Dense Deep Unfolding Network with 3D-CNN Prior for Snapshot Compressive Imaging

Zhuoyuan Wu[†] Jian Zhang[†,‡] Chong Mou[†]
[†]Peking University Shenzhen Graduate School, Shenzhen, China
[‡]Peng Cheng Laboratory, Shenzhen, China
wuzhuoyuan@pku.edu.cn; zhangjian.sz@pku.edu.cn; eechongm@gmail.com

Abstract

Snapshot compressive imaging (SCI) aims to record three-dimensional signals via a two-dimensional camera. For the sake of building a fast and accurate SCI recovery algorithm, we incorporate the interpretability of model-based methods and the speed of learning-based ones and present a novel dense deep unfolding network (DUN) with 3D-CNN prior for SCI, where each phase is unrolled from an iteration of Half-Quadratic Splitting (HQS). To better exploit the spatial-temporal correlation among frames and address the problem of information loss between adjacent phases in existing DUNs, we propose to adopt the 3D-CNN prior in our proximal mapping module and develop a novel dense feature map (DFM) strategy, respectively. Besides, in order to promote network robustness, we further propose a dense feature map adaption (DFMA) module to allow inter-phase information to fuse adaptively. All the parameters are learned in an end-to-end fashion. Extensive experiments on simulation data and real data verify the superiority of our method. The source code is available at https://github.com/jianzhangcs/SCI3D.

1. Introduction

As a major branch of compressive sensing (CS) [7, 45, 47], snapshot compressive imaging (SCI) develops for the aim of capturing high dimensional signals such as video [21] or spectral data [33]. Incorporating additional hardware components in the imaging system, SCI samples consecutive video frames or spectral channels and integrates sampled data to get compressed measurements. Then original sampled signals will be reconstructed from such measurements through various algorithms. Building encoder and decoder in a hardware plus software fashion, limitation on recording high dimensional signals in terms of bandwidth and memory have been overcome by a large margin. In this paper, we concentrate on video SCI.

Taking measurements comprising information on sampled signals and the sensing matrix as input, SCI recovery algorithms focus on how to solve an ill-posed inverse problem mathematically. Conventional methods employ prior knowledge of image or video as regularizers and solve such a sparsity-regularized optimization problem iteratively [41, 38, 24, 29]. Though superior in convergence and theoretical analysis, these algorithms tend to have complex computation, and it is quite difficult to select appropriate priors, which restricts the practical application of these methods. Thanks to the vigorous development of deep learning, several SCI reconstruction methods based on convolutional neural networks (CNN) have been proposed in recent years, which prefer to learn a direct mapping from measurements to sampled frames [5, 23, 9, 25, 11, 39, 42]. Compared to traditional ones, learning-based methods can achieve more decent results while lacking interpretability.

Combining the merits of both model-based and learning-based methods, we design a dense deep unfolding network with 3D-CNN prior by unrolling Half-Quadratic Splitting [8] into a deep network for optimizing the SCI recovery problem. As depicted in Fig. 1, each iteration of HQS is mapped to a phase of our proposed network. Unlike previous learning-based methods that construct 2D-CNN to model images, we design an effective 3D-CNN that is more suitable to exploit spatial-temporal correlation between frames. Besides, to decrease the information loss between neighboring phases in deep unrolling networks (DUN) caused by up-sampling and down-sampling in channel dimension, we propose to fuse corresponding dense feature map of adjacent phases. Moreover, to make information selectively transmitted through phases, we propose a dense feature map adaption module. Our contribution can be summarized as below:

- We build a novel dense deep unfolding network with 3D-CNN prior for SCI recovery. Our method achieves

This work was supported in part by National Natural Science Foundation of China (61902009). (*Corresponding author: Jian Zhang.*)

a significant improvement both visually and qualitatively compared with current state-of-the-art methods.

- We first incorporate 3D-CNN as priors into the proximal mapping in deep unfolding network, which better characterizes the spatial-temporal correlations among frames.
- We propose dense feature map (DFM) fusion to break the limitation on the transmission of network information, which brings another increment of 0.45dB.
- We present a dense feature map adaption (DFMA) module to adaptively fuse DFM, which helps to transmit useful information between phases.

