ISSN 1674-1056
http://cpb.iphy.ac.cn
http://iopscience.iop.org/cpb
中国物理 B
Chinese Physics
中国物理 Chinese Physics B

An image compressed sensing algorithm based on adaptive nonlinear network

Yuan Guo(郭媛), Wei Chen(陈炜), Shi-Wei Jing(敬世伟)
Citation: Chin. Phys. B . 2020, 29(5): 054203 . **doi:** 10.1088/1674-1056/ab7b4e
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Compressed ghost imaging based on differential speckle patterns

Le Wang(王乐), Shengmei Zhao(赵生妹)
Chin. Phys. B . 2020, 29(2): 024204 . **doi:** 10.1088/1674-1056/ab671a

Terahertz two-pixel imaging based on complementary compressive sensing

Yuye Wang(王与烨), Yuchen Ren(任宇琛), Linyu Chen(陈霖宇), Ci Song(宋词), Changzhao Li(李长昭), Chao Zhang(张超), Degang Xu(徐德刚), Jianquan Yao(姚建铨)
Chin. Phys. B . 2018, 27(11): 114204 . **doi:** 10.1088/1674-1056/27/11/114204

Multiple-image encryption by two-step phase-shifting interferometry and spatial multiplexing of smooth compressed signal

Xue Zhang(张学), Xiangfeng Meng(孟祥锋), Yurong Wang(王玉荣), Xiulun Yang(杨修伦), Yongkai Yin(殷永凯)
Chin. Phys. B . 2018, 27(7): 074205 . **doi:** 10.1088/1674-1056/27/7/074205

An iterative virtual projection method to improve the reconstruction performance for ill-posed emission tomographic problems

Liu Hua-Wei, Zheng Shu, Zhou Huai-Chun
Chin. Phys. B . 2015, 24(10): 104204 . **doi:** 10.1088/1674-1056/24/10/104204

An image compressed sensing algorithm based on adaptive nonlinear network*

Yuan Guo(郭媛), Wei Chen(陈炜)†, and Shi-Wei Jing(敬世伟)

School of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, *China*

(Received 1 November 2019; revised manuscript received 1 February 2020; accepted manuscript online 1 March 2020)

Traditional compressed sensing algorithm is used to reconstruct images by iteratively optimizing a small number of measured values. The computation is complex and the reconstruction time is long. The deep learning-based compressed sensing algorithm can greatly shorten the reconstruction time, but the algorithm emphasis is placed on reconstructing the network part mostly. The random measurement matrix cannot measure the image features well, which leads the reconstructed image quality to be improved limitedly. Two kinds of networks are proposed for solving this problem. The first one is ReconNet's improved network IReconNet, which replaces the traditional linear random measurement matrix with an adaptive nonlinear measurement network. The reconstruction quality and anti-noise performance are greatly improved. Because the measured values extracted by the measurement network also retain the characteristics of image spatial information, the image is reconstructed by bilinear interpolation algorithm (Bilinear) and dilate convolution. Therefore a second network USDCNN is proposed. On the BSD500 dataset, the sampling rates are 0.25, 0.10, 0.04, and 0.01, the average peak signal-noise ratio (PSNR) of USDCNN is 1.62 dB, 1.31 dB, 1.47 dB, and 1.95 dB higher than that of MSRNet. Experiments show the average reconstruction time of USDCNN is 0.2705 s, 0.3671 s, 0.3602 s, and 0.3929 s faster than that of ReconNet. Moreover, there is also a great advantage in anti-noise performance.

Keywords: compressed sensing, deep learning, bilinear interpolation, dilate convolution

PACS: 42.30.Wb, 42.68.Sq **DOI:** 10.1088/1674-1056/ab7b4e

1. Introduction

In order to recover the analog signal without distortion, the conventional Nyquist sampling frequency should be no less than twice the highest frequency in the analog signal spectrum. And the large number of data is not conducive to storage and transmission. In 2006, the compressed sensing theory[1–3] proposed by Candes, *et al.*, which can sample the signal at a much lower sampling frequency than Nyquist and fully reconstruct the original signal with high probability.

The traditional compressed sensing reconstruction method is based on sparse prior knowledge, which essentially solves an underdetermined system of equations ($y = \Phi x$). How to find the optimal solution from this set of underdetermined equations is the key to reconstruction.[4] The main research directions are as follows: (i) Sparse representation. Look for a sparse basis Ψ, on which the sparse matrix projected by the signal x. And the sparse matrix has the fewest number of non-zero elements. (ii) Measurement matrix. Find the measurement matrix Φ, which is unrelated to the sparse basis Ψ, to make the measured value y obtained by dimension reduction retain enough information about the original signal x. (iii) Refactor method. Find a reconstruction method with low reconstruction time and good robustness while the reconstruction quality is ensured. In the reconstruction method, the convex relaxation method,[5,6] the greedy matching pursuit method,[7–10] and the Bayesian method[11–13] are usually used to solve the corresponding sparse coding problem. However, the real image does not accurately satisfy the sparsity in the transform domain, the image reconstructed by the sparse modeling reconstruction algorithm is not high in quality. Moreover it is difficult to realize real-time performance by multiple iterations, which restricts the the development of compression sensing technology.

Some real images do not accurately satisfy the sparsity in the transform domain. By deep learning method, the measured values can be extracted and reconstructed by pure data-driven method, which relaxes the assumptions of the image signal sparsity. Convolutional neural networks and heap noise reduction self-coding networks have the ability to extract high-quality features of images, which can significantly improve the quality of image reconstruction by training and learning. In Ref. [14], Mousavi *et al.* reconstructed images by using the heap noise reduction self-coding model (SDA) and designed two kinds of networks. One is to use a linear reconstruction method and the input is a measured value, and the other is to use a nonlinear method from end to the end and the input is

*Project supported by the National Natural Science Foundation of China (Grant No. 61872204), the Natural Science Fund of Heilongjiang Province, China (Grant No. F2017029), the Scientific Research Project of Heilongjiang Provincial Universities, China (Grant No. 135109236), and the Graduate Research Project, China (Grant No. YJSCX2019042).

†Corresponding author. E-mail: 1010172469@qq.com

© 2020 Chinese Physical Society and IOP Publishing Ltd

the original image. In Ref. [15], a ReconNet was proposed, by which the reconstructed image was obtained from the linear mapping network, and the high quality reconstructed image was also found by two SRCNN models.[16] The reconstruction quality is better than that of the SDA. In Ref. [17], a DR^2-Net was proposed, which was composed of a linear mapping network and four residual network blocks.[18] The reconstruction quality was improved. However, the reconstruction time is longer. Lian *et al.*[19] proposed an MSRNet, which was comprised of a linear mapping network and a multi-scale residual network. The reconstruction quality is better than that of the DR^2-Net, but the reconstruction time is still longer than that of the ReconNet.

In this paper, we propose an IReconNet model. Experiments show the reconstruction quality is better than those of the ReconNet, the DR^2-Net, and MSRNet by using the adaptive nonlinear measurement network. However the measured values obtained by the measurement network still retain the image spatial information. Another model, *i.e.*, the USDCNN, is proposed by improving the reconstruction network. The reconstruction quality of the USDCNN is better than that obtained by using IReconNet with the fully connected layer and the reconstruction time is much shorter also.

