Global Sensing and Measurements Reuse for Image Compressed Sensing

Zi-En Fan, Feng Lian,* Jia-Ni Quan
Xi'an Jiaotong University, Xi'an, China

fze0012@stu.xjtu.edu.cn,lianfeng1981@mail.xjtu.edu.cn,jiani_quan@stu.xjtu.edu.cn

Abstract

Recently, deep network-based image compressed sensing methods achieved high reconstruction quality and reduced computational overhead compared with traditional methods. However, existing methods obtain measurements only from partial features in the network and use it only once for image reconstruction. They ignore there are low, mid, and high-level features in the network [38] and all of them are essential for high-quality reconstruction. Moreover, using measurements only once may not be enough for extracting richer information from measurements. To address these issues, we propose a novel Measurements Reuse Convolutional Compressed Sensing Network (MR-CCSNet) which employs Global Sensing Module (GSM) to collect all level features for achieving an efficient sensing and Measurements Reuse Block (MRB) to reuse measurements multiple times on multi-scale. Finally, we conduct a series of experiments on three benchmark datasets to show that our model can significantly outperform state-of-the-art methods. Code is available at: https://github.com/fze0012/MR-CCSNet.

1. Introduction

Compressed Sensing [11](CS), a signal processing technique for efficiently acquiring and reconstructing a signal, has developed rapidly and attracted the attention of many researchers. Given a high-dimensional signal $x \in \mathbb{R}^N$, the measurements of x, denoted by $y \in \mathbb{R}^M$, is formulated as $y = \Phi x$, where $\Phi \in \mathbb{R}^{M \times N}$ is sensing matrix and $\frac{M}{N}$ is sampling ratio. Because $M \ll N$, recovering x is generally impossible for the ill-posed problem. CS shows that the signal x can be reconstructed from y with high probability when the signal x is sparse in some domain [6, 12].

In the study of image CS, the two core problems are (1) the design of sensing matrix and (2) recovering the original signal x from its linear measurements y. For the former one, many matrices [1, 10, 15, 16, 25] are proposed, but they

---

*Corresponding author.

are hand-designed and ignore there are statistical correlation between different elements of signal. For the latter one, the papers of [7, 23, 30, 41, 42] propose methods for exploring image priors and combining optimization criteria and iterative thresholding algorithms [17]. These methods require high computational overhead and perform poorly when the sampling ratio is extremely low.

In recent years, deep learning has been widely used in computer vision and shows superior performance [21, 24]. Researchers were inspired to solve these two challenges of CS with deep learning, called Deep Compressed Sensing (DCS). A few DCS methods [27, 28, 32, 34, 40] have been proposed and achieve promising results since the powerful learning and representation capabilities of neural networks.

Despite their success, existing DCS methods only use a convolutional layer to learn the sensing matrix, which ignores the spatial features in the image. In addition, because the residual architecture is widely used in reconstruction network, the reconstruction quality relies on it. To address these issues, Zheng *et al.* proposed RK-CCSNet [43]. For the former one, RK-CCSNet use the Sequential Convolutional Module (SCM) to gradually compact the image size through a sequence of filters. For the latter one, RK-CCSNet proposed the second-order residual architecture according to the relationship between ResNet [18] and Ordinary Differential Equation.

Although RK-CCSNet proposed an effective strategy for image CS, it always suffers from these problems: (1) There are hierarchical nature of the features in the convolutional neural networks (CNNs): the low, mid, and high layer learn features such as edges, complex textures, and objects, respectively. But RK-CCSNet only samples from the highest layer, which ignores a large amount of rich features contained in those neglected layers; (2) Existing methods [27, 32, 40, 43] recover the original image from measurements using deep learning, which takes measurements as input and use it only once. It extracts features from measurements in a rather shallow manner.

To address these issues, we propose Global Sensing Module (GSM) and Measurements Reuse Block (MRB). In GSM, as shown in Fig. 1, we first use a convolutional layer
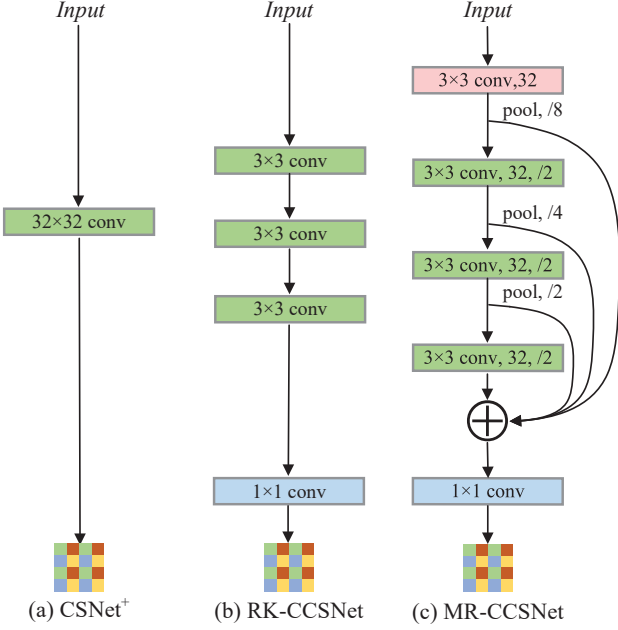
(a) CSNet⁺   (b) RK-CCSNet   (c) MR-CCSNet

Figure 1. This figure shows the sensing network of CSNet[+] [32], RK-CCSNet [43], and our proposed model. **(a)**: There is only a convolutional layer, so the spatial features in the image are ignored. **(b)**: Although there are multiple convolutional layers, it only samples from the highest layer. Therefore, there are the highest level features in the measurements and low and mid-level features are ignored. **(c)** Overview of GSM. To take advantage of the hierarchical nature of the CNNs, which are neglected in RK-CCSNet, we use a shortcut connection pass the features of different layer to the end. And then, 1×1 convolutions are used to sample. Because we sample from all level features, there are richer features in measurements than CSNet[+] and RK-CCSNet.

to get a high dimensional feature. After that, we gradually compact the feature maps by multiple convolutional layers. And then we collect all level features in the network. Finally, we obtain measurements by 1×1 convolutions. In order to match dimensions, we add pooling layer into the shortcut connection. In MRB, we first compact the phased reconstructed result and get multiple feature maps. After that, we extract matching information from measurements. Finally, we fuse them on multi-scale. It is a promising manner to move from shallow measurements utilizing to deep.