2. Related Work

2.1. Snapshot compressive imaging

Model-based methods serve structure sparsity of sampled video as a regularizer. For example, total variance [41], sparsity [24, 46], gaussian mixture model [38], optical flow [29] and non-local low rank [20] is employed as prior knowledge. Although model-based methods can be directly applied to different sensing matrices without retraining, the main drawback of such algorithms is that the performance is highly restricted by the selection of prior. Moreover, iteratively solving optimization problems can be time-consuming.

In recent years, researches based on deep learning have surged in computational imaging [43, 44, 40]. Some concentrate on constructing neural networks to learn a mapping from compressed measurements to sampled frames. For example, deep fully-connected network [11] is proposed to learn the linear mapping, a method of jointly optimizing the exposure patterns with the reconstruction framework [39] considering the constraints enforced by hardware. BIRNAT [5] is designed where a CNN reconstructs the first frame, and the other frames are reconstructed by a bidirectional dense neural network. Most Recently, MetaSCI [36] is composed of a shared backbone that has light-weight meta-modulation parameters for different masks. Others engage in building up deep unrolling networks, which map iterations of optimization algorithms to phases of DUN. Various methods like Tensor-ADMM [23], Tensor-FISTA [9], GAP-Net [25], PnP-FFDNet [42], Anderson-Accelerated unrolled network [18] and STEP-SCI [37] have been proposed for SCI. Nonetheless, previous DUN-based methods have three potential drawbacks: i) they tend to rely on 2D-CNNs to excavate the spatial-temporal correlation; ii) majority methods convey information that contains video frames from one to another phase, lacking more potential information; iii) when it comes to fusing inter-phase information, there is not an adaptive strategy using only useful information. Bearing the above concerns in mind, in this paper, we put forward a new design of unrolling network and show how to make use of spatial-temporal correlation and decrease information loss effectively.

2.2. 3D Convolutional Neural Networks

3D-CNN is good at capturing spatial-temporal correlation, especially for tackling video processing tasks. Over the years considerable applications of 3D-CNN are proposed in such areas as action recognition [14, 32], event classification [31], video super-resolution [16, 22], medical image segmentation [6], video inpainting [34, 3] and so on. However, only a few pieces of research on SCI recovery problems refer to 3D-CNN. Until lately, consisting of multi-group reversible 3D convolutional neural networks, RevSCI [4] is presented for large-scale video SCI, where several 3D convolutions are incorporated into their proposed networks of feature extraction, reversible mapping, and reconstruction. Nevertheless, shallow architecture restricts its performance due to limited receptive field. In this paper, we design an elaborate network for video SCI recovery to address such issues, which exhibits a more effective ability to utilize the spatial-temporal correlation.

3. Mathematical Model of SCI

Fig. 1 gives a brief depiction about the pipeline of video SCI system. Consider there is a measurement $\mathbf{Y} \in \mathbb{R}^{1\times H\times W}$ with the dimension of $H \times W$ which is compressed from a B-frame video sequence $\{\mathbf{X}_i\}_{i=1}^{B} \in \mathbb{R}^{1\times H\times W}$ through B masks $\{\mathbf{\Phi}_i\}_{i=1}^{B} \in \mathbb{R}^{1\times H\times W}$, such procedure can be expressed as bellow:

$$\mathbf{Y} = \sum_{i=1}^{B} \mathbf{\Phi}_i \otimes \mathbf{X}_i + \mathbf{N}, \tag{1}$$

where $\mathbf{N} \in \mathbb{R}^{1\times H\times W}$ denotes the noise and $\otimes$ is the Hadamard (element-wise) product. Eq. (1) is equivalent to the following linear form:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}, \tag{2}$$

where $\mathbf{y} = \text{vec}(\mathbf{Y}) \in \mathbb{R}^{HW}$ is the vectorized measurement, $\mathbf{n} = \text{vec}(\mathbf{N}) \in \mathbb{R}^{HW}$ is the vectorized noise and $\mathbf{x} = \text{vec}(\mathbf{X}) = [\text{vec}(\mathbf{X}_1)^\top, ..., \text{vec}(\mathbf{X}_B)^\top]^\top \in \mathbb{R}^{HWB}$ are the vectorized frames. The sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{HW\times HWB}$ can be expressed as a block diagonal form as below:

$$\mathbf{\Phi} = [\text{diag}(\text{vec}(\mathbf{\Phi}_1)), ..., \text{diag}(\text{vec}(\mathbf{\Phi}_B))]. \tag{3}$$

