This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Memory-Efficient Network for Large-scale Video Compressive Sensing

Ziheng Cheng, Bo Chen*, Guanliang Liu, Hao Zhang, Ruiying Lu and Zhengjue Wang
Xidian University
zhcheng@stu.xidian.edu.cn, bchen@xidian.edu.cn
{lgl_xidian,zhanghao_xidian,ruiyinglu_xidian,zhengjuewang}@163.com

Xin Yuan*
Bell Labs
xyuan@bell-labs.com

Abstract

Video snapshot compressive imaging (SCI) captures a sequence of video frames in a single shot using a 2D detector. The underlying principle is that during one exposure time, different masks are imposed on the high-speed scene to form a compressed measurement. With the knowledge of masks, optimization algorithms or deep learning methods are employed to reconstruct the desired high-speed video frames from this snapshot measurement. Unfortunately, though these methods can achieve decent results, the long running time of optimization algorithms or huge training memory occupation of deep networks still preclude them in practical applications. In this paper, we develop a memory-efficient network for large-scale video SCI based on ***multi-group reversible 3D*** *convolutional neural networks. In addition to the basic model for the grayscale SCI system, we take one step further to combine demosaicing and SCI reconstruction to directly recover color video from Bayer measurements. Extensive results on both simulation and real data captured by SCI cameras demonstrate that our proposed model outperforms previous state-of-the-art with less memory and thus can be used in large-scale problems. The code is at* https://github.com/BoChenGroup/RevSCI-net.

1. Introduction

Computational imaging (CI) [1, 34] introduces modulation (coding) in the optical path to advance the capability of traditional cameras. Snapshot compressive imaging (SCI) [11, 25, 40, 43, 51] is a promising CI technique that indirectly captures 3-dimensional (3D) data using a 2D detector, *i.e.*, the original 3-dimensional data (videos or hyperspectral images) are coded by different masks and then integrated into a single frame (measurement). As shown in Fig. 1, in video SCI, the temporal dimension is modulated and compressed, which avoids large memory storage and transmission bandwidth during imaging. To make the SCI system practical, an efficient reconstruction algorithm, *i.e.*, recovering the desired images from the compressed measurement is critical. In this work, we focus on the practical video SCI reconstruction algorithm that can scale to large data.

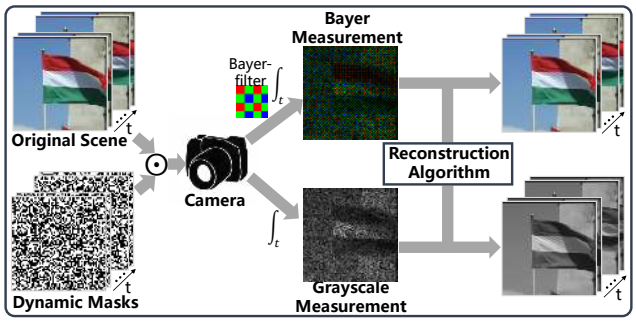

Figure 1. Principle of grayscale or color video SCI system. The original scene is modulated by dynamic masks and then is integrated by the camera to obtain a snapshot measurement. Note that for the color video SCI system, the camera captures the modulated scene through the Bayer-filter not directly collecting the brightness. Having obtained measurements, the reconstruction algorithm recovers the desired video from it.

The mainstream of reconstruction methods is the model-based optimization problems with various prior knowledge, *e.g.*, total variation (TV) used in GAP-TV [50] and TwIST [2], and non-local low-rank [9] used in DeSCI [24]. These methods can provide usable results in an unsupervised manner but cannot balance the reconstruction quality and speed (hours for DeSCI to reconstruct a $256 \times 256 \times 8$ video from a single measurement), which makes them un-

* Corresponding authors.

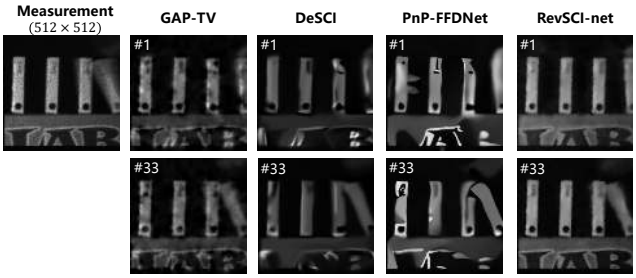

Figure 2. Reconstruction results using the proposed RevSCI-net on large scale (512×512×50) real data captured by [39]. Note that this is the first deep learning model can perform a compression rate at B=50. Videos are shown in the supplementary material (SM).

realistic for real applications. Inspired by deep learning and rich datasets, some researchers develop deep neural networks (DNNs) [4, 13, 33, 39, 47] or combine DNN with optimization methods, such as deep unfolding technique [21, 27, 30] and plug-and-play algorithms [52, 56], to reconstruct desired 3D data from SCI measurements. Benefiting from the efficient feed forward networks, DNN-based algorithms significantly decrease the inference time to less than one second. Most recently, BIRNAT [4], which develops a bidirectional recurrent neural network, has led to state-of-the-art reconstruction results. Most researches on video SCI reconstruction usually verify the performance on a low-resolution situation (less than 512×512). With the high-resolution images widely used in our daily life (HD with 1280×720 and FHD with 1920×1080), however, the squarely increasing pixels will significantly increase the running time and training memory consumption for DNNs. Although some methods can provide superior reconstructions, *e.g.*, DeSCI and BIRNAT, unpractical running time or GPU memory consumption still precludes them in the practical large-scale SCI system applications.

Bearing the above concerns in mind, in this paper, we propose an *end-to-end reversible 3D* convolutional neural network (CNN) for SCI reconstruction named RevSCI-net, in which 3D convolutional kernel jointly explores the *spatial and temporal correlation* within the desired data. Meanwhile, the reversible structure allows activations not to be stored in memory. The main contributions of our work are summarized as follows:

- Since the desired signal of video SCI is 3D, we build an end-to-end 3D CNN paradigm for video SCI reconstruction which jointly explores the spatial and temporal correlation of video frames by the 3D convolutional kernel. To our best knowledge, this is the first time that 3D CNN is applied in SCI problems.

