 8.Calculate the evaluation data PSNR.

ALGORITHM 1. Image deblurring

TABLE 4: Experiment results of different algorithms for image deblurring (PSNR/(dB)).

Dataset	$\sigma$	IDDBM3D	NCSR	MLP	Proposed
Gaussian blur with standard deviation 1.6					
Set3G	2	30.72	30.78	30.53	31.03
Set3C		29.34	29.53	29.81	30.62
Dataset	$\sigma$	EPLL		Proposed	
Kernel 1 ( $19 \times 19$ )					
Set3G	2.55	28.89		33.54	
Set3C		29.00		34.26	
Set3G	7.65	26.96		29.81	
Set3C		23.47		29.94	
Dataset	$\sigma$	EPLL		Proposed	
Kernel 2 ( $17 \times 17$ )					
Set3G	2.55	30.98		33.33	
Set3C		28.24		33.90	
Set3G	7.65	26.65		29.47	
Set3C		22.82		29.49	

method mentioned above, as shown in Equation (14), we can use the iterative back projection method to solve the Equation (9):

$$x_{k+1} = x_k - \alpha(y - x_k \downarrow_{sf}) \uparrow_{sf}, \quad (14)$$

where  $\downarrow_{sf}$  represents the downsampling with the scale factor of  $sf$ ,  $\uparrow_{sf}$  represents the bicubic interpolation with the scale factor of  $sf$ , and  $\alpha$  represents the iterative step. Similarly, the denoiser model can be used for solution.

We have designed the following single image super-resolution reconstruction process as shown in Algorithm 2.

The  $n$  in the above process is the number of internal iterations in order to speed up the convergence of iteration. The  $n$  set in this paper is 5,  $m$  is 30, the step length  $\alpha$  is 1.75, and the noise level of the denoiser attenuates from the exponent  $12 \times sf$  to  $sf$ .

We also select six kinds of algorithms for comparison: one is the model-based optimization method NCSR, another

is the method based on denoising prior SRBM3D [40], and the other four are discriminative learning methods (i.e., SRCNN, VDSR [44], LapSRN [45], and SPMSR [46]). Three different  $\downarrow_{sf}$  are adopted in the experiment, i.e., bicubic downsampling when  $sf$  is 2 and 3 [14, 47], and Gaussian blurred kernel with the standard deviation of 1.6, size of  $7 \times 7$  and  $sf$  of 3 [11]. The experiment data are shown in Table 5. The data sets are Set5 (including five color images) and Set14 (including 14 gray and color images) [48]. In terms of data, the method in this paper does not reach the best performance in every aspect, but its superiority can be fully represented from another perspective, specifically.

When  $\downarrow_{sf}$  is bicubic, VDSR and LapSRN can be called the state-of-the-art algorithm. The algorithm performance in this paper can be close to these two methods. When  $\downarrow_{sf}$  is in Gaussian condition, as suggested in Figure 6, the performance of three kinds of discriminative learning methods are greatly limited since no such model is trained in advance, so the whole model has to be trained again as to how to keep good reconstruction results. According to Equation (14) and the experimental data, the method proposed in this paper is able to adapt to different kinds of  $\downarrow_{sf}$  models when the whole model does not need to be trained again and the performance can be close to the most advanced algorithm level. The results show that the proposed image restoration method based on depth CNN denoising prior can perform super-resolution on degraded images only by adjusting fuzzy kernel and scale factor without training, while SRCNN, VDSR, and LapSRN need additional training to deal with these situations. This can fully demonstrate that the proposed method is more flexible than other discriminative learning methods. In addition, the algorithm proposed in this paper is still the fastest.

## 6. Conclusion

In this paper, a series of Gaussian denoisers are obtained through CNN learning, and the denoisers are integrated as modules into the model-based optimization method by combining variable splitting techniques (i.e., the fidelity term and regularization term are separated in the original problem), which greatly improves the flexibility of discriminative learning method in solving different image restoration problems.