2. Compressive sensing

The traditional compressed sensing mainly includes three parts: data sampling, data sparse representation, data reconstruction. When the image is $n = w \times h$, take the dimension of $m \times n$ ($m \ll n$) measurement matrix to sample the image $\boldsymbol{x}$, which has been vectorized into $n \times 1$ dimension,

$$\boldsymbol{y} = \boldsymbol{\Phi x}. \tag{1}$$

Since $m \ll n$, equation (1) is an underdetermined equation. The number of independent variables $\boldsymbol{x}$ is much larger than the number of dependent variables $\boldsymbol{y}$ and the equations have infinite solutions. It is difficult to reconstruct images. The image $\boldsymbol{x}$ must be converted by the image's sparse nature in some transform domains into

$$\boldsymbol{x} = \boldsymbol{\Psi s}. \tag{2}$$

According to Eq. (2), equation (1) can be converted into

$$\boldsymbol{y} = \boldsymbol{\Phi\Psi s}, \tag{3}$$

where $\boldsymbol{s} \in \boldsymbol{R}^{n\times 1}$ is a k-sparse signal, which is a vector composed of sparse coefficients of the original image $\boldsymbol{x}$ under the variation of the sparse basis $\boldsymbol{\Psi} \in \boldsymbol{R}^{n\times n}$. The k-sparse signal means that the vector has k non-zero values ($k \ll n$). Equation (3) can be solved by optimizing the problem of l_0 norm. The number of non-zero elements of $\boldsymbol{s}$ is maximized as follows:

$$\min \|\boldsymbol{s}\|_0, \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi\Psi s}. \tag{4}$$

Equation (4) is an NP problem, generally using l_1 norm to approximate the l_0 norm solution as indicated below:

$$\min \|\boldsymbol{s}\|_1, \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi\Psi s}. \tag{5}$$

The traditional optimization method requires multiple iterations of Eq. (5), which has the disadvantages of large computation and long time and cannot implement the reconstruction in real time. The deep learning method can avoid the sparsity problem in the traditional optimization method, and the learning weight $\boldsymbol{A}$ is trained by the measured value y and the original image x such that $\boldsymbol{x}'$ is close to $\boldsymbol{x}$ as shown below:

$$\boldsymbol{Ay} = \boldsymbol{x}' \approx \boldsymbol{x}. \tag{6}$$

It can be seen from Eq. (8) that the higher the quality of the measured value $\boldsymbol{y}$ with the same number of network layers, the closer to $\boldsymbol{x}$ the $\boldsymbol{x}'$ reconstructed is.

Due to the many and complex features of large images, there are many network layers required for reconstructing the entire image directly through deep learning, in which the reconstruction time is long and the input image size is limited. Dividing the image into blocks and then compressing it can use less network layers to reconstruct high-quality image blocks, and then stitch them into large images without any restrictions on the image size.

3. Network

In this article, the image is divided into blocks (33×33) and four different sampling rates (MR = 0.25, 0.1, 0.04, 0.01) are used. In the IReconNet and USDCNN, the same measurement network is used, the high quality features is measured through the convolutional neural networks. The IReconNet's reconstruction network the same as the ReconNet's, reconstructs the image through the fully connected layer to obtain the approximate solution $\boldsymbol{x}_i'$ of the original image block $\boldsymbol{x}_i$, then learns the residuals of the image blocks $\boldsymbol{x}_i$ and $\boldsymbol{x}_i'$ through two SRCNN network models and obtains high quality reconstructed images. The USDCNN reconstruction network obtains high quality reconstructed images directly through multiple Bilinear and dilate convolution.[20]

3.1. Measurement network

In all of the ReconNet, DR^2-Net, and MSRNet, the random Gaussian measurement matrix $\boldsymbol{\Phi}$ is used to reduce the dimension of the image block $\boldsymbol{x}_i$ to obtain the measured value. The measurement network first increases the dimension through the convolutional layer to obtain sufficient features,

and then reduces the dimension to obtain the required features. Using the BatchNorm[21] to prevent over fitting and speed up training, the Relu activation function[22] improves network expression, and the last layer of the measurement network uses the Sigmoid activation function to map values to 0–1. The original image block $\boldsymbol{x}_i$ is used as the input of the measurement network $F^{\mathrm{s}}(\cdot)$, and the convolution layer weight $\boldsymbol{W}^{\mathrm{s}}$ and the measured value $\boldsymbol{y}_i$ are trained by the Adam[23] method, it can be expressed as

$$\boldsymbol{y}_i = F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}). \tag{7}$$

Taking the sampling rate MR to be 0.25 for example, the measurement network is shown in Fig. 1.

The measurement network in this paper consists of four convolutional layers without using a pooling layer. Figure 1 shows the conversion process of convolution and matrix operations. The $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, and $\boldsymbol{X}_4$ are two-dimensional matrices that are spliced out of the network layer. The $\boldsymbol{\Phi}_1$, $\boldsymbol{\Phi}_2$, $\boldsymbol{\Phi}_3$, and $\boldsymbol{\Phi}_4$ are two-dimensional matrices composed of weights. The $\boldsymbol{Y}_1 = \boldsymbol{\Phi}_1\boldsymbol{X}_1$, $\boldsymbol{Y}_2 = \boldsymbol{\Phi}_2\boldsymbol{X}_2$, $\boldsymbol{Y}_3 = \boldsymbol{\Phi}_3\boldsymbol{X}_3$, and $\boldsymbol{Y}_4 = \boldsymbol{\Phi}_4\boldsymbol{X}_4$ are one-to-one corresponding to the network layers, and the four-dimensional convolution operation on the network layer can be converted into two dimensional matrix operation as shown below:

$$\begin{aligned} \boldsymbol{Y} &= \mathrm{reshape}(\boldsymbol{Y}_4) = \mathrm{reshape}(\boldsymbol{\Phi}_4\boldsymbol{X}_4) \\ &= \mathrm{reshape}(\boldsymbol{\Phi}_4\,[\boldsymbol{\Phi}_3\,[\boldsymbol{\Phi}_2\,[\boldsymbol{\Phi}_1 X_1]]]). \end{aligned} \tag{8}$$

The symbol, reshape$(\cdot)$, means that the vector is transformed into a four-dimensional matrix, and the symbol, $[\cdot]$, means that the matrix is divided and rearranged according to the size and step size of the convolution kernel, respectively.

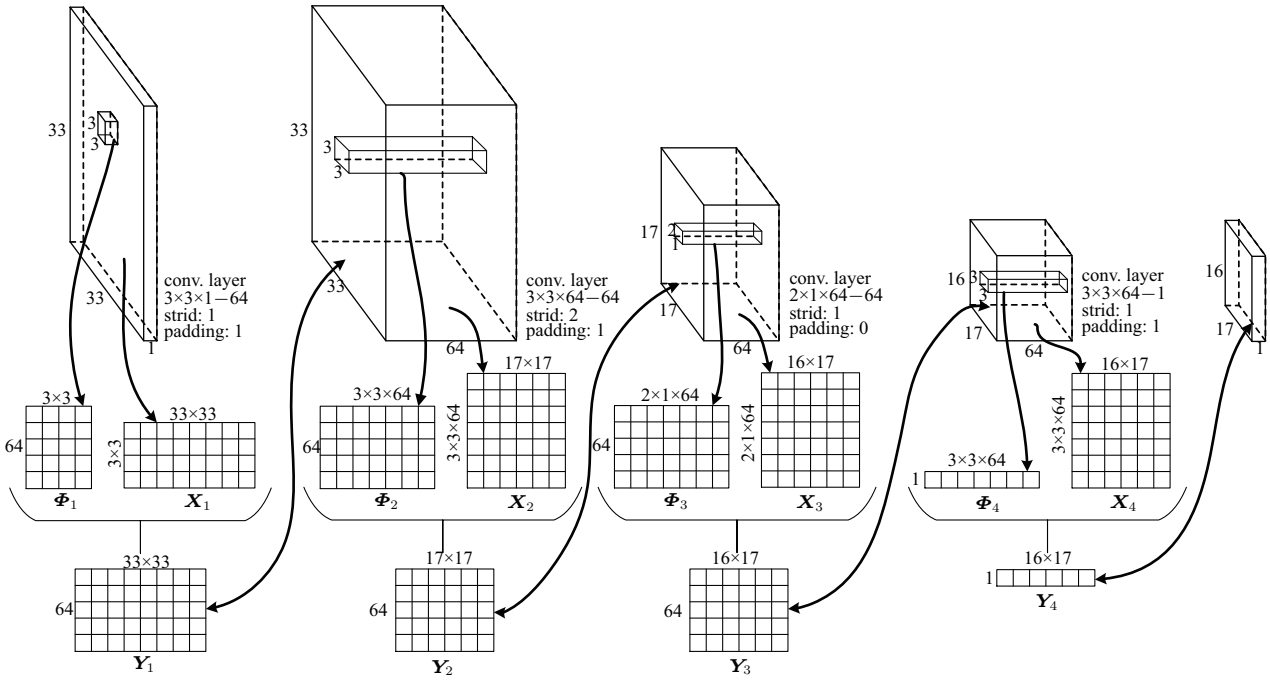


Fig. 1. Measurement network (MR = 0.25).