- We propose the multi-channel reversible CNN in the proposed network with less memory occupation during training. Benefit by this, we can reconstruct a $512 \times 512 \times 50$ video from a snapshot measurement, with an example shown in Fig. 2, where a compression rate of 50 is achieved. This is the first deep learning results that accomplish this high spatio-temporal resolution.

- We combine SCI reconstruction and demosaicing for color SCI systems into a single end-to-end network.

- In addition to the widely used grayscale test sets, we also conduct simulation on large-scale color datasets. Furthermore, we verify the proposed network on the real data (captured by SCI cameras). Only our model can recover large scale and high compression rate SCI measurements compared with other DNN based methods thanks to the memory-efficient structure.

The rest of this paper is organized as follows. Sec. 2 briefly reviews the related work. Sec. 3 presents the mathematical model of video SCI. Sec. 4 details our proposed model for grayscale and color video SCI reconstruction. Sec. 5 presents extensive results including simulation and real data. Sec. 6 concludes the entire paper.

2. Related Work

Video Snapshot Compressive Imaging Many different SCI hardware systems have been developed, by modulating the light in different approaches, *e.g.*, usually a digital micromirror device (DMD) [11, 28, 37, 38, 39, 40, 41] or a physical mask [25, 53]. Although hardware systems are mature in the laboratory, existing reconstruction algorithms are still far from real applications. Model-based optimization methods, *e.g.*, GAP-TV [50], GMM [48, 49], DeSCI [24], and PnP-FFDNet [52] consume high computational cost leading to long time reconstruction. Recently, some researchers have attempted to use deep learning in computational imaging [13, 20, 30, 33, 39, 47, 54]. Various networks have been proposed for SCI reconstruction, and significantly reduced the running time. However, these networks usually need a huge memory and long time for training. For instance, state-of-the-art method BIRNAT [4] requires more than 32GB GPU memory (batch size is 3 and costs weeks for training) to train the model of size $256 \times 256 \times 8$. Such a memory unfriendly model is not satisfying the increasing resolution in daily life, where HD and UHD videos are becoming widely used. Different from previous methods, in this work, we develop a 3D CNN based network and introduce the reversible structure to reduce the training memory without loss of performance.

Reversible neural network Flow-based generative models, *e.g.*, NICE [5], real NVP [6], and Glow [17] can jointly perform generation and inference using a shared stacked reversible structure. This means that the generative process can be easily inverted, and the inference process can be

computed by the inverse of the generation function. Specifically, for l-th blocks, given an input h^l, divided it into two-parts h_1^l, h_2^l, NICE [5] performs the simple additive affine transformations:

$$h_1^{l+1} = h_1^l, \quad h_2^{l+1} = h_2^l + m(h_1^l), \tag{1}$$

where $m(\,)$ is an arbitrary function. The output is the concatenation of h_1^{l+1} and h_2^{l+1}. The inverse transformation can be easily computed by

$$h_2^l = h_2^{l+1} - m(h_1^{l+1}), \quad h_1^l = h_1^{l+1}. \tag{2}$$

Inspired by this simple and effective setting, Rev-Net [8] introduces this idea into Res-Net [10], which has similar performance with Res-Net in the classification task and each block includes several reversible layers. The main strength of Rev-Net is that training such a network *does not need to save the middle activation produced by each layer*, which occupies most of the memory. During back-propagation, the previous layer activation can be easily computed by the reversible transformation to calculate the gradient. Therefore, saving the last activation of the stacked reversible layers allows learning the parameters, which makes the memory cost reduce from $O(L)$ to $O(1)$ (L is the number of the layer). A memory-efficient learning procedure [15] inspired by the reversible networks was proposed for unfolding networks, which is easy to act on the unfolding network to reduce the training memory without loss of accuracy. Most recently, researchers [42] have proved that flow models based on affine coupling can be universal distributional approximations.

One of the bottlenecks for the SCI reconstruction network applied in the large-scale scene is the huge GPU memory consumption as mentioned before, because the squarely increasing pixel numbers (for a larger size) make it impossible for high-resolution scenes. Inspired by the Rev-net [8], we propose a reversible 3D CNN for large-scale video SCI reconstruction. Specifically, we extend the original two branches additive affine transformations into *multi-group transformations*. The 3D CNN will also capture the spatio-temporal correlations in the desired video, and the reconstruction results will be more consistent in different frames.

Demosaicing For color imaging, common devices usually first capture pixels by a color filter (one pixel only sampling one color energy such as red, green or blue) and then impose an interpolation algorithm to achieve a color (usually RGB) image. This process is called demosaicing. Recently, some researchs [3, 18, 23] developed an end-to-end network to directly obtain a color image from the raw captured image. Motivated by this, we extend the proposed RevSCI-net to joint demosaicing and reconstruction for the color SCI system. To our best knowledge, this is the first attempt to use a unified end-to-end deep model to directly restore an RGB video from a compressive measurement in SCI.

3. Video Snapshot Compressive Imaging

In video SCI, a dynamic scene consisting of B high-speed two-dimensional frames $\{\mathbf{X}_k\}_{k=1}^B \in \mathbb{R}^{n_x \times n_y}$ are modulated by the coding patterns (masks) $\{\mathbf{C}_k\}_{k=1}^B \in \mathbb{R}^{n_x \times n_y}$, respectively. These coded frames are then integrated over time on a camera, forming a *compressed* coded measurement (Fig. 1). The measurement $\mathbf{Y} \in \mathbb{R}^{n_x \times n_y}$ is given by

$$\mathbf{Y} = \textstyle\sum_{k=1}^B \mathbf{X}_k \odot \mathbf{C}_k + \mathbf{G}\,, \tag{3}$$

where $\odot$ denotes the Hadamard (element-wise) product and $\mathbf{G} \in \mathbb{R}^{n_x \times n_y}$ represents the noise. From a pixel perspective, any B pixel (in the B frames) at position (i, j), $i = 1, \ldots, n_x$; $j = 1, \ldots, n_y$ are collapsed to form one pixel in the snapshot measurement by