We conduct experiments on three benchmark data sets: BSDS500 [2], Set5 [4], and Set14 [39] and chose PSNR and SSIM [19] as the evaluation metrics. Evaluation results indicate that MR-CCSNet can significantly outperform state-of-the-art methods. In particular, we show that our model can achieve high reconstruction quality at low sampling ratio. In addition, we show that GSM and MRB are effective by ablation studies.

To conclude, our contributions are three-fold: (1) proposal of the GSM which can achieve efficient sampling; (2) proposal of the MRB for making full use of measurements; (3) building an end-to-end network MR-CCSNet for image CS based on GSM and MRB, and demonstrating its effectiveness on three benchmark data sets.

2. Related work

The goal of compressed sensing is to recover the original signal x from its linear measurements y. We briefly review the relevant work by grouping the existing methods into the following two categories.

Traditional Compressed Sensing   Traditional CS methods recover a signal x from the measurements y by solving a sparsity-regularized optimization problem of the form

$$\min_{x} \frac{1}{2} \|\Phi x - y\|_2^2 + \lambda \|\Psi x\|_1, \tag{1}$$

where Ψx are the transform coefficients of x with respect to domain Ψ and the sparsity of Ψx is characterized by ℓ_1 norm.

Representative methods include the convex optimization methods [8], the greedy algorithms [26, 35], and the gradient descent methods [9,13,36]. For image compressed sensing, many researchers introduce other prior as a regularization item. In [23], Li *et al.* used the total variation (TV) regularized constraint to replace the sparsity-based one for enhancing the local smoothness. In [41], Zhang *et al.* proposed group sparse representation (GSR) to enhance both image sparseness and non-local self-similarity. Furthermore, some image CS methods incorporated additional criteria into the projected Landweber (PL) algorithm [3]. In [15], Gan proposed block-based CS by incorporating Wiener filtering into PL iteration. In recent years, researchers have also proposed many improved PL-based methods [7,14,31]. Besides image reconstruction methods, some attention is also paid to the sensing matrix. In most works, the sensing matrix is a random matrix such as a Gaussian or Bernoulli matrix, which satisfies the Restricted Isometry Property (RIP) [5] with a large probability. Although so many methods have been proposed in traditional CS, they all demand high computational overhead and perform poorly at low sampling ratios.

Deep Compressed Sensing   The main idea of DCS is to learn the inverse mapping from the measurements to the original signal using a neural network, so the speed and accuracy of reconstruction are improved. Generally, the network is trained by minimizing the loss function

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{k} \|x_i - F(y_i, \theta)\|_2^2, \tag{2}$$

where the x_i is the original image, y_i is the measurements of x_i, and F is the neural network parameterized by θ. Many
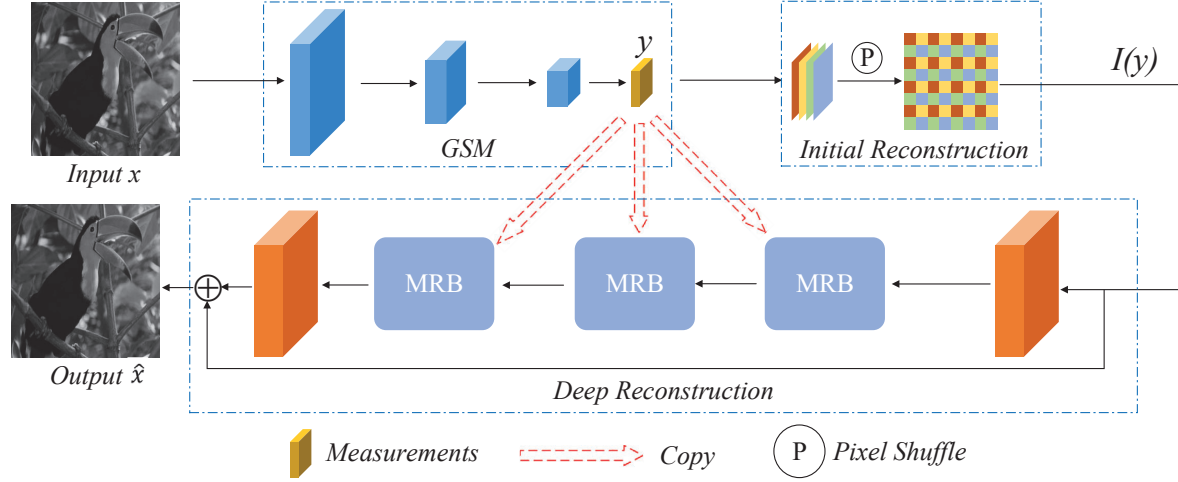
Figure 2. Overview of the proposed model. For the original image x, we obtain measurements y from the sensing network GSM. And then the initial reconstructed image $I(y)$ is generated by the initial reconstruction network. Finally, we refine the $I(y)$ by the deep reconstruction network.