Since the algorithm based on the deep learning method such as the MSRNet changes the matrix of 33×33 into a vector which is then measured, the measured value does not reveal the image outline. In order to visually show the measurement effect of the random Gaussian measurement matrix on the image, a random Gaussian measurement matrix is used to measure each column of the test image and then spliced. The contrast effect of MR = 0.25 is shown in Fig. 2.

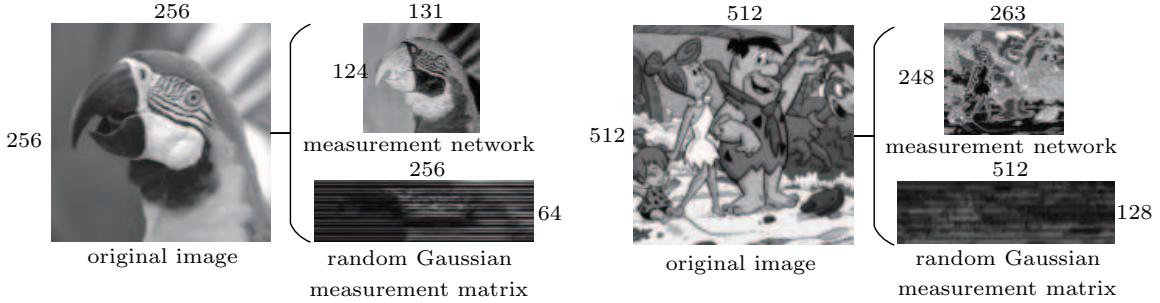


Fig. 2. Comparison of measurement network and random Gaussian measurement matrix.

It can be seen from Fig. 2 that the measurement network can obtain a high quality matrix, and the random Gaussian measurement matrix obtains a poor quality matrix, and many details are lost, and thus leading the reconstruction quality to degrade.

3.2. Refactoring network

3.2.1. IReconNet

(i) Linear generation network

The IReconNet is reconstructed for the image through the fully connected layer. According to the compressed sensing formula $\boldsymbol{y}_i = \boldsymbol{\Phi}\boldsymbol{x}_i$, the measured value $\boldsymbol{y}_i$ is obtained by the linear mapping of the fully connected layer, and the approximate solution $\boldsymbol{x}_i'$. The equation of the image block $\boldsymbol{x}_i$ is obtained as shown in Eq. (9). The loss function uses the mean square error as indicated in Eq. (10).

$$\boldsymbol{x}_i' = F^{\mathrm{f}}(\boldsymbol{y}_i, \boldsymbol{W}^{\mathrm{f}}) = F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}), \tag{9}$$

$$\begin{aligned} L(\boldsymbol{W}^{\mathrm{s}}, \boldsymbol{W}^{\mathrm{f}}) &= \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{x}_i' - \boldsymbol{x}_i \right\|_2^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} \left\| F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}) - \boldsymbol{x}_i \right\|_2^2, \end{aligned} \tag{10}$$

where N is the number of samples, and $\boldsymbol{W}^{\mathrm{f}}$ is the weight of the fully connected layer trained by the Adam method.

(ii) SRCNN model

As shown in Fig. 3, the measured value $\boldsymbol{y}_i$ is first reconstructed by the fully connecting layer $F^{\mathrm{f}}(\cdot)$ to the initial image $\boldsymbol{x}_i'$, and then the residual $\boldsymbol{d}_i$ is obtained by the two SRCNN models $F^{\mathrm{sr}}(\cdot)$. Finally, the reconstructed image $\boldsymbol{x}_i^*$ is obtained from the following equation:

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i' + \boldsymbol{d}_i. \tag{11}$$

Combining Eq. (9) with Eq. (11) and the resulting equation is converted into the following equation:

$$\begin{aligned} \boldsymbol{x}_i^* &= F^{\mathrm{f}}(\boldsymbol{y}_i, \boldsymbol{W}^{\mathrm{f}}) + F^{\mathrm{sr}}(F^{\mathrm{f}}(\boldsymbol{y}_i, \boldsymbol{W}^{\mathrm{f}}), \boldsymbol{W}^{\mathrm{sr}}) \\ &= F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}) \\ &\quad + F^{\mathrm{sr}}(F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}), \boldsymbol{W}^{\mathrm{sr}}), \end{aligned} \tag{12}$$

where $\boldsymbol{W}^{\mathrm{f}}$ and $\boldsymbol{W}^{\mathrm{s}}$ obtained by training Eq. (10) are used as initial values of $\boldsymbol{W}^{\mathrm{f}}$ and $\boldsymbol{W}^{\mathrm{s}}$ in Eq. (12), respectively, and $\boldsymbol{W}^{\mathrm{f}}$, $\boldsymbol{W}^{\mathrm{s}}$ and $\boldsymbol{W}^{\mathrm{sr}}$ in Eq. (12) are updated by using the Adam algorithm. The loss function uses the mean squared error:

$$\begin{aligned} & L(\boldsymbol{W}^{\mathrm{s}}, \boldsymbol{W}^{\mathrm{f}}, \boldsymbol{W}^{\mathrm{sr}}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \boldsymbol{x}_i^* \right\|_2^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{x}_i - F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}) \right. \\ &\quad \left. + F^{\mathrm{sr}}(F^{\mathrm{f}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{f}}), \boldsymbol{W}^{\mathrm{sr}}) \right\|_2^2. \end{aligned} \tag{13}$$

The specific parameters of the SRCNN model are the same as those of ReconNet.

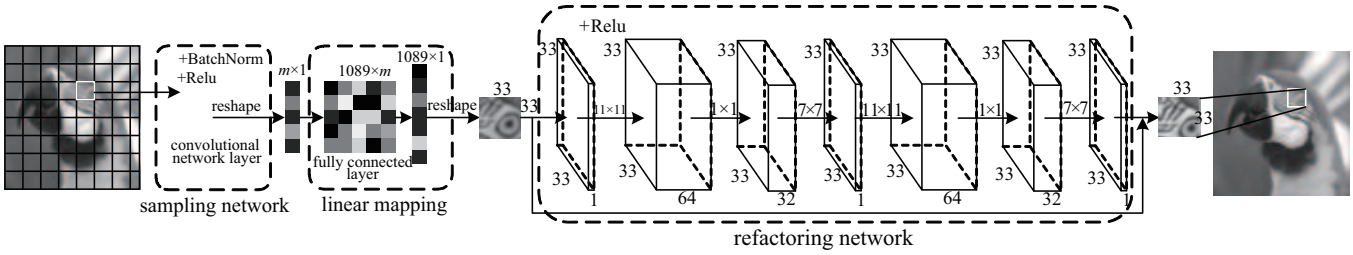


Fig. 3. IReconNet measurement network and reconstruction network.

3.2.2. USDCNN

(I) Dilate convolution

Receptive field is very important in image reconstruction. Large receptive fields can obtain more image features and the reconstruction quality will be higher. In convolutional neural networks, large convolution kernels and pooling layers are generally used to increase the receptive field, and large convolution kernels increase the computational complexity. Although the pooling layer does not increase the computational complexity, it will lose some information, affecting the quality of reconstruction. The dilate convolution is to insert a zero value in the convolution kernel to expand, and there is a large receptive field without adding additional cost, and at the same time, the size of the characteristic map of the output can be kept unchanged. For example, using the dilate factor $d = 3$ to dilate the convolution kernel of 3×3, we obtain a $(2d+1) \times (2d+1)$, which is a convolution kernel of 7×7, in which 9 positions are not zero, the rest are zero. The convolution kernel is shown in Fig. 4.