$$y_{i,j} = \textstyle\sum_{k=1}^B c_{i,j,k} x_{i,j,k} + g_{i,j}\,. \tag{4}$$

Define $\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_B^\top\right]$, where $\boldsymbol{x}_k = \mathrm{vec}(\mathbf{X}_k)$; let $\mathbf{D}_k = \mathrm{diag}(\mathrm{vec}(\mathbf{C}_k))$, for $k = 1, \ldots, B$, where $\mathrm{vec}(\,)$ vectorizes the matrix inside $(\,)$ by stacking the columns and $\mathrm{diag}(\,)$ diagonalizes the ensued vector into a diagonal matrix. The video SCI sensing process can be written as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{g}\,, \tag{5}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n \times nB}$ is the sensing matrix with $n = n_x n_y$, $\boldsymbol{x} \in \mathbb{R}^{nB}$ is the desired signal, and $\boldsymbol{g} \in \mathbb{R}^n$ again denotes the vectorized noise. Different from single-pixel imaging [7], the sensing matrix $\boldsymbol{\Phi}$ in (5) has a very special structure and can be written as

$$\boldsymbol{\Phi} = [\mathbf{D}_1, \ldots, \mathbf{D}_B]\,, \tag{6}$$

where $\{\mathbf{D}_k\}_{k=1}^B \in \mathbb{R}^{n \times n}$ are diagonal matrices of masks. Therefore, the compressive sampling rate in SCI is equal to $1/B$. Recently, researchers [14] proved that high quality reconstruction is achievable when $B > 1$.

In terms of color video SCI system, we consider the Bayer pattern filter sensor, where each pixel only captures the red (R), green (G) or blue (B) channel in a spatial layout such as 'RGGB'. Note that two green channels are used due to the sensitivity of the human eyes. In this case, $\mathbf{X}_k$ is a mosaic frame and since the neighbouring pixels are sampling different color components, the values are not necessarily continuous. To cope with this issue, previous studies [24, 52, 53] usually divide the original measurement $\mathbf{Y}$ into four-channel sub-measurements corresponding to the Bayer-filter $\{\mathbf{Y}^r, \mathbf{Y}^{g1}, \mathbf{Y}^{g2}, \mathbf{Y}^b\} \in \mathbb{R}^{\frac{n_x}{2} \times \frac{n_y}{2}}$ for the R, G1,

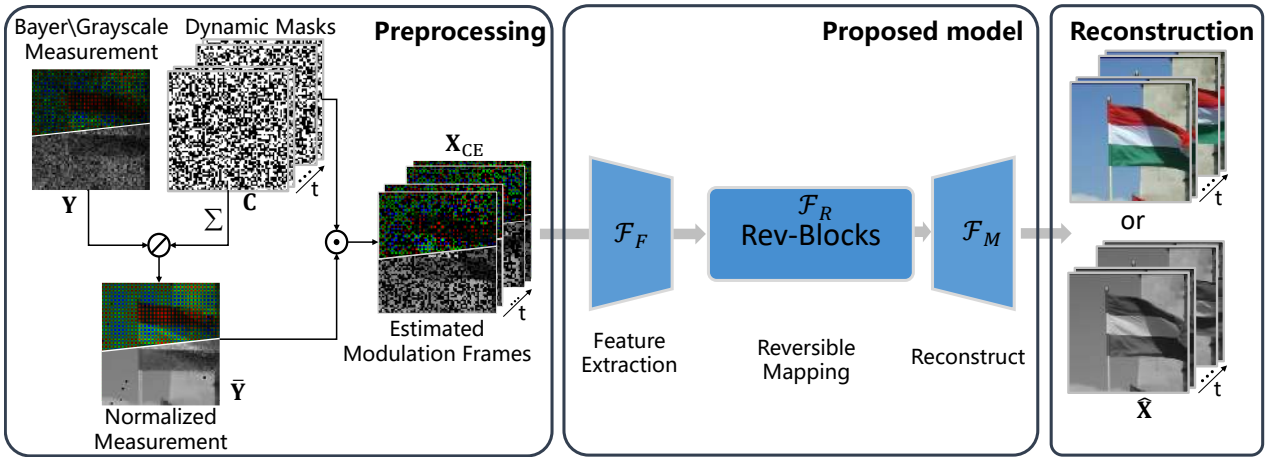

Figure 3. Proposed reconstruction pipeline. Left: the preprocessing stage to obtain the estimate of the modulated frames as the network input, which includes the information of coding masks and the normalized measurement. Middle: the RevSCI-net which concludes three parts, feature extraction, reversible non-linear mapping, and reconstruction. Right: the reconstruction video.

G2 and B components. Similarly, the mask and desired signal are also divided into four components. They reconstruct each sub-signal separately using the corresponding measurement and mask and then perform demosaicing (using off-the-shelf tools) in the recovered sub-videos to generate the final color (RGB) video.

4. The Proposed Model

Given the compressed measurement $\mathbf{Y}$ and coding pattern $\{\mathbf{C}_k\}_{k=1}^B$ captured by the SCI system, the goal of the proposed model RevSCI-net is to predict the desired high-speed frames $\{\mathbf{X}_k\}_{k=1}^B$, in other words, to learn a mapping from $\mathbf{Y}$ to $\{\mathbf{X}_k\}_{k=1}^B$. In this section, the details of the model will be described. Overall, our proposed model consists of three parts as shown in the middle of Fig. 3: 1) The feature extractor $\mathcal{F}_F$ uses several 3D CNN layers to capture the high-dimensional features from the input. 2) Feature level nonlinear mapping employs several reversible blocks $\mathcal{F}_R$ to transform the input features into the desired reconstruction domain. 3) The reconstructor $\mathcal{F}_M$ integrates the features to reconstruct the final video.

4.1. Model for Grayscale SCI system

4.1.1 Feature Extraction

Considering the measurement $\mathbf{Y}$ being a 2D matrix, we first normalize the original measurement and then combine masks and the normalized measurement to produce coarse estimates of modulated frames as follows:

$$\overline{\mathbf{Y}} = \mathbf{Y} \oslash \sum_{k=1}^B \mathbf{C}_k, \quad \mathbf{X}_{CE} = \overline{\mathbf{Y}} \odot \mathbf{C}, \tag{7}$$

where $\oslash$ denotes the matrix dot (element-wise) division, and coarse estimates $\mathbf{X}_{CE} \in \mathbb{R}^{B\times n_x \times n_y}$.