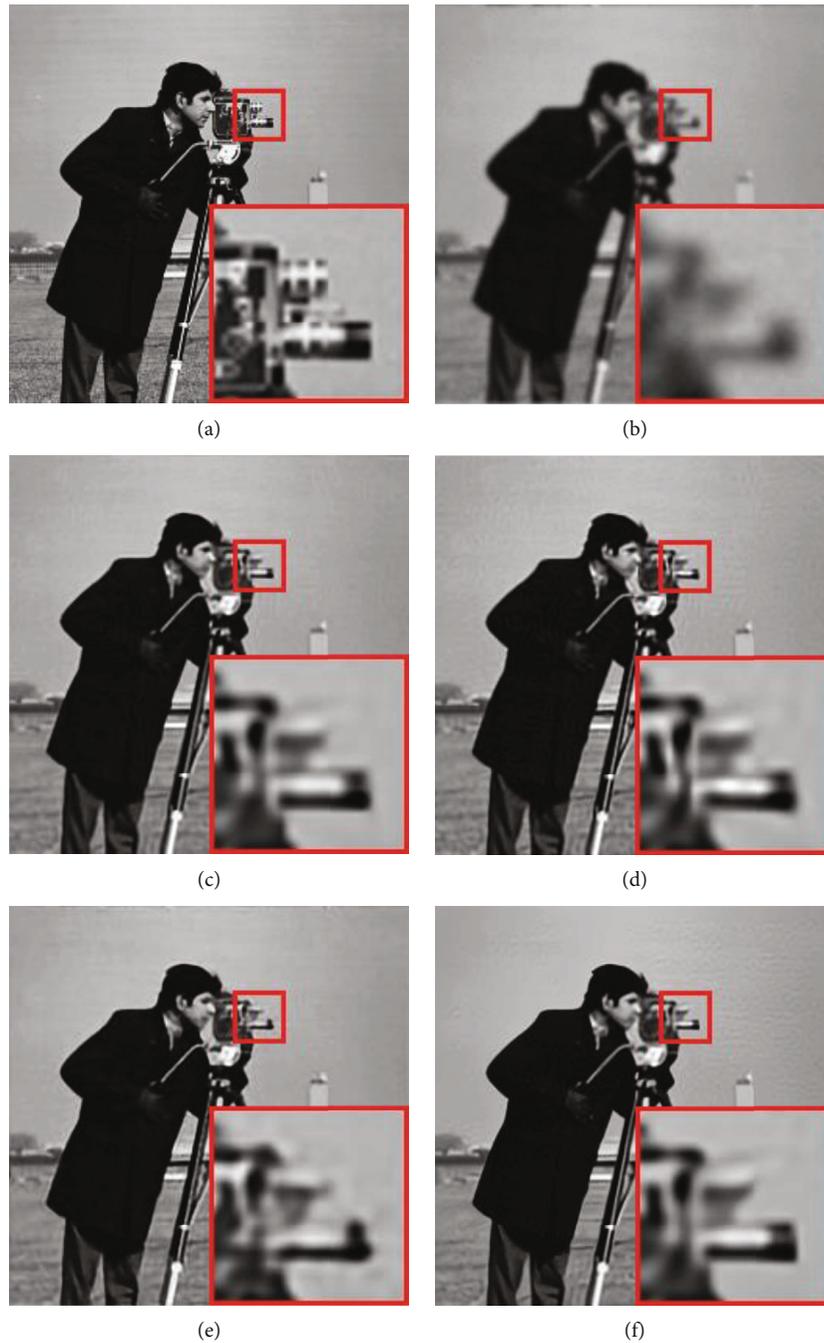


FIGURE 4: Restoration of different algorithms for gray image deblurring (Gaussian blur with standard deviation 1.6,  $\sigma = 2$ ). (a) Original image. (b) Blurred image. (c) IDDBM3D, PSNR = 27.68 dB, CPU: 168.41 s. (d) NCSR, PSNR = 27.95 dB, CPU: 239.49 s. (e) MLP, PSNR = 27.67 dB, GPU: 0.86 s. (f) Proposed, PSNR = 28.12 dB, GPU: 0.69 s.