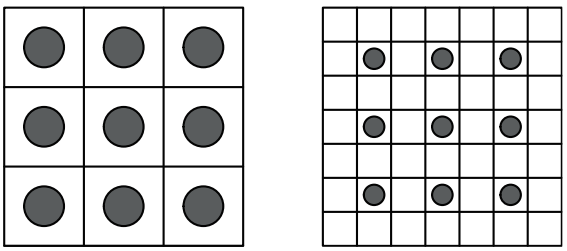


Fig. 4. Dilate convolution.

(II) Up sampling

Using the up sampling method, in this paper we test the nearest neighbor interpolation (Nearest), Bilinear, PixelShuffle,[24] transposed convolution.[25] The Nearest speed is fast but the reconstructed image quality is not high, the Bilinear reconstruction image quality is high but the speed is slower than the Nearest. The PixelShuffle is a matrix of $r^2C \times H \times W$, which is transformed into a matrix of $C \times rH \times rW$ by the Sub-pixel operation. The reconstructed image is high in quality and fast but not all sampling rates are fast, however, these sampling rates are still available. By transposed convolution, the reconstructed image will have checkerboard effect and noise. After comprehensive consideration, in the up sampling method in this paper Bilinear is chosen and used.

The USDCNN network is shown in Fig. 5. The original image $\boldsymbol{x}_i$ first obtains the measured value $\boldsymbol{y}_i$ through the sampling network $F^{\mathrm{s}}(\cdot)$, and then reconstructs the image $\boldsymbol{x}_i''$ through Bilinear and the dilate convolution $F^{\mathrm{mus}}(\cdot)$,

$$\boldsymbol{x}_i'' = F^{\mathrm{mus}}(\boldsymbol{y}_i, \boldsymbol{W}^{\mathrm{mus}}) = F^{\mathrm{mus}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{mus}}). \quad (14)$$

By updating the $\boldsymbol{W}^{\mathrm{s}}$ and $\boldsymbol{W}^{\mathrm{mus}}$ in Eq. (14) by the Adam algorithm, in the loss function, we use the mean square error

$$\begin{aligned} & L(\boldsymbol{W}^{\mathrm{s}}, \boldsymbol{W}^{\mathrm{mus}}) \\ = & \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{x}_i'' - \boldsymbol{x}_i \right\|_2^2 \\ = & \frac{1}{N}\sum_{i=1}^{N} \| F^{\mathrm{mus}}(F^{\mathrm{s}}(\boldsymbol{x}_i, \boldsymbol{W}^{\mathrm{s}}), \boldsymbol{W}^{\mathrm{mus}}) - \boldsymbol{x}_i \|_2^2. \end{aligned} \quad (15)$$

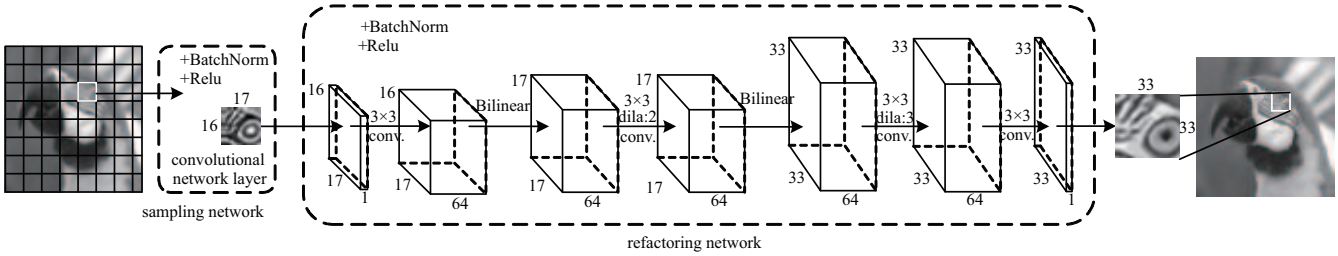


Fig. 5. USDCNN measurement network and reconstruction network.

4. Network training

4.1. Data set

In this paper, we use a data set from DR^2-Net for a total of 91 images. For the sake of fairness, the RGB image channel is converted to the YCrCb image channel which is the same as the image channel of ReconNet, MSRNet, and DR^2-Net. Select the image Y channel and first scale (0.75, 1, 1.5), then we will divide the image into blocks (33×33) and take the step size 14 to obtain a total of 87104 images which are used as the training set in this paper. Using four different sampling rates (MR = 0.25, 0.1, 0.04, 0.01), the measured value matrix sizes after 33×33 image block measurement are [16, 17], [9, 12], [4, 11], and [2, 5], respectively. Although the measurement network improves the reconstruction quality, it also reduces the anti-noise ability. When training, Gaussian noise with intensity $\sigma = 0.05$ is added to the measured value to improve the anti-noise ability against Gaussian noise. The 11 images in DR^2-Net and the data set BSD500 are used as test images. The data set BSD500 has a total of 500 test images, and the image size is 321×481 or 481×321.The network is trained with the Pytorch open source framework. All experiments are performed on the Inter Core i7-7700 CPU, with the main frequency of 3.6 GHz, memory of 16 GB, and Graphics card Quadro M2000 platform.

4.2. Training methods

IReconNet training is divided into two steps. First, the measurement network and the fully connected layer are trained, learning rate is 0.001. Each training is conducted 2×10^5 times, and the learning rate is reduced to 0.5 times of original rate, the training is conducted 1×10^6 times in total; The entire network is trained again, the learning rate is 10^{-4}, a total of 120 training rounds, the learning rate per 40 rounds drops to 0.5 times of original rate. The USDCNN directly trains the entire network, the learning rate is 10^{-3}, the training is carried out a 1×10^6 times in total, and the learning rate per 2×10^5 times is reduced to 0.5 times of original rate. Both IReconNet and USDCNN use the Adam method to train the network.

5. Experimental results and discussion

5.1. Refactoring results

In this paper, the TVAL3,[26] NLR-CS,[27] D-AMP,[28] ReconNet, DR^2-Net, and MSRNet algorithms are compared. The TVAL3, NLR-CS, and D-AMP are algorithms based on the iterative optimization, and the rest are based on deep learning algorithms. The experimental results are shown in the following Table 1.

Table 1. PSNRs of different algorithms at different sampling rates.