After obtaining $\mathbf{X}_{CE}$, we employ four 3D convolutional layers expressed as $\mathcal{F}_F$ (the kernel size is $5\times5\times5, 3\times3\times3, 1\times1\times1$, and $3\times3\times3$) to extract the feature as:

$$\mathbf{H}_f = \mathcal{F}_F(\mathbf{X}_{CE}), \tag{8}$$

where $\mathbf{H}_f \in \mathbb{R}^{c_1\times B\times n_x\times n_y}$ is a 4D tensor and c_1 is the channel number. Here, we set the stride of the final layer to 2, which reduces the resolution of the feature map by half to reduce the computational complexity. We apply the LeakyReLU [29] on each convolutional layer, and do not use the batch normalization following previous research on image deburring [35, 22] and video SCI [4]. After the feature extraction operation, we obtain the coarse features of the input modulated frames.

4.1.2 Reversible Non-linear Mapping

Having obtained the features of the input, we use stacked reversible blocks to transform them to the video domain features. The original reversible block in Rev-Net [8] splits the input features into two parts by channel, and the transformation is:

$$\boldsymbol{h}_1^{l+1} = \boldsymbol{h}_1^l + \mathcal{F}(\boldsymbol{h}_2^l), \quad \boldsymbol{h}_2^{l+1} = \boldsymbol{h}_2^l + \mathcal{G}(\boldsymbol{h}_1^{l+1}), \tag{9}$$

where $\boldsymbol{h}_1^l, \boldsymbol{h}_2^l, \boldsymbol{h}_1^{l+1}, \boldsymbol{h}_2^{l+1} \in \mathbb{R}^{\frac{c_1}{2}\times B\times n_x\times n_y}$, and $\mathcal{F}$ and $\mathcal{G}$ are arbitrary functions. Inspired by the group convolution [19, 46], we modify the formulation and extend it to a multi-group reversible transformation. As shown in Fig. 4(c), we split the feature into multiple parts, and the

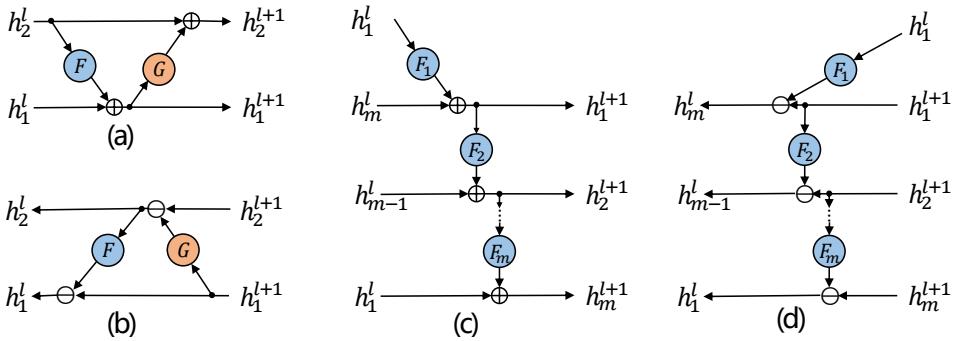

Figure 4. (a) and (b) are the forward and the reverse computations of the original reversible layer in Rev-Net [8], respectively. (c) and (d) are the forward and the reverse process of the proposed multi-group reversible block, respectively.

forward function is now:

$$\begin{aligned} \boldsymbol{h}_1^{l+1} &= \boldsymbol{h}_m^l + \mathcal{F}_1(\boldsymbol{h}_1^l), \\ \boldsymbol{h}_2^{l+1} &= \boldsymbol{h}_{m-1}^l + \mathcal{F}_2(\boldsymbol{h}_1^{l+1}), \\ &\vdots \\ \boldsymbol{h}_m^{l+1} &= \boldsymbol{h}_1^l + \mathcal{F}_m(\boldsymbol{h}_{m-1}^{l+1}), \end{aligned} \tag{10}$$

where m is the number of groups, and $\mathcal{F}_*$ can be an arbitrary function. In our experiments, we set it to two 3D convolutional layers with the kernel size of $3 \times 3 \times 3$. With the additional dimension on the group, we extend the original reversible form, and experimental results show that these changes have improved the performance. The inverse of the multi-group reversible transformation is thus

$$\begin{aligned} \boldsymbol{h}_1^l &= \boldsymbol{h}_m^{l+1} - \mathcal{F}_m(\boldsymbol{h}_{m-1}^{l+1}), \\ \boldsymbol{h}_2^l &= \boldsymbol{h}_{m-1}^{l+1} - \mathcal{F}_{m-1}(\boldsymbol{h}_{m-2}^{l+1}), \\ &\vdots \\ \boldsymbol{h}_m^l &= \boldsymbol{h}_1^{l+1} - \mathcal{F}_1(\boldsymbol{h}_1^l). \end{aligned} \tag{11}$$

The input feature $\mathbf{H}_f$ will be transformed by $\mathcal{F}_R$ (stacking L reversible blocks) into the reconstruction domain feature $\mathbf{H}_r$ as:

$$\mathbf{H}_r = \mathcal{F}_R(\mathbf{H}_f). \tag{12}$$

Note that during the back-propagation, we only save the last activation in $\mathcal{F}_R$, and activations of others can be computed by the (11) so that calculate the gradient to update the network parameters by the chain rule. For traditional convolutional layers, adding more layers to a certain extent is beneficial for the non-linearity and the performance, but it will significantly increase the activation memory of the model. Fortunately, due to the reversible structure, adding the number of layers will not increase the memory cost of activations in RevSCI-net.

4.1.3 Reconstruction

After the reversible non-linear transformation, the goal of the reconstruction stage is to integrate the features to obtain the desired video. We utilize four 3D convolutional layers (with the kernel size of $3 \times 3 \times 3, 3 \times 3 \times 3, 1 \times 1 \times 1$, and $3 \times 3 \times 3$) to reduce the channel to one and achieve the final reconstruction video, *i.e.*,

$$\widehat{\mathbf{X}} = \mathcal{F}_M(\mathbf{H}_r). \tag{13}$$

4.2. Model for Color SCI System

As mentioned before, color SCI systems capture the mosaic Bayer measurement as shown in Fig. 1. Inspired by the success of deep learning demosaicing and grayscale SCI reconstruction respectively, we conduct joint demosaicing and reconstruction using the proposed model.