In order to better explain the effect of the method proposed in this paper, several kinds of most advanced algorithms are selected for comparison. The experiment results show that the denoisers obtained with CNN learning have good image prior knowledge, which can solve well other image restoration problems when it is applied to model-based optimization method, i.e., image deblurring and single image super-resolution. Moreover, compared with the

most advanced algorithms, the proposed algorithm is also competitive enough.

Although the proposed method integrates the advantages of model-based optimization method and discriminant learning method, there are still many areas worth studying. For example, this paper is for image nonblind deblurring; i.e., the blurred kernel is known, and the blind deblurring can be further studied; during denoiser training, the number

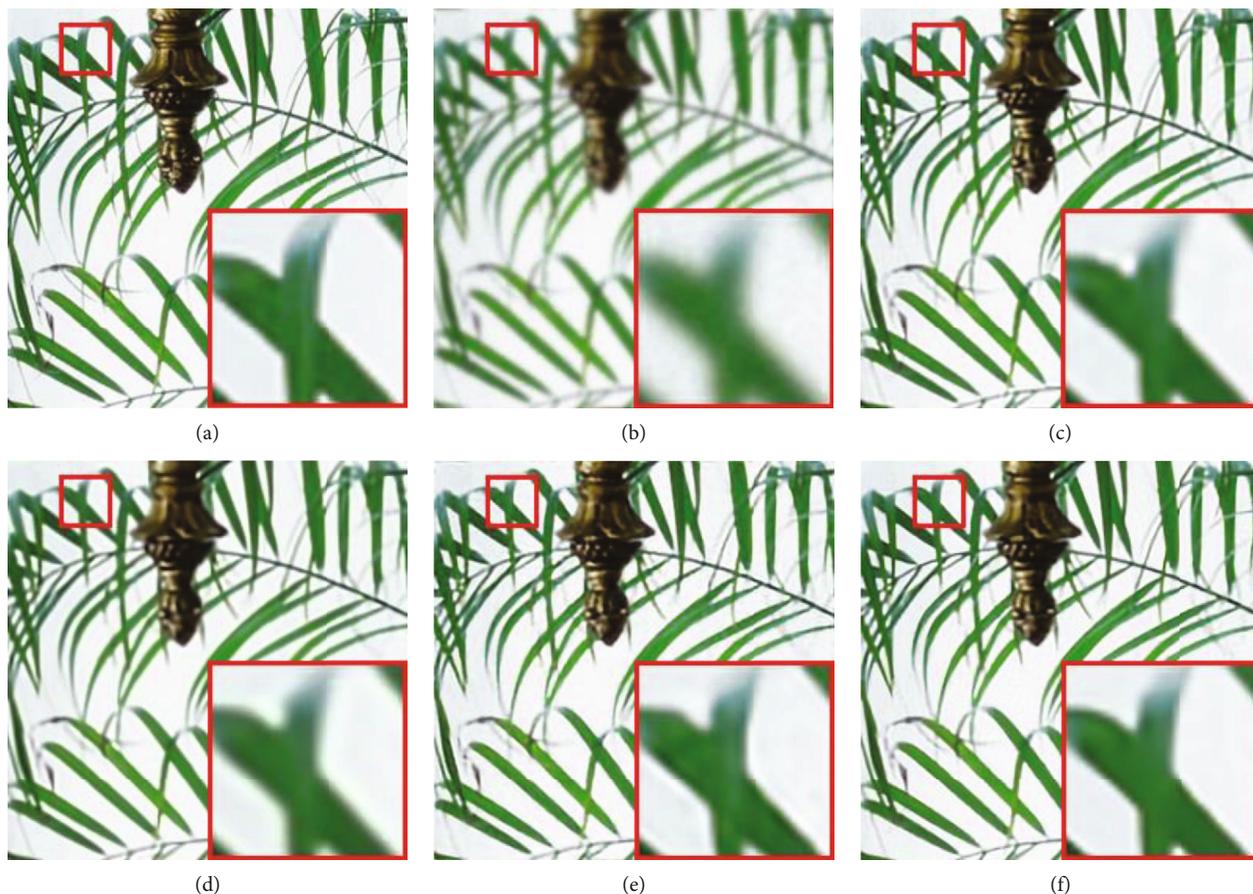


FIGURE 5: Restoration of different algorithms for color image deblurring (Gaussian blur with standard deviation 1.6,  $\sigma = 2$ ). (a) Original image. (b) Blurred image. (c) IDDBM3D, PSNR = 28.70 dB, CPU: 173.52 s. (d) NCSR, PSNR = 29.08 dB, CPU: 257.18 s. (e) MLP, PSNR = 29.40 dB, GPU: 1.52 s. (f) Proposed, PSNR = 29.78 dB, GPU: 0.71 s.

1. Read the high-resolution image (HR), cut the size to integral multiples of the scaling factor of  $sf$ , so that the down-sampling is changed to low-resolution image (LR);
2. Change the LR of RGB space to YCbCr space to get the image LR\_ycbcr, and take the LR\_y of component Y as the value  $y$  in Eqn. (14);
3. Restore the HR\_bic of the original image size with Bicubic interpolation for LR