Image	Algorithm	MR = 0.25		MR = 0.10		MR = 0.04		MR = 0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
Barbara	TVAL3	24.19	24.20	21.88	22.21	18.98	18.98	11.94	11.96
	NLR-CS	28.01	28.00	14.80	14.84	11.08	11.56	5.50	5.86
	D-AMP	25.08	25.96	21.23	21.23	16.37	16.37	5.48	5.48
	ReconNet	23.25	23.52	21.89	22.50	20.38	21.02	18.61	19.08
	DR^2-Net	25.77	25.99	22.69	22.82	20.70	21.30	18.65	19.10
	MSRNet	26.69	26.91	23.04	23.06	21.01	21.28	18.60	18.90
	IReconNet	26.17	25.46	**24.14**	23.97	23.09	23.10	21.15	21.23
	USDCNN	27.59	27.04	**24.14**	23.95	23.07	23.07	21.06	21.16
Boats	TVAL3	28.81	28.81	23.86	23.86	19.20	19.20	11.86	11.88
	NLR-CS	29.11	29.27	14.82	14.86	10.76	11.21	5.38	5.72
	D-AMP	29.26	29.26	21.95	21.95	16.01	16.01	5.34	5.34
	ReconNet	27.30	27.35	24.15	24.10	21.36	21.62	18.49	18.83
	DR^2-Net	30.09	30.30	25.58	25.90	22.11	22.50	18.67	18.95
	MSRNet	30.74	**30.93**	26.32	26.50	22.58	22.79	18.65	18.88
	IReconNet	31.80	29.89	28.06	**27.68**	24.63	24.69	20.75	20.80
	USDCNN	**32.50**	30.30	**28.12**	27.64	**24.78**	**24.89**	**20.82**	**20.91**
Flinstones	TVAL3	24.05	24.07	18.88	18.92	14.88	14.91	9.75	9.77
	NLR-CS	22.43	22.56	12.18	12.21	8.96	9.29	4.45	4.77
	D-AMP	25.02	24.45	16.94	16.82	12.93	13.09	4.33	4.34
	ReconNet	22.45	22.59	18.92	19.18	16.30	16.56	13.96	14.08
	DR^2-Net	26.19	26.77	21.09	21.46	16.93	17.05	14.01	14.18
	MSRNet	26.67	26.89	21.72	21.81	17.28	17.40	13.83	14.10
	IReconNet	27.51	26.86	22.83	22.92	18.81	18.89	**15.68**	15.71
	USDCNN	**28.48**	**27.65**	**23.28**	**23.32**	**19.03**	**19.15**	**15.68**	**15.73**
Lena	TVAL3	28.67	28.71	24.16	24.18	19.46	19.47	11.87	11.89
	NLR-CS	29.39	29.67	15.30	15.33	11.61	11.99	5.95	6.27
	D-AMP	28.00	27.41	22.51	22.47	16.52	16.86	5.73	5.96
	ReconNet	26.54	26.53	23.83	24.47	21.28	21.82	17.87	18.05
	DR^2-Net	29.42	29.63	25.39	25.77	22.13	22.73	17.97	18.40
	MSRNet	30.21	**30.37**	26.28	26.41	22.76	23.06	18.06	18.35
	IReconNet	32.11	29.98	28.13	27.83	25.46	25.59	21.36	21.47
	USDCNN	**32.77**	30.31	**28.36**	**27.91**	**25.66**	**25.76**	**21.58**	**21.72**
Monarch	TVAL3	27.77	27.77	21.16	21.16	16.73	16.73	11.09	11.11
	NLR-CS	25.91	26.06	14.59	14.67	11.62	11.97	6.38	6.71
	D-AMP	26.39	26.55	19.00	19.00	14.57	14.57	6.20	6.20
	ReconNet	24.31	25.06	21.10	21.51	18.19	18.32	15.39	15.49
	DR^2-Net	27.95	28.31	23.10	23.56	18.93	19.23	15.33	15.50
	MSRNet	28.90	29.04	23.98	24.17	19.26	19.48	15.41	15.61
	IReconNet	30.72	28.88	25.89	25.78	21.44	21.60	17.43	17.47
	USDCNN	**31.42**	**29.16**	**26.26**	**26.05**	**22.09**	**22.30**	**17.48**	**17.54**
Peppers	TVAL3	29.62	29.65	22.64	22.65	18.21	18.22	11.35	11.36
	NLR-CS	28.89	29.25	14.93	14.99	11.39	11.80	5.77	6.10
	D-AMP	**29.84**	28.58	21.39	21.37	16.13	16.46	5.79	5.85
	ReconNet	24.77	25.16	22.15	22.67	19.56	20.00	16.82	16.96
	DR^2-Net	28.49	29.10	23.73	24.28	20.32	20.78	16.90	17.11
	MSRNet	29.51	**29.86**	24.91	25.18	20.90	21.16	17.10	17.33
	IReconNet	29.21	28.13	25.70	25.58	22.70	22.77	19.44	19.50
	USDCNN	29.50	28.32	**25.89**	**25.67**	**23.15**	**23.18**	**19.63**	**19.72**
Mean PSNR	TVAL3	27.84	27.87	22.84	22.86	18.39	18.40	11.31	11.34
	NLR-CS	28.05	28.19	14.19	14.22	10.58	10.98	5.30	5.62
	D-AMP	28.17	27.67	21.14	21.09	15.49	15.67	5.19	5.23
	ReconNet	25.54	25.92	22.68	23.23	19.99	20.44	17.27	17.55
	DR^2-Net	28.66	29.06	24.32	24.71	20.80	21.29	17.44	17.80
	MSRNet	29.48	29.67	25.16	25.38	21.41	21.68	17.54	17.82
	IReconNet	30.70	29.13	26.72	26.59	23.26	23.36	20.02	20.09
	USDCNN	**31.38**	**29.60**	**26.91**	**26.68**	**23.54**	**23.68**	**20.08**	**20.18**

In Table 1, the "w/o BM3D" means that BM3D[29] is not used to remove block effect, and "w/ BM3D" means that BM3D is used to remove block effect, the peak signal-to-noise ratios (PSNRs) of the six test pictures at four sampling rates are shown, and the "Mean PSNR" is the average PSNR of the 11 test pictures. At sampling rates of MR = 0.25, 0.10, 0.04, 0.01, the IReconNet improves on the "mean PSNR" by 1.22 dB, 1.56 dB, 1.85 dB, and 2.48 dB (without using BM3D) compared with the MSRNet (reconstructed image is uncorrected). At sampling rates of MR = 0.25, 0.10, 0.04, 0.01, IReconNet improves on the "Mean PSNR" by 1.22 dB, 1.56 dB, 1.85 dB, and 2.48 dB (without using BM3D) compared with MSRNet (reconstructed image uncorrected), and USDCNN is 0.68 dB, 0.19 dB, 0.28 dB, and 0.06 dB higher than IReconNet. As can be seen from the data, the measurement network improves the reconstruction quality greatly, especially at low sampling rates; the reconstruction network uses Bilinear over the fully connected layer at high sampling rates. The IReconNet and USDCNN can improve the reconstruction quality by using BM3D algorithm at low sampling rate MR = 0.04 and 0.01, but by using BM3D algorithm at sampling rate MR = 0.25, 0.10 can reduce the reconstruction quality.

Figure 6 shows the reconstruction effects of four test pictures at four sample rates (without using BM3D). The ReconNet and DR^2-Net contain more artifacts at large sampling rates. The MSRNet (reconstructed image uncorrected) has fewer artifacts at high sampling rates, but many details are not well reconstructed. Both IReconNet and USDCNN can reconstruct image details at large sampling rates, and are superior to other algorithms at small sampling rates.

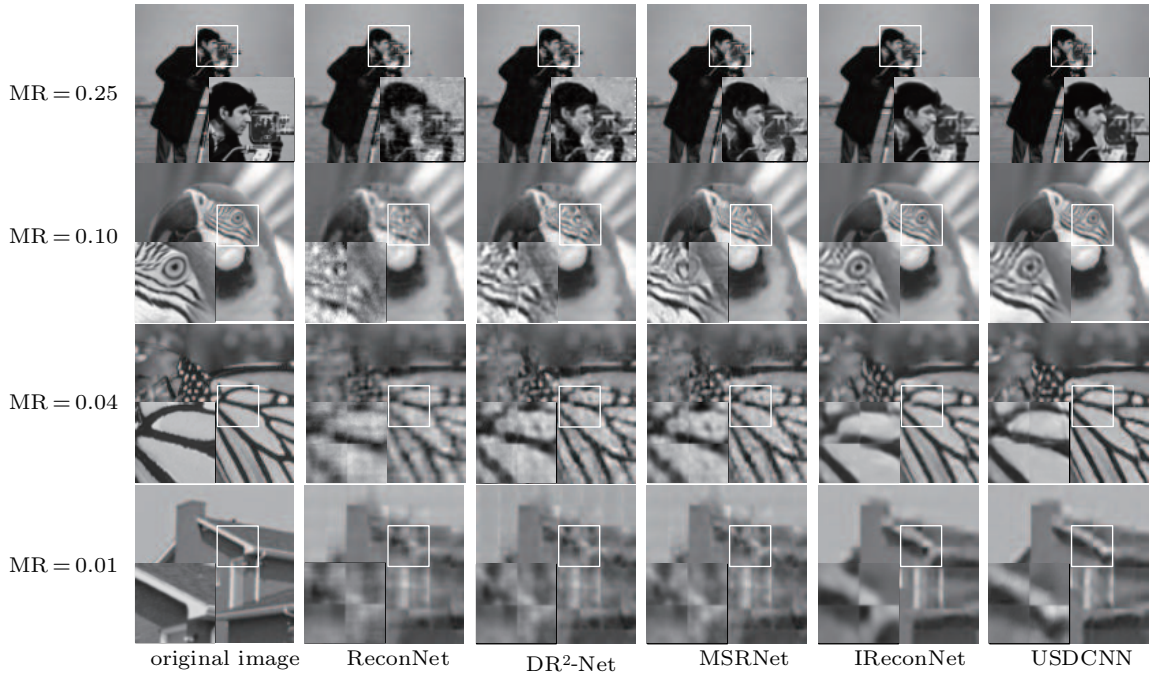


Fig. 6. Reconstruction performances of different algorithms.