To avoid mixture of different color channels, we first separate the coarse estimates of modulated frames obtained by (7) into four individual parts corresponding to the Bayer-filter, one for red, one for blue, and two for green,

$$\begin{aligned} \mathbf{X}_{CE}^{color} = [&\overline{\mathbf{Y}}^r \odot \mathbf{C}_1^r, ..., \overline{\mathbf{Y}}^r \odot \mathbf{C}_B^r; \\ &\overline{\mathbf{Y}}^{g1} \odot \mathbf{C}_1^{g1}, ..., \overline{\mathbf{Y}}^{g1} \odot \mathbf{C}_B^{g1}; \\ &\overline{\mathbf{Y}}^{g2} \odot \mathbf{C}_1^{g2}, ..., \overline{\mathbf{Y}}^{g2} \odot \mathbf{C}_B^{g2}; \\ &\overline{\mathbf{Y}}^{b} \odot \mathbf{C}_1^{b}, ..., \overline{\mathbf{Y}}^{b} \odot \mathbf{C}_B^{b}]_3, \end{aligned} \tag{14}$$

where $\mathbf{X}_{CE}^{color} \in \mathbb{R}^{4 \times B \times \frac{n_x}{2} \times \frac{n_y}{2}}$ includes four color channel modulation information and superscripts r, g and b denote the red, green and blue channels, respectively.

These color independent estimates $\mathbf{X}_{CE}^{color}$, are fed into the network. Because of the differences on the channel and the spatial resolution of the input compared with the grayscale SCI, we change the number of kernels on the first convolutional layer, and set the stride to 1 on the feature extraction stage to keep the resolution unchanged. For reconstruction, because the color image is 3 channels, we modify the number of the kernel on the last convolutional layer. In this manner, we extend RevSCI-net to directly obtain an RGB color video from the Bayer measurement.

4.3. Training

4.3.1 Loss Function

We jointly train our proposed model with mean square error (MSE) loss, *i.e*.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{c\mathbf{B}n_x n_y} \sum_{k=1}^{B} ||\widehat{\mathbf{X}}_k - \mathbf{X}_k||_2^2 , \tag{15}$$

where $\widehat{\mathbf{X}}_k$ is the final reconstruction from RevSCI-net, and $\mathbf{X}_k$ is the ground-truth; c is the channel number of $\widehat{\mathbf{X}}_k$, one for grayscale image and three for RGB image.

4.3.2 Back-propagation

Note that we do not directly use the automatic differentiation routine, *e.g*., `Loss.backward()` in PyTorch, to calculate the gradient of parameters because this will save all activations during the forward propagation and thus costs a huge memory. Instead, for the forward pass, we directly obtain the desired reconstruction without storing the activations of reversible blocks except the last one. As mentioned before, for back-propagation, due to the reversible block, we calculate the previous layer activation to compute the gradient of the parameters using the chain rule; for the feature extraction and reconstruction stage, we calculate the gradient as usual. Thereby, during training, we only save the full activations of the feature extraction stage and the reconstruction stage (each has only four layers), and the last layer of reversible blocks whatever the number of blocks.

5. Experiments

In this section, we compare RevSCI-net with several state-of-the-art methods on both simulation datasets and real data captured by two different video SCI cameras.

5.1. Data sets and Experimental Setting

Training and testing datasets Following [4], we choose the data set `DAVIS2017` [36] as the training set for all experiments. `DAVIS2017` has 90 different scenes in total 6208 frames with two resolutions: 480×894 and 1080×1920.

To demonstrate the quantitative performance, we first evaluate RevSCI-net on six widely used grayscale simulation data sets including `Kobe`, `Runner`, `Drop`, `Traffic` [24], `Aerial` and `Vehicle` [52]. The resolution of these test sets is 256×256. We follow the setting in [24], eight sequential ($B = 8$) frames are modulated by the shifting binary random masks $\{\mathbf{C}_k\}_{k=1}^{B}$ and then collapsed into a single measurement $\mathbf{Y}$. Under this setting, we randomly crop patch cubes ($256 \times 256 \times 8$) from the original scenes in `DAVIS2017`, and obtain 26000 training data pairs with data augmentation.

In addition, we evaluate RevSCI-net on the RGB large-scale scene, *e.g*., `Messi` and `Hummingbird` [52] with a resolution of $1080 \times 1920 \times 3$ (here 3 denotes the RGB channels) and 24 sequential frames are modulated and integrated into a single Bayer measurement by the shifting binary random masks. We generate 2000 data pairs for training from `DAVIS2017` with the resolution of $1080 \times 1920 \times 3$.

Lastly, we evaluate RevSCI-net on the measurements captured by two real SCI systems [25, 37].

Implementation details We jointly train RevSCI-net on the RTX 2080Ti GPU for 100 epochs using PyTorch. Adam optimizer [16] is used to minimize the loss function with the starting learning rate of 2×10^{-4}. Then, we reduce the learning rate by 5% every 10 epochs. It takes about a week to train the entire network. The detailed architecture for RevSCI-net is given in the supplement material (SM).

Counterparts and Performance Metrics We compare RevSCI-net with five competitive counterparts: two iterative optimization methods – GAP-TV [50] and DeSCI [24], and three methods using deep learning – the plug-and-play method PnP-FFDNet [52] integrated the deep denoiser as a prior, E2E-CNN [39] which is a deep CNN model, and BIRNAT [4] which builds a bidirectional RNN and produces current state-of-the-art results. For the simulation datasets, both peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM) [44] are used as metrics to evaluate the reconstruction quality. Besides, we give the running time at the testing stage which determines the usability of the method in real applications.

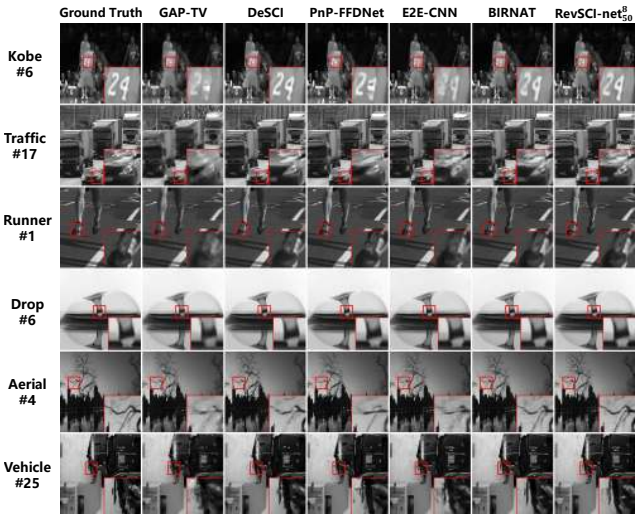

Figure 5. Selected reconstruction frames of six grayscale benchmark datasets.