The compression rate is equal to $1/B$. The reconstruction error of SCI is bounded even when $B > 1$, which is proved in [13]. The purpose of SCI recovery problem is to reconstruct video frames $\mathbf{x}$ given the measurement $\mathbf{y}$ and masks $\mathbf{\Phi}$, where $\mathbf{\Phi}$ is often to be binary. Thus, $\mathbf{\Phi}\mathbf{\Phi}^\top$ is assumed invertible, commonly in a diagonal form with a straightforward calculation of inversion.

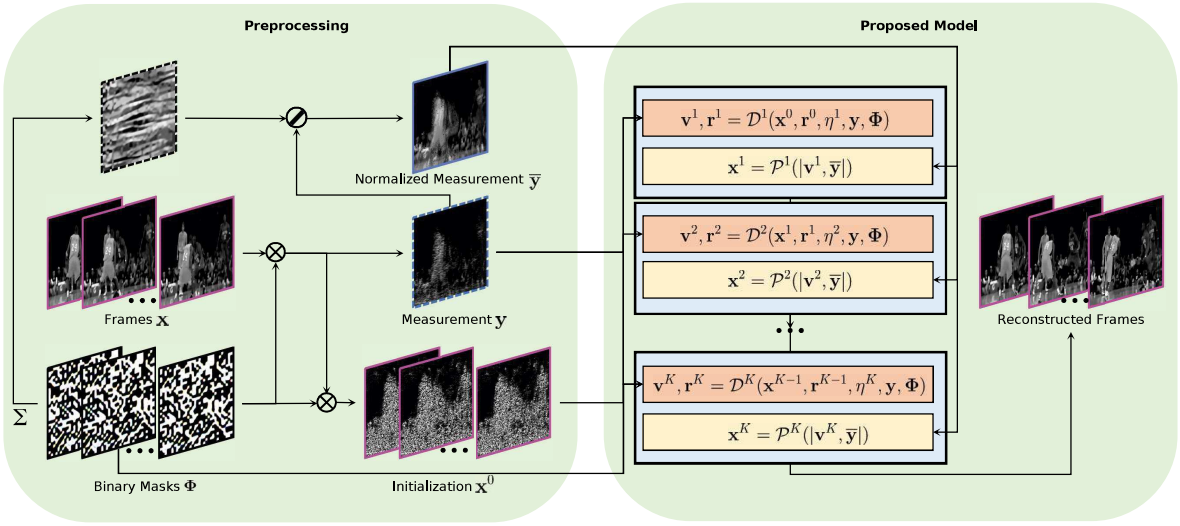

Figure 1: Illustration of the video SCI (left) and our proposed method (right). **Left**: a sequence of video frames $\mathbf{x}$ are passed through dynamic masks $\mathbf{\Phi}$ and then compressed to the measurement $\mathbf{y}$. Normalized measurement $\overline{\mathbf{y}}$ is acquired by element-divide the sum of dynamic masks. $\mathbf{x}^0$ is a coarse reconstruction of multiple frames. **Right**: details of our proposed network comprising $K = 10$ phases where each iteration of HQS update step is unrolled to a phase including data module $\mathcal{D}^k$ and prior module $\mathcal{P}^k$.