4. Change the HR\_bic to YCbCr space to get the image HR\_bic\_ycbcr, and take the component Y as the iterative output, i.e.,  $x_k$  initialization of in Eqn. (14);
5. for  $i=1:m$  ( $m$  is the number of iterations)
6. for  $j=1:n$
7. Execute the Eqn. (14) and continue to update the  $x_k$  to  $x_{k+1}$
8. end
9. Load relevant level of denoiser according to iterative value  $i$  and take the obtained  $x_{k+1}$  as the model input
10. The model output is residual and get the final output after making difference
11. end
12. Take the output as the component Y of HR\_bic\_ycbcr and then convert it to RGB space to get the final reconstruction result
13. Calculate the evaluation data PSNR.

ALGORITHM 2. Single image super-resolution.

of denoisers and the iterations is worthy of study; CNN is used for learning in this paper and it is necessary to further study and improve as to whether other deep learning models

can achieve better results. In addition, it is also worth noting that there are various reasons for the degradation of image quality when the camera actually takes pictures. If only the

TABLE 5: Average PSNR/(dB) of different SISR methods on data sets Set5 and Set14.

Dataset	$sf$	NCSR	SPMSR	SRBM3D	SRCNN	VDSR	LapSRN	Proposed
				Kernel Bicubic				
Set5	2	—	36.43	36.47	36.66	37.56	37.61	37.43
Set14		—	32.35	32.38	32.45	33.02	33.10	32.88
Set5	3	—	32.49	32.54	32.65	33.67	33.74	33.39
Set14		—	28.98	29.09	29.24	29.77	29.83	29.61
				Kernel Gaussian				
Set5	3	33.03	32.86	32.95	30.50	30.54	30.59	33.38
Set14		29.27	29.03	29.18	27.78	27.80	27.85	29.63