Table 1. The average results of PSNR in dB (left entry), SSIM (right entry) and running time per measurement/shot in seconds by different algorithms on six grayscale benchmark datasets. The best results are **bold**, and the second best results are underline.

Algorithm	Kobe	Traffic	Runner	Drop	Aerial	Vehicle	Average	Time
GAP-TV	26.45, 0.845	20.89, 0.715	28.81, 0.909	34.74, 0.970	25.05, 0.828	24.82, 0.838	26.79, 0.858	4.2
DeSCI	33.25, 0.952	28.72, 0.925	38.76, 0.969	**43.22**, **0.993**	25.33, 0.860	27.04, 0.909	32.72, 0.935	6180
PnP-FFDNet	30.50, 0.926	24.18, 0.828	32.15, 0.933	40.70, 0.989	25.27, 0.829	25.42, 0.849	29.70, 0.892	3.0
E2E-CNN	29.02, 0.861	23.45, 0.838	34.43, 0.958	36.77, 0.974	27.52, 0.882	26.40, 0.886	29.26, 0.900	0.023
BIRNAT	32.71, 0.950	29.33, 0.942	38.70, 0.976	42.28, 0.992	28.99, 0.927	27.84, 0.927	33.31, 0.951	0.16
RevSCI-net$^{8}_{50}$	**33.72**,**0.957**	**30.02**, **0.949**	**39.40**, **0.977**	42.93, 0.992	**29.35**, **0.924**	**28.12**, **0.937**	**33.92**, **0.956**	0.19

5.2. Results on Simulation Datasets

We first show the results of six grayscale datasets in Table 1 and Fig. 5. Table 1 summarizes the comparisons with previous methods on PSNR, SSIM, and running time. RevSCI-net$^{8}_{50}$ in Table 1 indicates that the number of rev-blocks and groups in RevSCI-net are 50 and 8, respectively. It can be observed that the proposed RevSCI-net outperforms others, specifically 0.61dB in PSNR higher than previous state-of-the-art method BIRNAT, and using a similar testing time. Fig. 5 plots the selected reconstruction frames of different methods compared with the ground truth. RevSCI-net provides cleaner and sharper reconstructions than other algorithms; the fine details are recovered accurately.

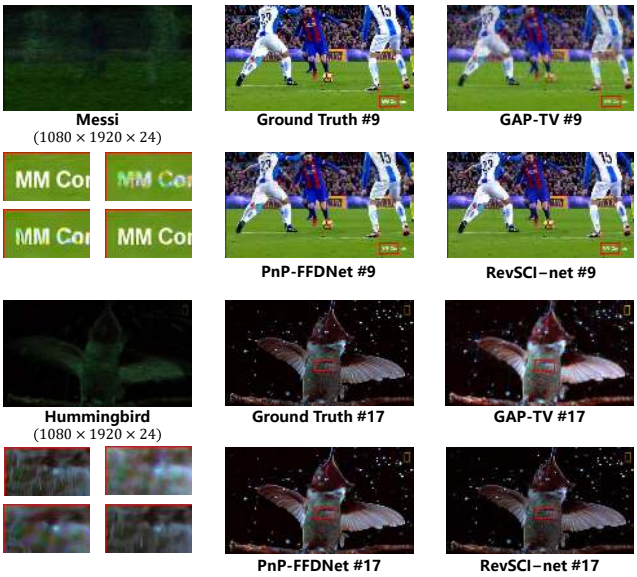

Figure 6. The reconstruction frames of RGB large-scale dataset `Messi` and `Hummingbird`. 24 RGB frames of size 1080 × 1920 × 3 are reconstructed from a single Bayer measurement of size 1080 × 1920.

Next, we show the results of RGB large-scale simulation dataset `Messi` and `Hummingbird` (1080×1920×3×24, where B=24) in Fig 6 and Table 2. It worth noting that the proposed RevSCI-net is the first end-to-end training network (joint reconstruction and demosaicing) to recover such a large SCI scene, and the reconstruction quality of RevSCI-net outperforms others. DeSCI will consume days to reconstruct, and therefore we only compare with GAP-TV and PnP-FFDNet. More analysis of memory and time is shown in Table 3. RevSCI-net occupies 10 times lower memory during training than the previous SOTA network BIRNAT.

Table 2. The average results of PSNR in dB (left entry), SSIM (right entry) by different algorithms on two color benchmark datasets.

Algorithm	Messi	Hummingbird
GAP-TV	18.56, 0.7209	18.29, 0.6449
PnP-FFDNet	21.54, 0.7959	24.13, 0.8340
RevSCI-net	24.35, 0.8576	31.97, 0.8816

Table 3. Training memory occupation (MB) and running time (seconds) in videos of different resolution and compression ratio. We only show the GPU memory occupation during training on BIRNAT and RevSCI-net with a single sample. '-' means not available due to too long time or too big memory consumption.

Method		256×256×8	256×256×14	512×512×50	1920×1080×24
GAP-TV	Time	4.2	11.6	180	524
DeSCI	Time	6180	3185.8	12600	-
PnP-FFDNet	Time	3.0	2.7	88	253
BIRNAT	Memory	17748	23912	>48000	>48000
	Time	0.16	0.28	-	-
RevSCI-net$^{8}_{50}$	Memory	1350	1876	11648*	46215*
	Time	0.19	0.33	3.56	12.46

* We used NVIDIA RTX8000 GPU with 48GB memory to train the model for the large-scale data.