4. Proposed Method

4.1. Measurement Energy Normalization

As proposed in [5], a strategy of measurement energy normalization is designed to introduce more motion information which assists in reconstructing. The normalized measurement $\overline{\mathbf{Y}}$ is defined by

$$\overline{\mathbf{Y}} = \mathbf{Y} \oslash \overline{\mathbf{\Phi}}, \tag{4}$$

where $\oslash$ is the Hadamard (element-wise) division and $\overline{\mathbf{\Phi}} = \sum_{i=1}^{B} \mathbf{\Phi}_i$ is the sum of B sensing matrices which denotes how many pixels in $\{\mathbf{X}_i\}_{i=1}^{B}$ are integrated into the measurement $\mathbf{Y}$. Equivalently, $\overline{\mathbf{y}} = \text{vec}(\overline{\mathbf{Y}}) \in \mathbb{R}^{HW}$ is the vectorized normalized measurement. Generally speaking, the range of pixel values in $\mathbf{Y}$ can be broad, making it less effective to concatenate a non-energy-normalized measurement into the network directly. As depicted in Fig. 1, visually more pleasant than $\mathbf{Y}$, $\overline{\mathbf{Y}}$ can be served as a coarse reconstruction result which preserves a certain amount of motion information.

4.2. Optimization-based Unrolled Network

The SCI recovery problem can be solved by the following optimization problem:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi}\mathbf{x}||_2^2 + \lambda\Psi(\mathbf{x}), \tag{5}$$

where $\Psi(\mathbf{x})$ denotes the prior regularization with λ being the regularization parameter.

Following the framework of HQS, by introducing an auxiliary parameter $\mathbf{v}$, Eq. (5) can be converted to a constraint optimization problem:

$$(\mathbf{x}, \mathbf{v}) = \arg\min_{\mathbf{x},\mathbf{v}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi}\mathbf{v}||_2^2 + \lambda\Psi(\mathbf{x}),\ s.t.\ \mathbf{x} = \mathbf{v}. \tag{6}$$

The first term can be served as the data term, and the second term is the prior term. In order to obtain an unrolling inference, Eq. (6) can be divided into the following sub-problems and solved iteratively:

$$\mathbf{v}^k = \arg\min_{\mathbf{v}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi}\mathbf{v}||_2^2 + \frac{\eta}{2}||\mathbf{v} - \mathbf{x}^{k-1}||_2^2, \tag{7}$$

$$\mathbf{x}^k = \arg\min_{\mathbf{x}} \frac{\eta}{2}||\mathbf{v}^k - \mathbf{x}||_2^2 + \lambda\Psi(\mathbf{x}). \tag{8}$$

Here, k denotes the HQS iteration index, and η is another regularizer.

It can be observed that data term and prior term in Eq. (6) are decoupled to sub-problems Eq. (7) and Eq. (8). Intuitively, the aim of the data term and the prior term is to reconstruct frames $\mathbf{x}^k$ clearer and cleaner. Conventional optimization-based methods resolve the inverse problem by iteratively solving sub-problems according to optimization algorithms such as ADMM [2], GAP [19], ISTA [1] and so on. However, it is prolonged for hundreds of iterations to obtain a decent reconstruction result. The basic idea of our proposed method is to unroll Eq. (7) and Eq. (8) into a deep network with a fixed number of phases, where each iteration