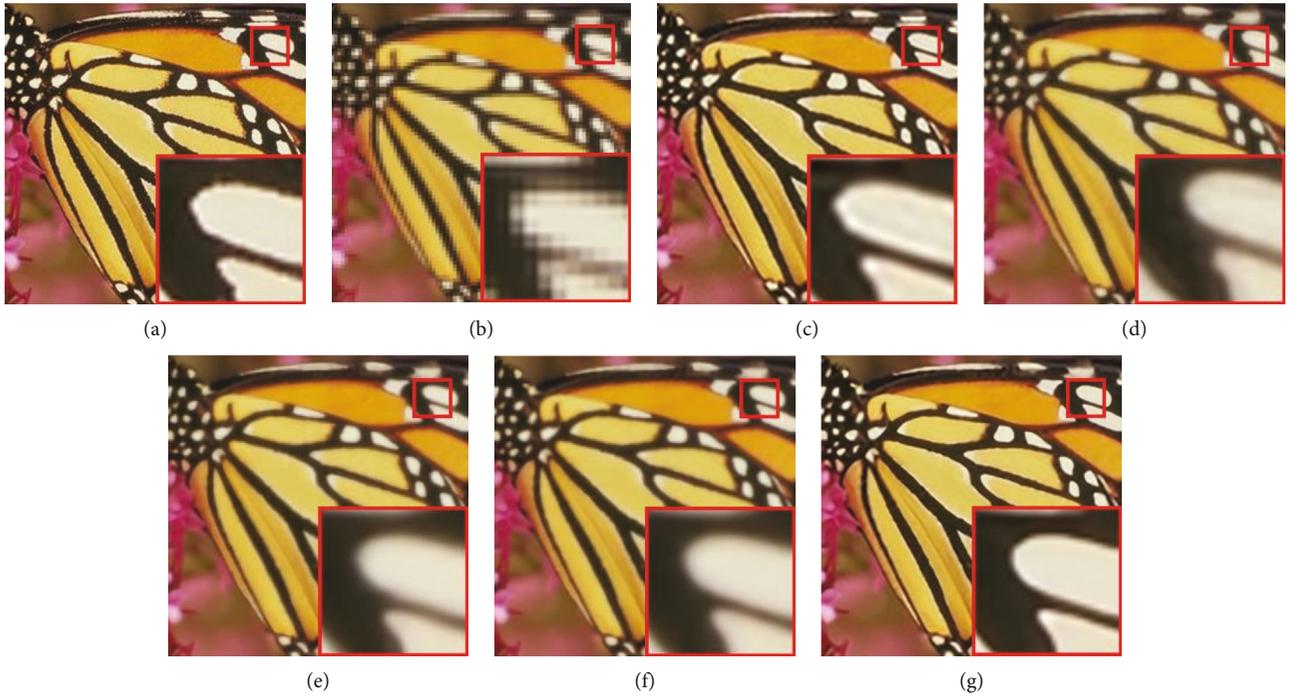


FIGURE 6: Reconstruction of different algorithms for SISR (Gaussian blur with standard deviation 1.6,  $sf = 3$ ). (a) Original image. (b) Low resolution image. (c) NCSR, PSNR = 28.09 dB, CPU: 350.04 s. (d) SRCNN, PSNR = 23.93 dB, CPU: 5.02 s. (e) VDSR, PSNR = 24.73 dB, CPU: 2.12 s. (f) LapSRN, PSNR = 24.60 dB, GPU: 1.18 s. (g) Proposed, PSNR = 29.32 dB, GPU: 0.72 s.

Gaussian denoising model is used, it is not enough to restore the details of low-quality images in all cases. The future work of this paper also focuses on this aspect, so that the denoising model can be closer to real image degradation model.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Nature Science Foundation of China (Grant No. 61471304), and we wish to acknowledge them for their support. The authors also thank Southwest Jiaotong University Photoelectric Engineering Institute for their kind support in the experiment.

## References

- [1] H. C. Andrews and B. R. Hunt, *Digital image restoration*, vol. 1 of Prentice-Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, 1977.
- [2] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications*, CRC press, 2016.