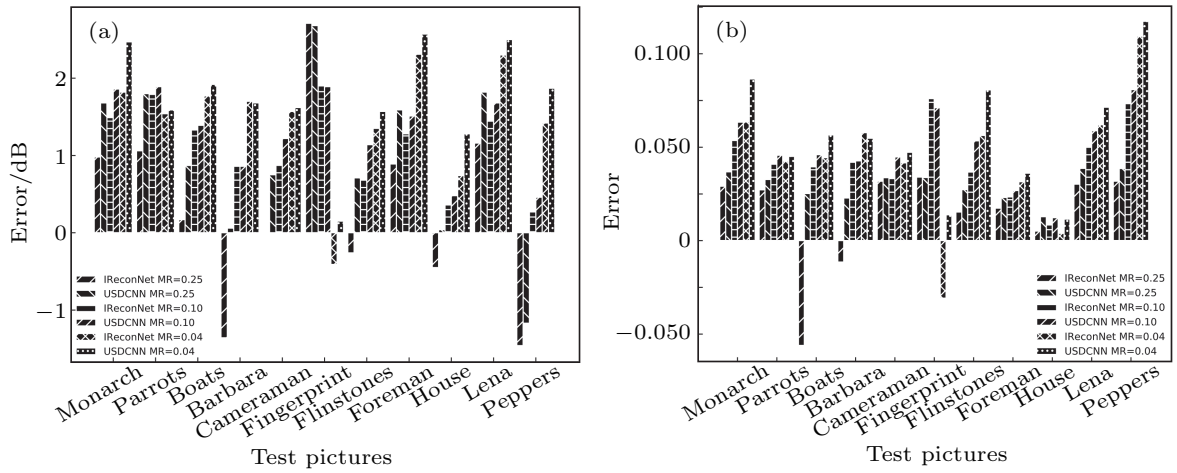


Fig. 7. (a) PSNR error between algorithm and MSRNet (reconstructed image corrected) on 11 test images, and (b) SSIM error of algorithm and MSRNet (reconstructed image corrected) on 11 test images.

Structural similarity index (SSIM) is a measure of the similarity between two images. Here, the similarity refers to the resemblance in brightness, contrast and structure. The value of SSIM is between 0 and 1, and the closer to 1 the value, the more similar the two images are. After the MSRNet reconstructs the image through the network, a correction process is started to further improve the quality of the reconstruction. The PSNR and SSIM errors of the algorithm and MSRNet (reconstructed image corrected) are shown in Figs. 7(a) and 7(b).

As can be seen from Fig. 7, the error values of most of PSNRs and SSIMs are greater than 0, indicating that the reconstruction effect of IReconNet and USDCNN are still better than that of the modified MSRNet.

5.2. Time complexity

The reconstruction time of traditional iterative algorithm is much longer than that based on deep learning,[30] so only the reconstruction time based on deep learning algorithm is compared here. To be fair, Table 2 shows ReconNet, DR^2-Net, MSRNet, IReconNet, USDCNN's refactoring network consumption time, and the total time that IReconNet and USDCNN measure the network and restructure the network.

Table 2. Average reconstruction times on data set BSD500.

Algorithm	MR = 0.01 (CPU/GPU)	MR = 0.04 (CPU/GPU)	MR = 0.1 (CPU/GPU)	MR = 0.25 (CPU/GPU)
ReconNet	0.7702/0.0030	0.7761/0.0027	0.7978/0.0028	0.7876/0.0028
DR^2-Net	1.5923/0.0052	1.5962/0.0051	1.5912/0.0051	1.5811/0.0051
MSRNet	1.2054/0.0044	1.2110/0.0044	1.2291/0.0044	1.2306/0.0044
IReconNet($F^{\mathrm{f}}(\cdot)$)	0.7788/0.0023	0.7804/0.0023	0.7869/0.0024	0.8198/0.0024
IReconNet($F^{\mathrm{s}}(\cdot)+F^{\mathrm{f}}(\cdot)$)	0.8487/0.0045	0.8669/0.0045	0.8981/0.0046	0.9546/0.0044
USDCNN($F^{\mathrm{mus}}(\cdot)$)	**0.3773/0.0019**	**0.4159/0.0019**	**0.4307/0.0020**	**0.5171/0.0019**
USDCNN($F^{\mathrm{s}}(\cdot)+F^{\mathrm{mus}}(\cdot)$)	0.4432/0.0041	0.4990/0.0041	0.5328/0.0040	0.6593/0.0040

IReconNet($F^{\mathrm{f}}(\cdot)$) and USDCNN($F^{\mathrm{mus}}(\cdot)$) represent the average reconstruction time of this algorithm. IReconNet($F^{\mathrm{s}}(\cdot)+F^{\mathrm{f}}(\cdot)$) and USDCNN($F^{\mathrm{s}}(\cdot)+F^{\mathrm{mus}}(\cdot)$) represent the average time of measurement network and reconstruction network. As can be seen from Table 2, the USDCNN is the shortest in refactoring time, and the measurement network consumption time is faster on the CPU than those from the other algorithms.

5.3. Generalization ability

In order to test the generalization ability of the algorithm on the big data set, the reconstruction performances of ReconNet, DR^2-Net, and MSRNet are compared in the data set BSD500 (500 pictures). As shown in Fig. 8, the mean PSNR and SSIM, the algorithms used in this paper on the BSD500 data set, are higher than those from other algorithms. The average PSNR of USDCNN is higher than that of IReconNet at the four sampling rates. The average SSIM of USDCNN is also higher than that of IReconNet at sampling rate MR = 0.04, 0.10, 0.25. Experimental results show that the USDCNN has the best generalization ability, and IReconNet is next only to USDCNN.

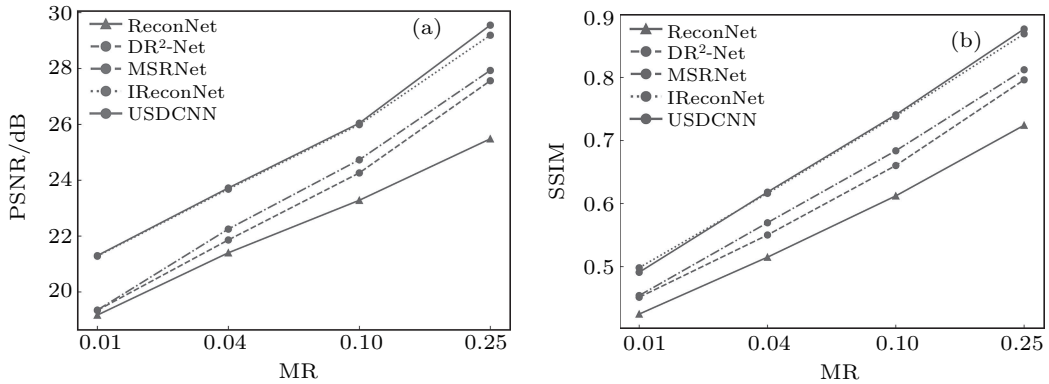


Fig. 8. (a) Average PSNR of each algorithm in the data set BSD500 and sampling rate MR = 0.01, 0.04, 0.10, 0.25; (b) average SSIM of each algorithm in data set BSD500 and the sampling rate MR = 0.01, 0.04, 0.10, 0.25.