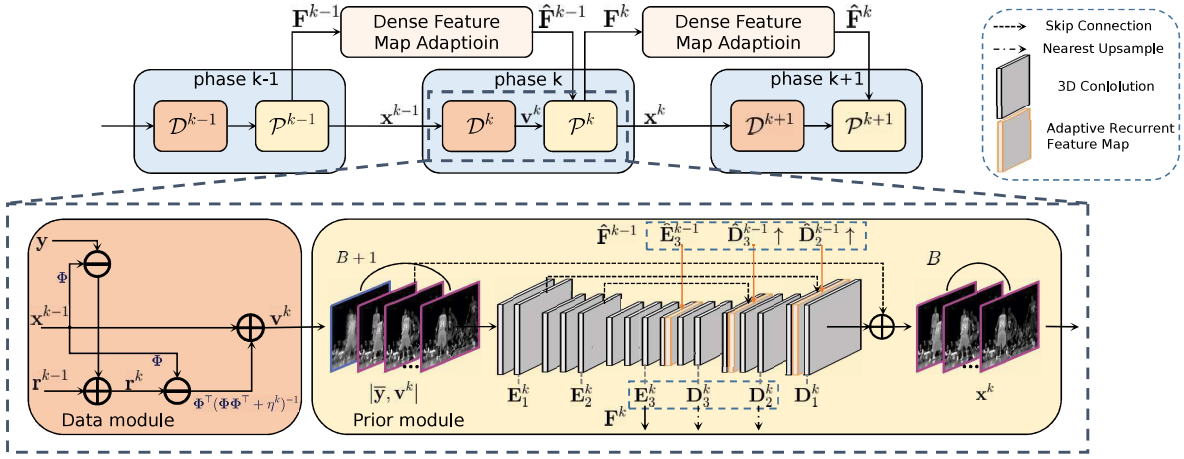

Figure 2: Illustration of each phase in our proposed network. Data module solves Eqs. (10)(11) and prior module solves Eq. (13). $\mathbf{F}^{k-1}$ represents dense feature map from the $(k-1)$-th phase and $\hat{\mathbf{F}}^{k-1}$ denotes its adaptive counterpart. The module of dense feature map adaption makes information selectively transmitted through phases. $\mathbf{E}_3^k$, $\mathbf{D}_3^k\uparrow$, and $\mathbf{D}_2^k\uparrow$ constitute the dense feature map $\mathbf{F}^k$.

corresponds to one phase. As shown in Fig. 2, k-th iteration of HQS is cast to k-th phase comprising data module $\mathcal{D}$ and prior module $\mathcal{P}$. Details will be discussed in the following:

- Data module $\mathcal{D}$: It corresponds to Eq. (7) and is used to generate the immediate reconstructed result $\mathbf{v}^k$. Given $\mathbf{x}^{k-1}$, the update of $\mathbf{v}$ in Eq. (7) can be regarded as a euclidean projection of $\mathbf{v}$ on the linear manifold:

$$\mathbf{v}^k = \mathbf{x}^{k-1} + \mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top + \eta^k)^{-1}(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{k-1}). \quad (9)$$

  As shown in [19], the linear manifold can be adjusted adaptively, so Eq. (9) can be modified as:

$$\mathbf{r}^k = \mathbf{r}^{k-1} + (\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{k-1}),\ \forall k \geq 1, \quad (10)$$

$$\mathbf{v}^k = \mathbf{x}^{k-1} + \mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top + \eta^k)^{-1}(\mathbf{r}^k - \mathbf{\Phi}\mathbf{x}^{k-1}). \quad (11)$$

  To increase the flexibility of network, η^k for $k \in \{1, 2, ..., K\}$ is set as a separate learnable variable in each phase with initialization of 0.01. For simplification, the process of the data module can be described as:

$$\mathbf{v}^k = \mathcal{D}^k(\mathbf{x}^{k-1}, \mathbf{r}^{k-1}, \eta^k, \mathbf{y}, \mathbf{\Phi}). \quad (12)$$

- Prior module $\mathcal{P}$: The prior module aims to make $\mathbf{v}^k$ closer to the desired signal domain. It can be defined as:

$$\mathbf{x}^k = \mathcal{P}^k(|\mathbf{v}^k, \overline{\mathbf{y}}|), \quad (13)$$

  where $|\cdot|$ denotes the concatenation. For the prior module $\mathcal{P}^k$ in each phase, the input is the concatenate of $\mathbf{v}^k$ and $\overline{\mathbf{y}}$. As explained in Sec. 4.1, the normalized measurement can provide visually pleasant motion information to guide reconstruction proceedings. Eq. (8) is a special case of proximal mapping $\text{prox}_{\Psi,\lambda/\eta}$ associated with the non-linear transform $\Psi(\cdot)$. Traditionally the sparse transformation $\Psi(\cdot)$ is set by handcraft. Inspired by [42], we design a deep network to solve the problem of proximal mapping.