DCS methods have been proposed [22, 27, 29, 32, 33, 43]. In [29], Mousavi *et al*. proposed a stacked denoising autoencoder (SDA) to capture statistical correlation between different elements of signals. However, SDA has the computational complexity because it is full connection between any two successive layers. In [22], Kulkarni *et al*. introduced a CNN-based method called ReconNet, which can reduce computational complexity by weight sharing. Mousavi and Baraniuk [27] argued that real world signals are not exactly sparse on a fixed basis and the recovery algorithms take a lot of time to converge. And they proposed Deep-Inverse which learns both a effective representation for the signals and an inverse map. Shi *et al*. [32] argued that these methods ignore the characteristics of signal and proposed a end-to-end model CSNet$^+$ which uses convolutional neural network in sampling and reconstruction stage. However, these methods train different models for different sampling ratios, which is difficult to deploy for practical applications. Hence, Shi *et al*. [33] attempted to solve this problem with greedy method and proposed SCSNet. Mousavi *et al*. [28] proposed DeepSSRR which employs a parallelization scheme in the signal sensing and recovery process to accelerate the convergence speed. In [43], Zheng argued that existing end-to-end methods do not preserve the spatial features in the image and proposed RK-CCSNet, which applies Sequential Convolutional Module (SCM) to gradually compact measurements through a series of convolution filters. In addition, RK-CCSNet also proposed a novel Learned Runge-Kutta Block (LRKB) based on the famous Runge-Kutta methods for improving the reconstruction quality.

Our work is also inspired by the idea of multi-scale in image processing. In [37], Xu *et al*. proposed a Laplacian pyramid reconstructive adversarial network (LAPRAN) which reconstructs the original image through multiple stages with different resolution simultaneously. Our model also employs MRB to fuse features learned from measurements on multi-scale.

3. Methodology

In this section, we will introduce our model in the case of sampling ratio is 6.25%. Fig. 2 shows the architecture of MR-CCSNet. Following CSNet$^+$ [32] and RK-CCSNet [43], MR-CCSNet has a sensing network GSM, an initial reconstruction network, and a deep reconstruction network. Firstly, we obtain the measurements from the sensing network. And then the initial reconstruction network generates initial reconstructed image by a linear mapping. Because the quality of initial reconstructed image is not enough, we refine the initial reconstructed image by a non-linear deep reconstruction network. To move from shallow measurements utilizing to deep, we stack multiple MRBs in the deep reconstruction network.

In the sensing network $S(\cdot)$, we directly use convolutional layers for the whole images instead of dividing the images into non-overlapping block [32, 33, 43]. For satisfying the linear property, there is no bias and activation function in the network. This process can be written as:

$$y = S(x), \qquad (3)$$

where $x \in \mathbb{R}^{1 \times H \times W}$ and $y \in \mathbb{R}^{4 \times \frac{H}{8} \times \frac{W}{8}}$.

In the initial reconstruction network $I(\cdot)$, the depth-wise convolution layer expands the measurements in channel dimension and the shape becomes $64 \times \frac{H}{8} \times \frac{W}{8}$. Then we get

a $1 \times H \times W$ tensor by a pixel shuffle layer. This is the first time to utilize the measurements.

In the deep reconstruction network $D(\cdot)$, we first convert the initial reconstructed $I(y)$ image to a high dimensional feature by a convolutional layer. Then repeated MRBs, which share the same internal structure, are used to fuse them with matching features extracted from measurements y multiple times on multi-scale. This is the second time to utilize the measurements.

Finally, we use a convolutional layer to reconstruct the image from high dimensional features. In addition, we add a shortcut connection to the deep reconstruction network. The final reconstructed image $\hat{x}$ can be written as:

$$\hat{x} = D(I(y)) + I(y) \tag{4}$$

Our model uses two novel modules, GSM and MRB. They are explained below.

3.1. Global Sensing Module

By analyzing existing methods, we argue that a good feature extraction network can help sample. In addition, we learn that convolutional neural networks extract features in a hierarchical manner which means layers close to the input to learn low-level features, like lines and simple textures, and layers deeper in the model to learn high-order features, like shapes or specific objects from [38]. Based on these two principles, our proposed method GSM, as shown in Fig. 3a, has two stages. In the first stage, we use 3×3 convolution layers to extract features. In the second stage, we collect all level features in the network and use a 1×1 convolution layer to sample, rather than only from the low features (*i.e.* CSNet$^+$) or high features (*i.e.* RK-CCSNet).

In GSM, to collect all level features for sampling, we use a shortcut connection to pass the features of different layers to the end, and the pooling layer is added for matching the dimensions.

When the sampling ratio changes, the GSM is not flexible for meeting the new requirements. Inspired by ResNet [18], we propose the GSM$^+$, as shown in Fig. 3b. Different from GSM, we add a shortcut connection between two successive layers rather than add it from different layers to the end directly. The building block of GSM$^+$ is marked with red dotted box and defined as:

$$y_{t+1} = Conv(y_t) + P(y_t), \tag{5}$$

where $Conv$ and P denote convolution layer and mean-pooling layer, respectively. The sampling ratio is controlled by the number of building block and the blue block, so it is flexible and can be easily used at various sampling ratios by repeating the building block.

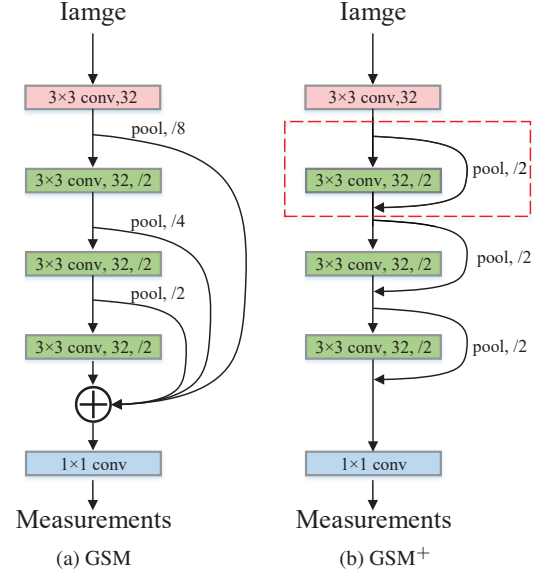In GSM$^+$, we can observe that it collect all level features for sampling, which is equivalent to GSM, by an iterative

Figure 3. Comparison of the GSM and the GSM$^+$. The GSM$^+$ is flexible and can be easily used at various sampling ratios by repeating the building block, which is marked with red dotted box.

manner. Furthermore, there are richer features than GSM at each layer, because the features from former layer are passed to the current layer by shortcut connections. In a way, it achieve a more efficient feature extraction. When the sampling ratio is 50%, there is only one building block in GSM$^+$, so GSM$^+$ degenerate into GSM. As the sampling ratio decreases, GSM is a special form of GSM$^+$.