5.4. Anti-noise performance

5.4.1. Gaussian noise

The sampling rate MR = 0.25, 0.10 and the Gaussian noise of four different noise intensities (σ = 0.01, 0.05, 0.10, 0.25) are added to the measured values, respectively. As shown in Figs. 9 and 10, the average PSNR of USDCNN is the highest in the four different noise intensities. The average PSNR of IReconNet is lower than that of USDCNN, but

the overall effect is still better than those of ReconNet, DR^2-Net, and MSRNet. The USDCNN is more robust to Gaussian noise, and IReconNet is next only to USDCNN.

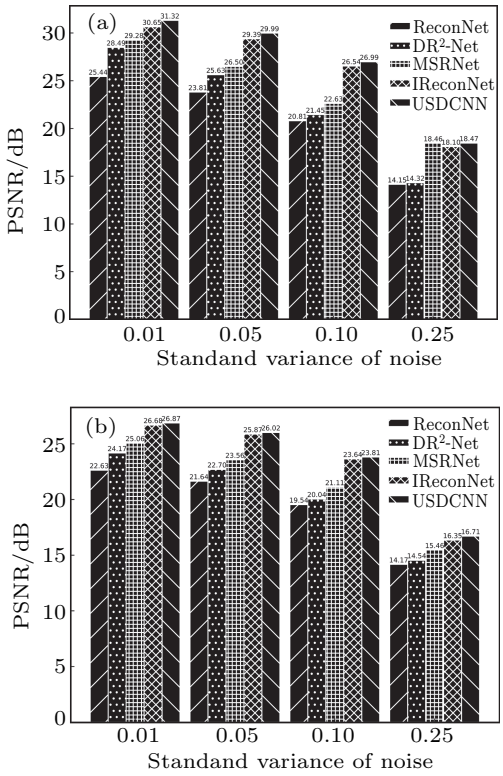


Fig. 9. Robustness of 11 test images to Gaussian noise at (a) sampling rate MR = 0.25 and noise intensities $\sigma = 0.01, 0.05, 0.10, 0.25$ and (b) sampling rate MR = 0.10 and noise intensities $\sigma = 0.01, 0.05, 0.10, 0.25$.

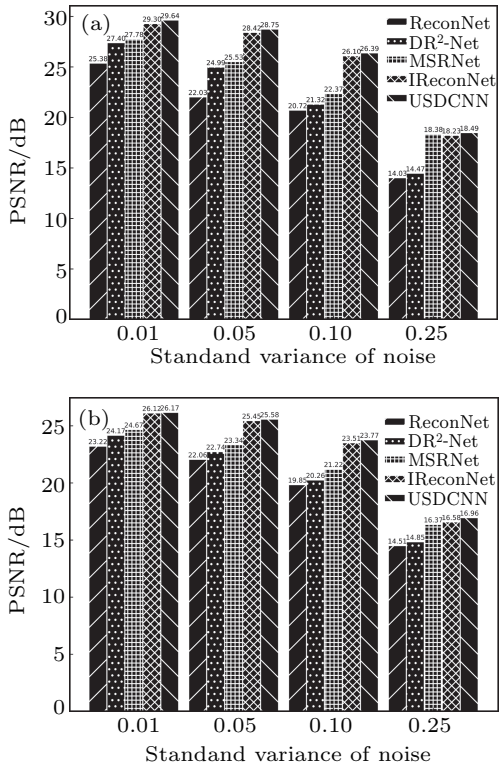


Fig. 10. Robustness of data set BSD500 to Gaussian noise, at (a) sampling rate MR = 0.25 and noise intensities $\sigma = 0.01, 0.05, 0.10, 0.25$, and (b) sampling rate MR = 0.10 and noise intensities $\sigma = 0.01, 0.05, 0.10, 0.25$.

5.4.2. Salt and pepper noise

At the sampling rate MR = 0.25, 0.10, four different noise intensities ($\sigma = 0.001, 0.01, 0.05, 0.10$) of salt and pepper noise are added to the measured value. As shown in Figs. 11

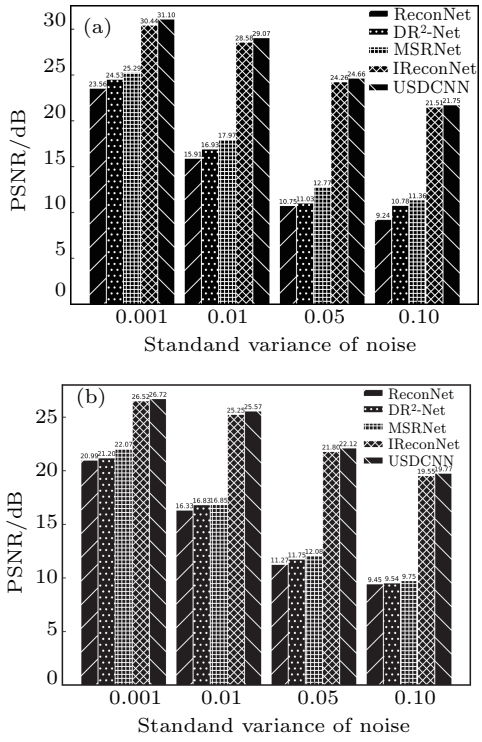


Fig. 11. Robustness of 11 test images to salt and pepper noise, at (a) sampling rate MR = 0.25 and noise intensities $\sigma = 0.001, 0.01, 0.05, 0.10$, and (b) sampling rate MR = 0.10 and noise intensities $\sigma = 0.001, 0.01, 0.05$, and 0.10.

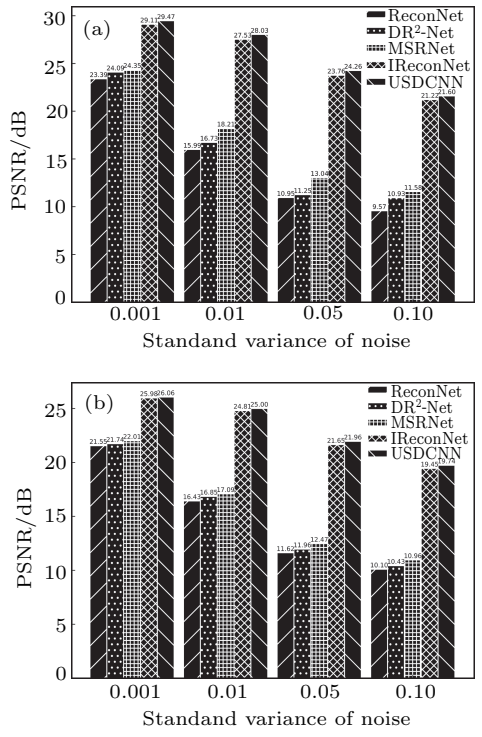


Fig. 12. Robustness of data set BSD500 to salt and pepper noise, at (a) sampling rate MR = 0.25 and the noise intensities $\sigma = 0.001, 0.01, 0.05, 0.10$, and (b) sampling rate MR = 0.10 and the noise intensities $\sigma = 0.001, 0.01, 0.05, 0.10$.

and 12, the average PSNR of USDCNN and IReconNet are higher than that of ReconNet, DR^2-Net, and MSRNet under four different noise intensities, and the difference is large. The USDCNN is the most robust against salt and pepper noise. Although the IReconNet is not so good as USDCNN, it is better than other algorithms.

Reconstructed images with four noise intensities added at a sampling rate of MR $= 0.25$ are shown in Fig. 13.