Our network shares the similar architecture of U-Net as designed in [30]. However, we apply 3D convolution layers instead of the 2D ones. Fig. 3 gives an illustration of the differences between 2D and 3D convolution. By contrast with filters in 2D convolution, 3D ones sweep in additional dimension (*i.e.*, temporal duration), making it more suitable to exploit spatial-temporal correlation, an intuitional idea that is proved by our later experiments.

As portrayed in Fig. 2, the architecture of the prior module $\mathcal{P}^k$ in the k-th phase is a modified U-Net consisting of an encoder and a decoder. Considering there are three scales in both encoder along with decoder and architecture in each scale is equivalent to a residual block, we use $j \in [1, 2, 3]$ to denote the scale where $j = 1$ represents the largest scale, thereby the output of residual block in separate scale can be denoted as $\mathbf{E}_j^k$ for the encoder and $\mathbf{D}_j^k$ for the decoder. The procedure of the whole network can be expressed as

$$\begin{aligned} \mathbf{E}_j^k &= \text{enc}_j(\mathbf{E}_{j-1}^k), \\ \mathbf{D}_j^k &= \text{dec}_j(\overline{\mathbf{D}}_j), \end{aligned} \quad (14)$$

where $\overline{\mathbf{D}}_j$ is the input for decoder at level j, details will be discussed in Sec. 4.3. It should be noticed that $\mathbf{E}_0^k = |\mathbf{v}^k, \overline{\mathbf{y}}|$.

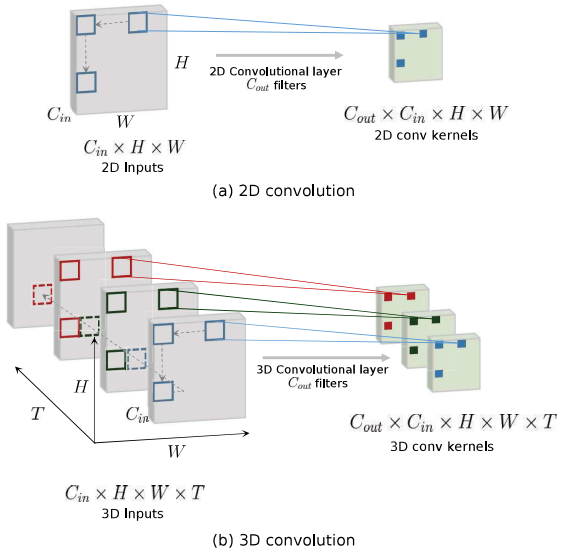

Figure 3: Illustration of comparison between 2D and 3D convolution. For 3D convolution, the input feature map has an additional dimension, which means filters of 3D convolution slide both along the spatial and temporal dimensions.

Finally, the prior module learns a residual map:

$$\mathbf{x}^k = \mathbf{D}_1^k + \mathbf{v}^k. \tag{15}$$

4.3. Dense Feature Map Fusion

First, we are going to discuss the bottleneck in existing DUNs. As shown in Fig. 1, the input of reconstruction network $\mathbf{x}^0$ is assigned with $\mathbf{\Phi}^\top \mathbf{y}$. Then, each phase contains data module $\mathcal{D}$ and prior module $\mathcal{P}$ cast from Eqs. (10)(11) and Eq. (13).

In the data module, for some intermediate result $\mathbf{u}$ the notation of $\mathbf{\Phi u}$ represents the operation of image-wise sampling while $\mathbf{\Phi}^\top \mathbf{u}$ denotes the operation of image-wise zero-filling. Given $\mathbf{x}^{k-1} \in \mathbb{R}^{B\times H\times W}$, after corresponding image-wise sample & zero-fill operation according to Eqs. (10)(11), we get $\mathbf{v}^k$, then in the prior module, $\mathbf{x}^k$ is often obtained by solving a problem of the proximal mapping via a designed denoising network.