In the CS theory, the measurements is obtained by a linear mapping. It is trivial that the convolution layer and the mean-pooling layer are linear mappings. So the building block is linear mapping. According to composition preserves linearity, the GSM$^+$ is a linear mapping.

3.2. Measurements Reuse Block

The measurements are used only once for image reconstruction, which is difficult to extract richer information from measurements. The goal of MRB is to explore a novel approach for making full use of measurements multiple times on multi-scale.

Fig. 4 illustrates the architecture of MRB. Phased reconstructed result $f_t \in \mathbb{R}^{C \times H \times W}$ and measurements $y \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$ are fed into MRB. We firstly use two convolutional layers, denoted as $Conv_1$ and $Conv_2$, to obtain a compacted feature map $f^\downarrow$ and $f^{\Downarrow}$. This process can be written as:

$$f^\downarrow = Conv_1(f_t), \tag{6}$$

$$f^{\Downarrow} = Conv_2(f^\downarrow), \tag{7}$$

where $f^\downarrow \in \mathbb{R}^{C \times \frac{H}{2} \times \frac{W}{2}}$, $f^{\Downarrow} \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$. To fuse
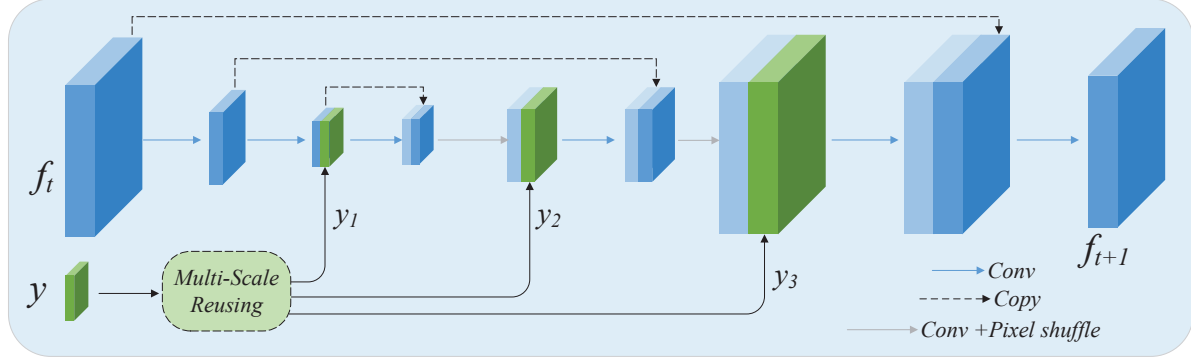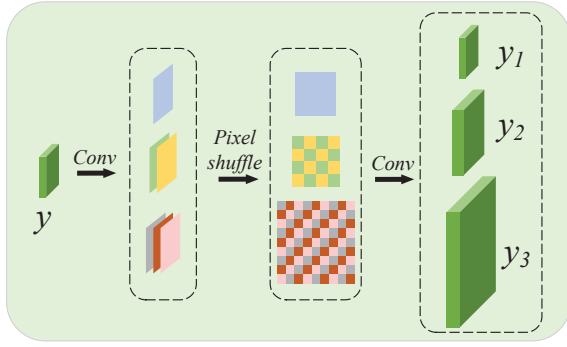
Figure 4. Overview of the proposed MRB.

Figure 5. Overview of Multi-Scale Reusing. It aims to extract matching information for the backbone network of MRB.

them with measurements on multi-scale, we then extract matching information from measurements and obtain three feature maps $y_1 \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$, $y_2 \in \mathbb{R}^{C \times \frac{H}{2} \times \frac{W}{2}}$, and $y_3 \in \mathbb{R}^{C \times H \times W}$ by Multi-Scale Reusing, which is shown in Fig. 5. Next, y_1 is added into the backbone network of MRB and obtain F_1 by a concatenation operation and a convolutional layer. To preserve existing reconstruction results, we copy the $f^{\Downarrow}$ again and fuse them with F_1 by a convolutional layer. Finally, a pixel shuffle layer is used to expand the fused feature map for next process. This process can be written as:

$$F_1 = Conv_3(f^{\Downarrow} \oplus y_1), \qquad (8)$$

$$f^{\uparrow} = Pixel(Conv_4(F_1 \oplus f^{\Downarrow})), \qquad (9)$$

where $\oplus$ denotes a concatenation operation, $y \in \mathbb{R}^{2 \times \frac{H}{4} \times \frac{W}{4}}$, $y_1 \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$, $F_1 \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$ and $f^{\uparrow} \in \mathbb{R}^{C \times \frac{H}{2} \times \frac{W}{2}}$. By repeating this process, phased reconstructed result measurements and measurements are fused at multi-scale. Then the output $f_{t+1} \in \mathbb{R}^{C \times H \times W}$ is utilized as the input of next operation.

The MRB is not only a promising way for improving utilization of measurements, but also refines the phased reconstructed result at multi-scale.