5.3. Ablation Study

To quantitatively verify the contributions of the RevSCI-net, we modify the number of rev-blocks and groups in RevSCI-net with results shown in Table 4. The models are tested on the six grayscale datasets with results in Table 1. Note that stacking the rev-block will significantly increase the reconstruction quality, and adding the number of groups will help the reconstruction by more sufficiently

affine transformations in the feature-level. As mentioned before, adding rev-blocks will not increase the activation memory during training, while adding parameters will only increase a small amount of storage.

Table 4. Computational complexity and average reconstruction quality on six grayscale test sets using RevSCI-net with different reversible blocks and groups. MAC means Multiply Accumulate.

Model	Parameters ($\times 10^6$)	MACs ($\times 10^{11}$)	Memory (MB)	PSNR	SSIM
RevSCI-net$^2_{18}$	2.11	3.02	1283	33.11	0.947
RevSCI-net$^2_{28}$	3.22	4.47	1301	33.34	0.951
RevSCI-net$^2_{50}$	5.65	7.67	1350	33.62	0.954
RevSCI-net$^4_{50}$	5.65	7.67	1350	33.76	0.955
RevSCI-net$^5_{50}$	5.65	7.67	1350	33.84	0.956

Figure 7. The reconstruction frames of real data `Wheel` with size $256 \times 256 \times 14$.

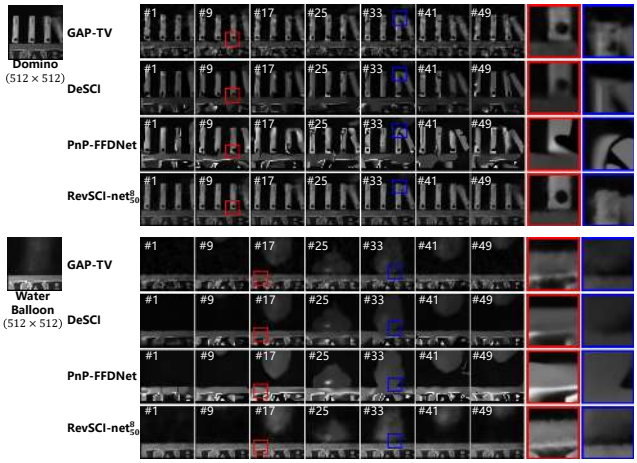

Figure 8. The reconstruction frames of real data `Domino` and `Water Balloon` with size $512 \times 512 \times 50$.

5.4. Results on Real Datasets

We now apply the proposed RevSCI-net on real data captured by two SCI cameras [25, 37]. The results of `Wheel` with a size of 256×256×14 are shown in Fig. 7. It can be observed that the results of RevSCI-net provide sharper edges and clearer letter 'D' than others. The results of `Domino` and `Water Balloon` with a size of 512×512×50 are shown in Fig. 8. In such a large compression ratio (50), the results of DeSCI are extremely over smooth, and GAP-TV introduces significant noise. Unpleasant artifacts exist in the results of PnP-FFDNet. The results of RevSCI-net have more accurate motions and contours. As mentioned before, our proposed RevSCI-net is the first end-to-end deep model that can handle such a large-scale problem, while existing deep model will fail due to high demands of GPU memory. Thanks to the reversible network, we can now apply RevSCI-net to large-scale SCI reconstruction problems in our daily life.

6. Conclusions

Efficient reconstruction algorithms for large scale problems have been a long-term challenge in inverse problems. Inspired by the recent advances of deep learning, fast inference is promising by training a deep network. However, for real life large-scale problems, deep networks are usually starving for memory and power. In this paper, based on the application of video snapshot compressive imaging, we propose a novel memory efficient network for large-scale reconstruction. Specifically, we introduce the reversible 3D CNN in SCI reconstruction, and build the memory-efficient RevSCI-net. For the first time, we have achieved end-to-end training network to recover FHD SCI measurements. In addition, we combine demosaicing and SCI reconstruction to directly obtain RGB videos from raw Bayer measurements and thus pave the way of real applications of SCI [26]. Extensive results demonstrated that RevSCI-net has significant improved reconstruction quality and running time. Besides video SCI, we believe RevSCI-net will work well in other computational imaging problems such as compressive spectral imaging [31, 32, 55].

Another way to apply CNN to large scale data is to train a small network but to adapt it to different modulation masks. One recent work has been done in [45] demonstrating the promise of this direction using meta learning. As mentioned in [51], the other line of work is using deep unfolding [30]. The work in [12] unfolds the Gaussian scale mixture model and is able to train a small-size but multi-stage network to be used in the large scale spectral SCI problem [31, 57].

Acknowledgement

B. Chen acknowledges the support of NSFC (61771361), the 111 Project (No. B18039), and the Program for Oversea Talent by Chinese Central Government.

References

[1] Yoann Altmann, Stephen McLaughlin, Miles J Padgett, Vivek K Goyal, Alfred O Hero, and Daniele Fac-

cio. Quantum-inspired computational imaging. *Science*, 361(6403):eaat2298, 2018. 1

[2] J.M. Bioucas-Dias and M.A.T. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, December 2007. 1

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018. 3

[4] Ziheng Cheng, Ruiying Lu, Zhengjue Wang, Hao Zhang, Bo Chen, Ziyi Meng, and Xin Yuan. BIRNAT: Bidirectional recurrent neural networks with adversarial training for video snapshot compressive imaging. In *European Conference on Computer Vision (ECCV)*, August 2020. 2, 4, 6

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2, 3

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 2

[7] 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. 3

[8] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pages 2214–2224, 2017. 3, 4, 5

[9] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, 2014. 1

[10] K. He, X. Zhang, S. Ren, and Sun J. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[11] Yasunobu Hitomi, Jinwei Gu, Mohit Gupta, Tomoo Mitsunaga, and Shree K Nayar. Video from a single coded exposure photograph using a learned over-complete dictionary. In *2011 International Conference on Computer Vision*, pages 287–294. IEEE, 2011. 1, 2

[12] T. Huang, W. Dong, X. Yuan, J. Wu, and G. Shi. Deep gaussian scale mixture prior for spectral compressive imaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 8

[13] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9–18, 2018. 2

[14] S. Jalali and X. Yuan. Snapshot compressed sensing: Performance bounds and algorithms. *IEEE Transactions on Information Theory*, 65(12):8005–8024, Dec 2019. 3

[15] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller. Memory-efficient learning for large-scale computational imaging. *IEEE Transactions on Computational Imaging*, 6:1403–1414, 2020. 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018. 2