- [3] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.
- [4] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
- [5] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *2011 International Conference on Computer Vision*, pp. 479–486, Barcelona, Spain, November 2011.
- [6] A. Barbu, "Training an active random field for real-time image denoising," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2451–2462, 2009.
- [7] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [8] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2774–2781, Columbus, OH, USA, June 2014.
- [9] J. Sun and M. F. Tappen, "Separable markov random field model and its applications in low level vision," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 402–407, 2013.
- [10] M. F. Tappen, "Utilizing variational optimization to learn markov random fields," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.
- [11] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [12] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: can plain neural networks compete with BM3D?," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399, Providence, RI, USA, June 2012.
- [13] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Advances in Neural Information Processing Systems*, pp. 1790–1798, Montreal Convention Center, Montreal, Canada, 2014.
- [14] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
- [15] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012.
- [16] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1067–1074, Portland, OR, USA, June 2013.
- [17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [18] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, 1995.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, 2012.
- [20] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456, Lille, France, 2015.
- [21] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *International Conference for Learning Representations*, San Diego, CA, USA, 2015.
- [22] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, <https://arxiv.org/abs/1511.07122>.
- [23] K. Egiazarian and V. Katkovnik, "Single image super-resolution via BM3D sparse coding," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2849–2853, Nice, France, August 2015.
- [24] V. K. Danielyan and K. Egiazarian, "Image deblurring by augmented lagrangian with BM3D frame prior," in *Workshop on Information Theoretic Methods in Science and Engineering*, pp. 16–18, Tampere, Finland, 2010.
- [25] S. V. Venkatakrisnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, Austin, TX, USA, December 2013.
- [26] S. Li, B. Qin, J. Xiao, Q. Liu, Y. Wang, and D. Liang, "Multi-channel and multi-model-based autoencoding prior for grayscale image restoration," *IEEE Transactions on Image Processing*, vol. 29, pp. 142–156, 2020.
- [27] K. Cheng, J. Du, H. Zhou, D. Zhao, and H. Qin, "Image super-resolution based on half quadratic splitting," *Infrared Physics & Technology*, vol. 105, p. 103193, 2020.
- [28] A. Brifman, Y. Romano, and M. Elad, "Turning a denoiser into a super-resolver using plug and play priors," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1404–1408, Phoenix, AZ, USA, September 2016.
- [29] S. H. Chan, X. Wang, and O. A. Elgandy, "Plug-and-play ADMM for image restoration: fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2017.
- [30] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could regularization by denoising (RED)," 2016, <https://arxiv.org/abs/1611.02862>.
- [31] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 96–108, 2016.
- [32] S. Sreehari, S. Venkatakrisnan, B. Wohlberg, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," 2015, <https://arxiv.org/abs/1512.07331>.
- [33] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. Figueiredo, "Image restoration and reconstruction using variable splitting and class-adapted image priors," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3518–3522, Phoenix, AZ, USA, September 2016.
- [34] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.
- [36] K. Ma, Z. Duanmu, Q. Wu et al., "Waterloo exploration database: new challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.

- [37] A. Vedaldi and K. Lenc, "MatConvNet: convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia - MM '15*, pp. 689–692, Brisbane, Australia, 2015.
- [38] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [39] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862–2869, Columbus, OH, USA, June 2014.
- [40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [41] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: toward a fast and flexible solution for CNN based image denoising," *CoRR*, 2017, <https://arxiv.org/abs/1710.04026>.
- [42] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, Miami, FL, USA, June 2009.
- [43] M. Irani and S. Peleg, "Motion analysis for image enhancement: resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.
- [44] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, Las Vegas, NV, USA, June 2016.
- [45] W. S. Lai, J. B. Huang, and N. Ahuja, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2599–2613, 2017.
- [46] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569–2582, 2014.
- [47] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Computer Vision – ECCV 2014*, pp. 49–64, Springer, 2014.
- [48] R. Timofte, V. de Smet, and L. van Gool, "A+: adjusted anchored neighborhood regression for fast super-resolution," in *Computer Vision – ACCV 2014*, pp. 111–126, Springer, 2015.