As can be seen from Fig. 13, the ReconNet, DR^2-Net, and MSRNet are poorly robust to salt and pepper noise. The noise affects the entire image block. The image profile is not visible when the noise intensity $\sigma = 0.05$, while in the IReconNet and USDCNN the reconstruction qualities have good robustness to salt and pepper noise. The image contour can still be seen when the noise intensity $\sigma = 0.10$.

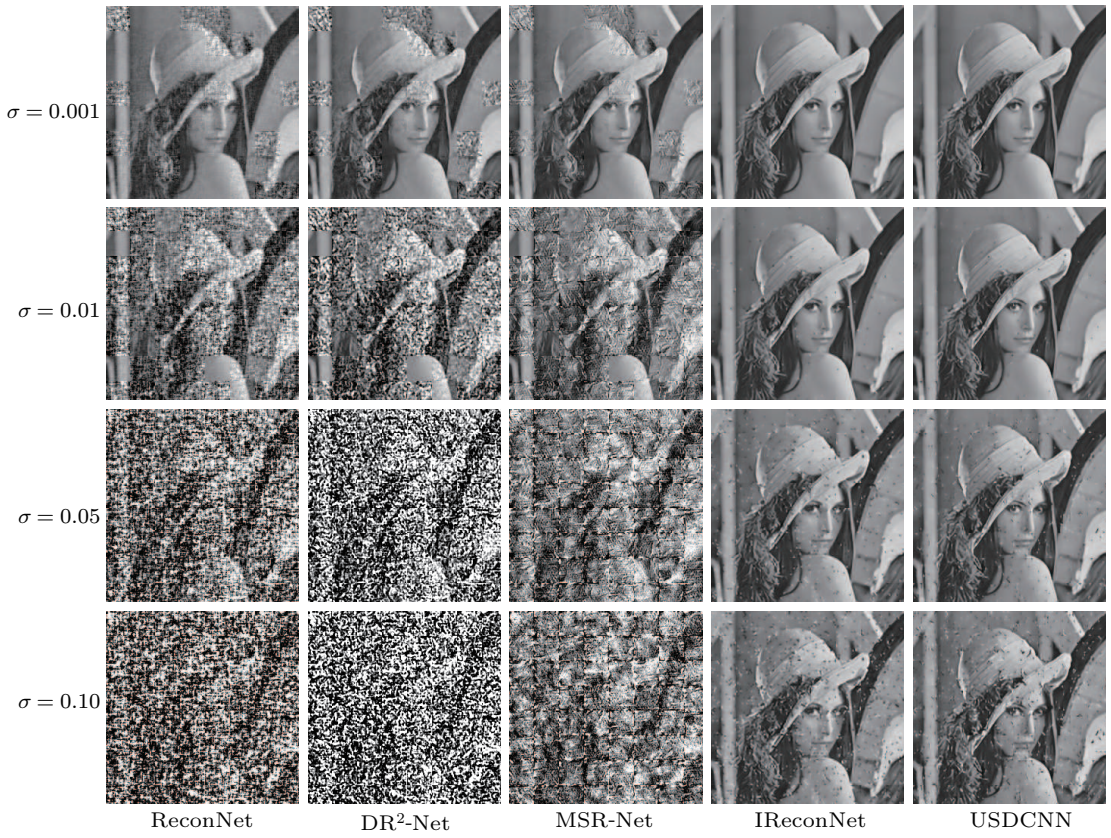


Fig. 13. Reconstruction diagrams after adding salt and pepper noise to each algorithm.

6. Conclusions

In this paper, we proposed two network models based on deep learning, IReconNet and USDCNN. The experimental results of 11 commonly used test images and datasets BSD500 show that in the IReconNet the random Gaussian measurement matrix is replaced with the measurement network, which greatly improves the reconstruction quality, and is better than those from the algorithms in other literature. In order to solve the problem that the measured values obtained by the measurement network still retain the spatial information about the image, the USDCNN introduces Bilinear and dilated convolution into the reconstruction network, which has better performance in reconstruction time, reconstruction quality, and robustness.

References

[1] Donoho D L 2006 *IEEE Trans. Inf. Theory* **52** 1289
[2] Candes E J, Romberg J and Tao T 2006 *IEEE Trans. Inf. Theory* **52** 489
[3] Candes E J and Wakin M B 2008 *IEEE Signal Process. Mag.* **25** 21
[4] Rani M, Dhok S B and Deshmukh R B 2018 *IEEE Access* **6** 4875
[5] Liu X J, Xia S T and Fu F W 2017 *IEEE Trans. Inf. Theory* **63** 2922
[6] Moshtaghpour A, Jacques L, Cambareri V, Degraux K and Vleeschouwer C D 2016 *IEEE Signal Process. Lett.* **23** 25
[7] Nguyen N, Needell D and Woolf T 2017 *IEEE Trans. Inf. Theory* **63** 6869
[8] Lee J, Choi J W and Shim B 2016 *J. Commun. Netw.* **18** 699
[9] Wang R, Zhang J L, Ren S L and Li Q J 2016 *Tsinghua Sci. Technol.* **21** 71
[10] Davenport M A, Needell D and Wakin M B 2013 *IEEE Trans. Inf. Theory* **59** 6820
[11] Kong M, Chen M S, zhang L, Cao X Y and Wu X L 2016 *Chin. Phys. Lett.* **33** 018402
[12] Zhang Y X, Li Y Z, Wang Z Y, Song Z H, Lin R, Qian J Q, Yao J N 2019 *Meas. Sci. Technol.* **30** 025402
[13] Zhao R Q, Wang Q, Fu J and Ren L Q 2020 *IEEE Trans. Image Process.* **29** 1654
[14] Mousavi A, Patel A B and Baraniuk R G 2015 *53rd Annual Allerton Conference on Communication, Control, and Computing*, September 29–October 2, 2015, Monticello, USA, p. 1336
[15] Kulkarni K, Lohit S, Turaga P, Kerviche R and Ashok A 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, June 27–30, 2016, Las Vegas, USA, p. 449

[16] Dong C, Loy C C, He K and Tang X 2016 *IEEE Trans. Pattern Anal. Mach. Intell.* **38** 295
[17] Yao H T, Dai F, Zhang SL, Zhang Y D, Tian Q and Xu C S 2019 *Neurocomputing* **359** 483
[18] He K M, Zhang X Y, Ren S Q and Sun J 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, June 27–30, 2016, Las Vegas, USA, p. 770
[19] Lian Q S, Fu L P, Chen S Z and Shi B S 2019 *Acta Autom. Sin.* **45** 2082
[20] Zhang Z D, Wang X R and Jung C 2019 *IEEE Trans. Image Process.* **28** 1625
[21] Ioffe S and Szegedy C 2015 *Proceedings of the* 32*nd International Conference on Machine Learning*, July 6–11, 2015, Lille, France, p. 448
[22] Liu Y N, Niu H Q and Li Z L 2019 *Chin. Phys. Lett.* **36** 044302
[23] Kingma D P and Ba J L 2014 arXiv: 1412.6980v9 [cs.LG]
[24] Shi W Z, Caballero J, Huszár F, Totz J, Aitken A P, Bishop R, Rueckert D and Wang Z H 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, June 27–30, 2016, Las Vegas, USA, p. 1874
[25] Zeiler M D, Taylor G W and Fergus R 2011 *International Conference on Computer Vision*, November 6–13, 2011, Barcelona, Spain, p. 2018
[26] Li C B, Yin W T, Jiang H and Zhang Y 2013 *Comput. Optim. Appl.* **56** 507
[27] Dong W S, Shi G M, Li X, Ma Y and Huang F 2014 *IEEE Trans. Image Process.* **23** 3618
[28] Metzler C A, Maleki A and Baraniuk R G 2016 *IEEE Trans. Inf. Theory* **62** 5117
[29] Dabov K, Foi A, Katkovnik V and Egiazarian K 2007 *IEEE Trans. Image Process.* **16** 2080
[30] Shi H and Wang L D 2019 *Acta Phys. Sin.* **68** 200501 (in Chinese)