3.3. Loss function

In the training phase, we use the mean square error to measure the reconstruction quality. Specifically, for the initial reconstruction network, the loss function can be written as :

$$l_{int} = \sum_{k=1}^{n} \|I(S(y_k; \theta); \phi_{int}) - x_k\|_F^2. \qquad (10)$$

For the deep reconstruction network, the loss function can be written as :

$$l_{deep} = \sum_{k=1}^{n} \|D(I(S(y_k; \theta); \phi_{int}); \phi_{deep}) - x_k\|_F^2, \qquad (11)$$

where the θ, ϕ_{int}, and ϕ_{deep} denote the parameters of the sensing network $S(\cdot)$, the initial reconstructed network $I(\cdot)$, and the deep reconstructed network $D(\cdot)$, respectively. Therefore, the loss function of MR-CCSNet is defined as:

$$l = l_{deep} + l_{int}. \qquad (12)$$

4. Experiments

4.1. Datasets and implementation details

Following RK-CCSNet [43], we use 400 images from BSDS500 [2] dataset to train our model. For testing, we report the performance on three standard benchmark datasets: Set5 [4], Set14 [39], and BSDS100 [2]. We convert these images into YCbCr color space and only the Y channel is used as the input for training and testing. During training, in order to increase the number of samples, we randomly crop the image with patch size 96×96, and randomly flip horizontally. During testing, because the size of these images is inconsistent, we resize the image from Set5 and Set14 into 256×256 and the image from BSDS100 into 480×320. To optimize our model, we use Adam optimizer [20] with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is set to 4 and our model is trained for 200 epochs. The initial learning rate is set to 10^{-3} and reduced to quarter at 60, 90, 120, 150 and

Data	Ratio	TVAL3		GSR		CSNet$^+$		RK-CCSNet		MR-CCSNet		MR-CCSNet$^+$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	1.5625%	19.00	0.4844	21.39	0.5815	24.45	0.6360	25.31	0.7033	25.72	0.7193	**25.79**	**0.7189**
	3.125%	19.89	0.5415	23.70	0.6822	27.19	0.7666	27.79	0.8061	28.19	0.8174	**28.27**	**0.8208**
	6.25%	22.03	0.6175	27.59	0.8163	28.68	0.8002	30.63	0.8799	31.10	0.8901	**31.25**	**0.8918**
	12.5%	23.75	0.7365	31.61	0.9016	33.55	0.9243	34.27	0.9393	35.03	0.9464	**35.16**	**0.9471**
	25%	27.39	0.8522	36.32	0.9510	37.69	0.9650	38.04	0.9712	39.24	0.9761	**39.37**	**0.9766**
	50%	33.11	0.9430	42.18	0.9908	42.54	0.9852	43.90	0.9901	45.07	0.9919	**45.11**	**0.9920**
Set14	1.5625%	16.79	0.3993	18.93	0.4399	22.78	0.5369	23.36	0.5917	23.61	0.5993	**23.69**	**0.6034**
	3.125%	18.40	0.4514	20.26	0.5184	24.96	0.6602	25.26	0.6914	25.56	0.6997	**25.63**	**0.7029**
	6.25%	19.65	0.5287	23.59	0.6526	26.33	0.7178	27.24	0.7836	27.91	0.7986	**28.00**	**0.7996**
	12.5%	21.03	0.6379	28.08	0.7915	30.12	0.8610	30.42	0.8798	30.97	0.8889	**31.06**	**0.8898**
	25%	22.69	0.7731	31.82	0.8939	33.81	0.9339	34.16	0.9443	35.04	0.9510	**35.11**	**0.9512**
	50%	26.61	0.9004	37.47	0.9619	38.59	0.9752	40.15	0.9837	41.21	0.9864	**41.25**	**0.9864**
Average		22.53	0.6555	28.58	0.7651	30.89	0.8135	31.71	0.8470	32.39	0.8554	**32.47**	**0.8567**

Table 1. Quantitative results on Set5 and Set14.

Data	Ratio	CSNet$^+$		RK-CCSNet		MR-CCSNet		MR-CCSNet$^+$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BSDS100	1.5625%	24.51	0.6344	25.02	0.6691	24.35	0.6775	**25.44**	**0.6791**
	3.125%	26.18	0.7102	26.51	0.7266	26.75	0.7334	**26.84**	**0.7361**
	6.25%	27.82	0.7728	28.08	0.7879	28.34	0.7949	**28.40**	**0.7952**
	12.5%	29.77	0.8424	29.98	0.8559	30.39	0.8632	**30.43**	**0.8639**
	25%	32.41	0.9073	32.68	0.9186	33.27	0.9251	**33.29**	**0.9253**
	50%	36.21	0.9582	37.29	0.9695	38.03	0.9731	**38.07**	**0.9732**
Average		29.48	0.8042	29.93	0.8213	30.19	0.8279	**30.41**	**0.8288**

Table 2. Quantitative results on BSDS100.

180 epochs respectively. Six sampling ratios, *i.e.* 1.5625%, 3.1250%, 6.2500%, 12.5000%, 25.0000%, and 50.0000% are investigated. PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural SIMilarity) [19] are chosen as the evaluation metrics. We implement the model using PyTorch, and train it on Nvidia RTX 2080Ti GPU.

4.2. Comparison with the state-of-the-arts

To verify the effectiveness of MR-CCSNet and MR-CCSNet$^+$ where the sensing network is GSM and GSM$^+$ respectively, we quantitatively and visually compare them with 4 state-of-the-art methods with available codes, which is **TVAL3** [23], **GSR** [41], **CSNet$^+$** [32], and **RK-CCSNet** [43]. The implementation codes of compared methods are download from the author's websites and all default values are used in experiments.