[18] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 303–319, 2018. 3

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 4

[20] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed random measurements. In *CVPR*, 2016. 2

[21] Y. Li, M. Qi, R. Gulve, M. Wei, R. Genov, K. N. Kutulakos, and W. Heidrich. End-to-end video compressive sensing using anderson-accelerated unrolled networks. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020. 2

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 4

[23] Lin Liu, Xu Jia, Jianzhuang Liu, and Qi Tian. Joint demosaicing and denoising with self guidance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[24] Yang Liu, Xin Yuan, Jinli Suo, David Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2990–3006, Dec 2019. 1, 2, 3, 6

[25] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David J Brady. Coded aperture compressive temporal imaging. *Optics Express*, 21(9):10526–10545, 2013. 1, 2, 6, 8

[26] S. Lu, X. Yuan, and W. Shi. Edge compression: An integrated framework for compressive imaging processing on cavs. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 125–138, 2020. 8

[27] Jiawei Ma, Xiaoyang Liu, Zheng Shou, and Xin Yuan. Deep tensor ADMM-Net for snapshot compressive imaging. In *IEEE/CVF Conference on Computer Vision (ICCV)*, 2019. 2

[28] Xiao Ma, Xin Yuan, Chen Fu, and Gonzalo R. Arce. Led-based compressive spectral temporal imaging system. *Optics Express*, 2021. 2

[29] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 4

[30] Ziyi Meng, Shirin Jalali, and Xin Yuan. Gap-net for snapshot compressive imaging. *arXiv: 2012.08364*, December 2020. 2, 8

[31] Ziyi Meng, Jiawei Ma, and Xin Yuan. End-to-end low cost compressive spectral imaging with spatial-spectral self-attention. In *European Conference on Computer Vision (ECCV)*, August 2020. 8

[32] Ziyi Meng, Mu Qiao, Jiawei Ma, Zhenming Yu, Kun Xu, and Xin Yuan. Snapshot multispectral endomicroscopy. *Opt.*

Lett., 45(14):3897–3900, Jul 2020. 8

[33] Xin Miao, Xin Yuan, Yunchen Pu, and Vassilis Athitsos. λ-net: Reconstruct hyperspectral images from a snapshot measurement. In *IEEE/CVF Conference on Computer Vision (ICCV)*, 2019. 2

[34] Joseph N. Mait, Gary W. Euliss, and Ravindra A. Athale. Computational imaging. *Adv. Opt. Photon.*, 10(2):409–483, Jun 2018. 1

[35] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017. 6

[37] Mu Qiao, Xuan Liu, and Xin Yuan. Snapshot spatial–temporal compressive imaging. *Opt. Lett.*, 45(7):1659–1662, Apr 2020. 2, 6, 8

[38] Mu Qiao, Xuan Liu, and Xin Yuan. Snapshot temporal compressive microscopy using an iterative algorithm with untrained neural networks. *Opt. Lett.*, 2021. 2

[39] Mu Qiao, Ziyi Meng, Jiawei Ma, and Xin Yuan. Deep learning for video compressive sensing. *APL Photonics*, 5(3):030801, 2020. 2, 6

[40] Dikpal Reddy, Ashok Veeraraghavan, and Rama Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. In *CVPR 2011*, pages 329–336. IEEE, 2011. 1, 2

[41] Yangyang Sun, Xin Yuan, and Shuo Pang. Compressive high-speed stereo imaging. *Opt Express*, 25(15):18182–18190, 2017. 2

[42] Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *arXiv preprint arXiv:2006.11469*, 2020. 3

[43] Ashwin A Wagadarikar, Nikos P Pitsianis, Xiaobai Sun, and David J Brady. Video rate spectral imaging using a coded aperture snapshot spectral imager. *Optics Express*, 17(8):6368–6388, 2009. 1

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[45] Z. Wang, H. Zhang, Z. Cheng, B. Chen, and X. Yuan. Metasci: Scalable and adaptive reconstruction for video compressive sensing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 8

[46] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[47] Kai Xu and Fengbo Ren. CSVideoNet: A real-time end-to-end learning framework for high-frame-rate video compressive sensing. *arXiv: 1612.05203*, Dec 2016. 2

[48] Jianbo Yang, Xuejun Liao, Xin Yuan, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin. Compressive sensing by learning a Gaussian mixture model from measurements. *IEEE Transaction on Image Processing*, 24(1):106–119, January 2015. 2

[49] Jianbo Yang, Xin Yuan, Xuejun Liao, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin. Video compressive sensing using Gaussian mixture models. *IEEE Transaction on Image Processing*, 23(11):4863–4878, November 2014. 2

[50] Xin Yuan. Generalized alternating projection based total variation minimization for compressive sensing. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2539–2543, Sept 2016. 1, 2, 6

[51] X. Yuan, D. J. Brady, and A. K. Katsaggelos. Snapshot compressive imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 38(2):65–88, 2021. 1, 8

[52] Xin Yuan, Yang Liu, Jinli Suo, and Qionghai Dai. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 6

[53] Xin Yuan, Patrick Llull, Xuejun Liao, Jianbo Yang, David J. Brady, Guillermo Sapiro, and Lawrence Carin. Low-cost compressive sensing for color video and depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3325, 2014. 2, 3

[54] Xin Yuan and Yunchen Pu. Parallel lensless compressive imaging via deep convolutional neural networks. *Optics Express*, 26(2):1962–1977, Jan 2018. 2

[55] Xin Yuan, Tsung-Han Tsai, Ruoyu Zhu, Patrick Llull, David Brady, and Lawrence Carin. Compressive hyperspectral imaging with side information. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):964–976, September 2015. 8

[56] Xin Yuan, Jinli Suo Yang Liu, Frédo Durand, and Qionghai Dai. Plug-and-play algorithms for video snapshot compressive imaging. *arXiv: 2101.04822*, Jan 2021. 2

[57] Siming Zheng, Yang Liu, Ziyi Meng, Mu Qiao, Zhishen Tong, Xiaoyu Yang, Shensheng Han, and Xin Yuan. Deep plug-and-play priors for spectral snapshot compressive imaging. *Photon. Res.*, 9(2):B18–B29, Feb 2021. 8