Quantitative comparisons In Tab. 1, we report the quantitative comparisons on Set5 and Set14. The best results are marked in bold font. The results show that MR-CCSNet and MR-CCSNet$^+$ are outperforms the four methods at all sampling ratios. Note that all DCS methods show a significant improvement comparing with the best traditional method, *i.e.* GSR. Specifically, our model achieve the best performance in low sampling ratios. In average, MR-CCSNet$^+$

outperforms TVAL3, GSR, CSNet$^+$, and RK-CCSNet by **9.94dB**, **3.89dB**, **1.58dB**, and **0.76dB** in terms of PSNR, respectively, on Set5 and Set14. In addition, the average SSIM of MR-CCSNet$^+$ can be improved **0.2012**, **0.0916**, **0.0432**, and **0.0097**, respectively. We further compare MR-CCSNet and MR-CCSNet$^+$ with CSNet$^+$ and RK-CCSNet on BSDS100. Tab. 2 show the evaluation results. It can be seen that both MR-CCSNet and MR-CCSNet$^+$ achieve a better reconstruction quality at all sampling ratios. Especially in the case of sampling ratio of 1.5625%, MR-CCSNet$^+$ outperforms CSNet$^+$ and RK-CCSNet by **0.93dB** and **0.48dB**, respectively. Finally, we compare the performance of MR-CCSNet and MR-CCSNet$^+$. When the sampling ratio is 50%, we observe that the PSNR and SSIM of them are very close. The reason is that GSM$^+$ degenerate into GSM in this case. As the sampling ratio decreases, MR-CCSNet$^+$ outperforms MR-CCSNet. The reason is that GSM$^+$ not only collects all level features, which is equivalent to GSM, but also extracts richer features for sampling and reconstructing. This is corresponding to our theoretical analysis in Sec. 3.1. All the experimental results demonstrate our model has state-of-the-art performance.

Visual comparisons We also visually compare our method with the state-of-the-art image CS methods. We
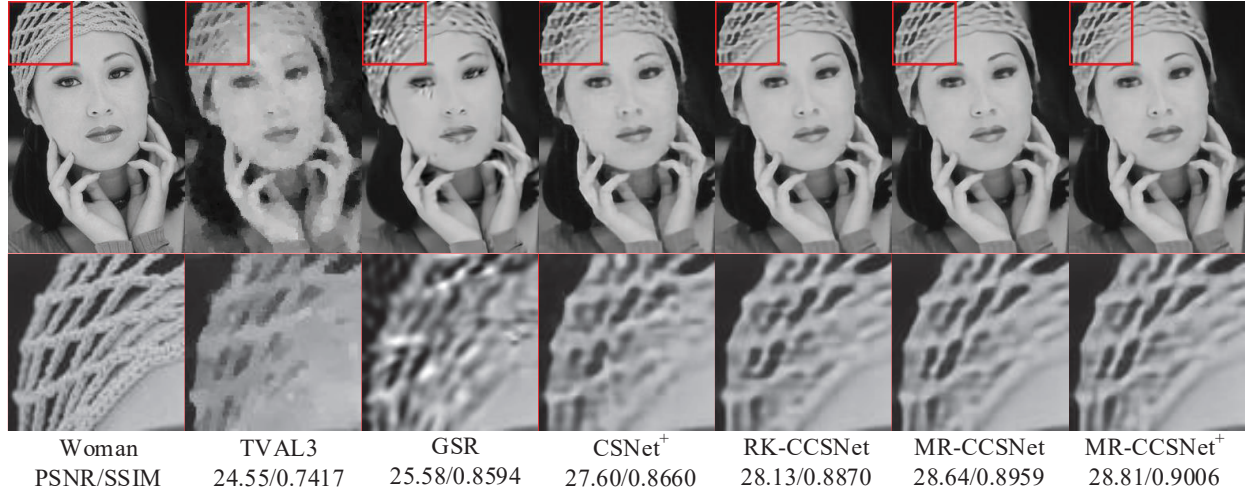
| Woman PSNR/SSIM | TVAL3 24.55/0.7417 | GSR 25.58/0.8594 | CSNet$^+$ 27.60/0.8660 | RK-CCSNet 28.13/0.8870 | MR-CCSNet 28.64/0.8959 | MR-CCSNet$^+$ 28.81/0.9006 |

Figure 6. Visual comparisons of reconstructed image on Woman from Set5 in the sampling ratio of 6.25%.

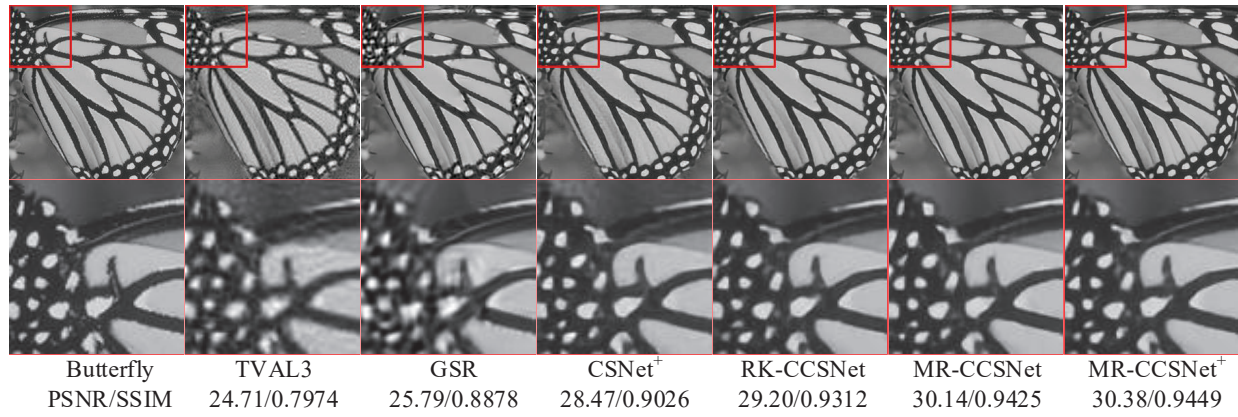| Butterfly PSNR/SSIM | TVAL3 24.71/0.7974 | GSR 25.79/0.8878 | CSNet$^+$ 28.47/0.9026 | RK-CCSNet 29.20/0.9312 | MR-CCSNet 30.14/0.9425 | MR-CCSNet$^+$ 30.38/0.9449 |

Figure 7. Visual comparisons of reconstructed image on Butterfly from Set5 in the sampling ratio of 12.5%.

magnify the results in order to compare the reconstruction details. Fig. 6 and Fig. 7 show the visual comparisons in the case of sampling ratio of 6.25% and 12.5%, respectively. We can see that DCS methods can achieve higher reconstruction quality than traditional methods in extremely low sampling ratios. In addition, our model also recover finer details than DCS methods CSNet$^+$ and RK-CCSNet. For example, in the figure of Butterfly and Woman, it is obvious that our model is able to reconstruct texture details, which is smoother and sharper than other methods. This is mainly because the measurements in our model contain all level features where the low and mid-level features relate to the edges and complex textures in the image. In addition, extracting richer features by utilizing measurements multiple times also plays an important role.

4.3. Running time comparison

The running time is important in many practical applications. Tab. 3 shows the average running time on GPU/CPU for reconstructing a 256×256 image. The running times of TVAL3 and GSR are taken from [22], and they are implemented on the platform of an Intel Core i7-3770 CPU. The running times of CSNet$^+$, RK-CCSNet, MR-CCSNet, and MR-CCSNet$^+$ are implemented on the platform of an Intel Core i9-9900k CPU plus a Nvidia RTX 2080Ti GPU. It is obvious that traditional methods take about seconds

Algorithm	sampling ratio=0.01		sampling ratio=0.1	
	CPU	GPU	CPU	GPU
TVAL3	2.3349	-	2.5871	-
GSR	235.6297	-	230.4755	-
CSNet$^+$	-	0.0075	-	0.0078
RK-CCSNet	-	0.0184	-	0.0181
MR-CCSNet	-	0.0284	-	0.0272
MR-CCSNet$^+$	-	0.0282	-	0.0271

Table 3. Average running time (in seconds) for reconstructing a 256 × 256 image.
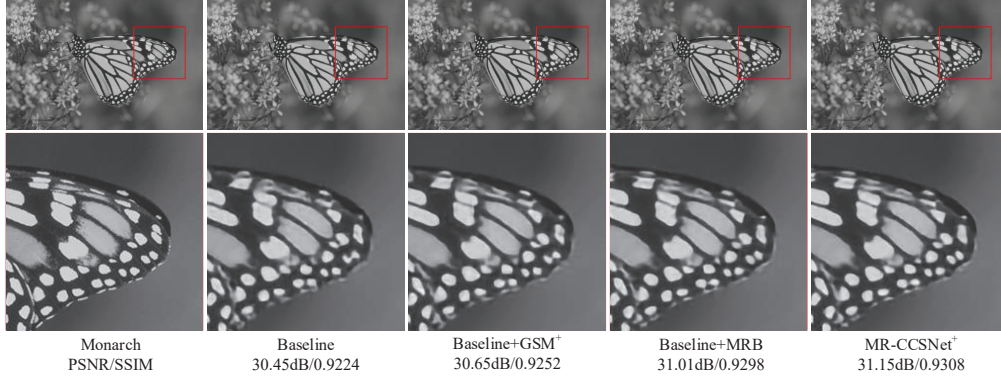
to minutes to reconstruct the image. This is because they

Figure 8. Visual comparisons of reconstructed image on Monarch from Set14 in the sampling ratio of 6.25%.

GSM$^+$	MRB	1.56250%		3.12500%		6.25000%		12.50000%		25.00000%		50.00000%	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
		25.02	0.6691	26.51	0.7266	28.08	0.7879	29.98	0.8559	32.68	0.9186	37.29	0.9695
✓		25.14	0.6737	26.58	0.7281	28.18	0.7915	30.15	0.8591	32.88	0.9211	37.71	0.9722
	✓	25.29	0.676	26.61	0.7307	28.26	0.7931	30.27	0.8614	33.01	0.9216	37.53	0.9706
✓	✓	25.49	0.6811	26.88	0.7359	28.38	0.7955	30.36	0.8629	33.24	0.9248	37.98	0.9730

Table 4. The ablation studies of MR-CCSNet$^+$ on BSDS100.

need multiple iterative operations during reconstruction. By comparison, the running time of DCS methods are improved by several orders of magnitude. The reason why our model slower than CSNet$^+$ and RK-CCSNet is MR-CCSNet$^+$ has more parameters. But it is more fast compared with traditional methods and achieves better reconstruction quality. We can see that the running time of MR-CCSNet and MR-CCSNet$^+$ are equal. This is because two models have approximately the same number of parameters.

4.4. Ablation studies

In order to verify the efficacy of GSM$^+$ and MRB, we further conduct ablation studies on BSDS100. The models compared incude: Baseline (RK-CCSNet), Baseline with GSM$^+$, Baseline with MRB, and MR-CCSNet$^+$. From the results, as shown in Tab. 4, we can observe that:

(1) Both GSM$^+$ and MRB are effective for improving the performance of reconstruction quality. This may be because GSM$^+$ can preserve more features in the image, and MRB can extract richer features for image reconstruction.

(2) When the sampling ratio is low, MRB plays a more important role than GSM$^+$ for image reconstruction. Alternatively, GSM$^+$ plays a more important role than MRB when the sampling ratio is 50%.

We also visually compare results of these four models, as shown in Fig. 8. The results is corresponding to our theoretical analysis. When the reconstruction algorithm is fixed, because GSM$^+$ takes advantage of the hierarchical nature of the network, the texture details of Baseline with GSM$^+$ smoother and sharper than Baseline. When the sensing net-

work is fixed, because we utilize measurements in a deep manner, Baseline with MRB outperform Baseline.

5. Conclusion and future work

In this paper, we propose Global Sensing Module and Measurements Reuse Block for image CS. GSM can take advantage of the hierarchical nature of the network for sampling. MRB can make full use of the measurements for improving the reconstructed image quality. In the experiments, we show that our model significantly and consistently outperforms state-of-the-art image CS methods. In particular, our methods also have good performances in extremely low sampling ratios. In addition, we demonstrate that GSM and MRB are effective by ablation studies.

In the future, we will explore the following directions:

(1) In the sensing network, pooling operation loses information about the low-level features. We will explore a more effective way for collecting all level features.

(2) Attention mechanism can effectively help us in extracting matching features from measurements. We are interested in adding attention mechanism into MRB to improve its performance.

(3) In the real-world, because there are noise in the measurements, using them multiple times will introduce noise in the reconstruction process. We will explore how to improve the robustness for using measurements multiple times.

References

[1] Arash Amini and Farokh Marvasti. Deterministic construction of binary, bipolar, and ternary compressed sensing matrices. *IEEE Transactions on Information Theory*, 57(4):2360–2370, 2011. 1

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 2, 5

[3] Mario Bertero and Patrizia Boccacci. *Introduction to inverse problems in imaging*. CRC press, 2020. 2

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 2, 5

[5] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008. 2

[6] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006. 1

[7] Chen Chen, Eric W Tramel, and James E Fowler. Compressed-sensing recovery of images and video using multihypothesis predictions. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 1193–1198. IEEE, 2011. 1, 2

[8] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001. 2

[9] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004. 2

[10] Khanh Quoc Dinh, Hiuk Jae Shim, and Byeungwoo Jeon. Measurement coding for compressive imaging using a structural measuremnet matrix. In *2013 IEEE International Conference on Image Processing*, pages 10–13. IEEE, 2013. 1

[11] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 1

[12] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008. 1

[13] Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in signal processing*, 1(4):586–597, 2007. 2

[14] James E Fowler, Sungkwang Mun, and Eric W Tramel. Multiscale block compressed sensing with smoothed projected landweber reconstruction. In *2011 19th European Signal Processing Conference*, pages 564–568. IEEE, 2011. 2

[15] Lu Gan. Block compressed sensing of natural images. In *2007 15th International conference on digital signal processing*, pages 403–406. IEEE, 2007. 1, 2

[16] Xinwei Gao, Jian Zhang, Wenbin Che, Xiaopeng Fan, and Debin Zhao. Block-based compressive sensing coding of natural images by local structural measurement matrix. In *2015 Data Compression Conference*, pages 133–142. IEEE, 2015. 1

[17] Jarvis Haupt and Robert Nowak. Signal reconstruction from noisy random projections. *IEEE Transactions on Information Theory*, 52(9):4036–4048, 2006. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4

[19] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 2, 6

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[22] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2016. 3, 7

[23] C Li, W Yin, and Y Zhang. Tval3: Tv minimization by augmented lagrangian and alternating direction agorithm 2009, 2013. 1, 2, 6

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[25] Weizhi Lu, Tao Dai, and Shu-Tao Xia. Binary matrices for compressed sensing. *IEEE Transactions on Signal Processing*, 66(1):77–85, 2017. 1

[26] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993. 2

[27] Ali Mousavi and Richard G Baraniuk. Learning to invert: Signal recovery via deep convolutional networks. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2272–2276. IEEE, 2017. 1, 3

[28] Ali Mousavi, Gautam Dasarathy, and Richard G Baraniuk. A data-driven and distributed approach to sparse signal representation and recovery. In *International Conference on Learning Representations*, 2018. 1, 3

[29] Ali Mousavi, Ankit B Patel, and Richard G Baraniuk. A deep learning approach to structured signal recovery. In *2015 53rd annual allerton conference on communication, control, and computing (Allerton)*, pages 1336–1343. IEEE, 2015. 3

[30] Sungkwang Mun and James E Fowler. Block compressed sensing of images using directional transforms. In *2009 16th IEEE international conference on image processing (ICIP)*, pages 3021–3024. IEEE, 2009. 1

[31] Sungkwang Mun and James E Fowler. Residual reconstruction for block-based compressed sensing of video. In *2011 Data Compression Conference*, pages 183–192. IEEE, 2011. 2

[32] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Image compressed sensing using convolutional neural network. *IEEE Transactions on Image Processing*, 29:375–388, 2019. 1, 2, 3, 6

[33] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Scalable convolutional neural network for image compressed sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12290–12299, 2019. 3

[34] Yubao Sun, Jiwei Chen, Qingshan Liu, Bo Liu, and Guodong Guo. Dual-path attention network for compressed sensing image reconstruction. *IEEE Transactions on Image Processing*, 29:9482–9495, 2020. 1

[35] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007. 2

[36] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on signal processing*, 57(7):2479–2493, 2009. 2

[37] Kai Xu, Zhikang Zhang, and Fengbo Ren. Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 485–500, 2018. 3

[38] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1, 4

[39] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 2, 5

[40] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018. 1

[41] Jian Zhang, Debin Zhao, and Wen Gao. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*, 23(8):3336–3351, 2014. 1, 2, 6

[42] Jian Zhang, Debin Zhao, Chen Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. Image compressive sensing recovery via collaborative sparsity. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(3):380–391, 2012. 1

[43] Runkai Zheng, Yinqi Zhang, Daolang Huang, and Qingliang Chen. Sequential convolution and runge-kutta residual architecture for image compressed sensing. In *European Conference on Computer Vision*, pages 232–248. Springer, 2020. 1, 2, 3, 5, 6