It is obvious that the input and output of each phase contain limited channels, that is, each frame of $\mathbf{v}^k$ and $\mathbf{x}^k$ is simply containing minor channels (*e.g.*, 1 for gray images and 3 for RGB images). There exist information loss during transmissions between phases, which greatly hinders the performance of the entire network. To break the bottleneck, we come up with a strategy of dense feature map (DFM) fusion. Our basic idea can be exhibited as follows

$$\begin{aligned} \mathbf{D}_3^k &= \text{dec}_3(|\mathbf{E}_3^k, \mathbf{E}_3^{k-1}|), \\ \mathbf{D}_2^k &= \text{dec}_2(|\mathbf{D}_3^k \uparrow + \mathbf{E}_2^k, \mathbf{D}_3^{k-1} \uparrow|), \\ \mathbf{D}_1^k &= \text{dec}_1(|\mathbf{D}_2^k \uparrow + \mathbf{E}_1^k, \mathbf{D}_2^{k-1} \uparrow|) \end{aligned} \tag{16}$$

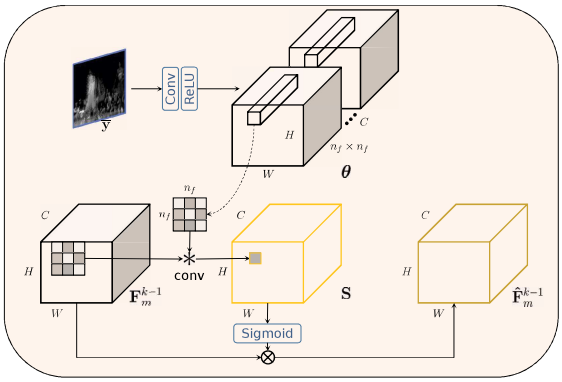

Figure 4: Visualization of our dense feature map adaption (DFMA) module. Similarity matrix $\mathbf{S}$ is derived by convolving a plain m-th DFM $\mathbf{F}_m^{k-1}$ where kernels θ is generated through a convolution from normalized measurement $\overline{\mathbf{y}}$. Adaptive DFM is obtained by multiplying $\mathbf{S}$ and $\mathbf{F}_m^{k-1}$.

where $\uparrow$ denotes the nearest up-sample and feature maps from $(k-1)$-th phase are named as dense feature map

$$\mathbf{F}^{k-1} = [\mathbf{E}_3^{k-1}, \mathbf{D}_3^{k-1} \uparrow, \mathbf{D}_2^{k-1} \uparrow]. \tag{17}$$

We fuse dense feature map from the beginning of the decoder at each scale because, particularly for such architecture as encoder-decoder, information passed through the encoder has been compressed and necessary to provide extra information for the decoder.

4.4. Dense Feature Map Adaption Module

As described in Sec. 4.3, notwithstanding that our proposed DFM fusion has decrease information loss to a great extent, we realize that applying the same integration manner to all DFMs is not an optimal way. Intuitively, different channels of DFM are able to contribute differently to final fusion. Inspired by [12], we design a dense feature map adaption (DFMA) module to adaptively fuse inter-phase information.

Our assumption is that each unit in DFM $\mathbf{F}^{k-1}$ which exhibits a similar appearance with the normalized measurement $\overline{\mathbf{y}}$ should be highlighted while others are supposed to be suppressed. In this way, only useful information in DFM can be transmitted between phases. As depicted in Fig. 4, our basic idea is calculating a similarity map $\mathbf{S}$. As mentioned in [15], one can generate spatially variant filters and samples on each location to compute the similarity between DFM and normalized measurement. More concretely, spatially variant filters $\boldsymbol{\theta} \in \mathcal{R}^{H\times W\times C\times n_f\times n_f}$ are generated from normalized measurement $\overline{\mathbf{y}}$, where H, W and C are the height, length and channels of corresponding scale j. n_f is the kernel size, which is being set as 3 in our experiment. The similarity $\mathbf{S}(p,q,c)$ between $\mathbf{F}_m^{k-1}$ and $\overline{\mathbf{y}}$ at location

Table 1: Comparison with other algorithms. The average results of PSNR in dB (left entry in each cell) and SSIM (right entry in each cell) and running time per measurement in seconds. Best results are in red and the second-best ones are in blue.

Dataset	Kobe	Traffic	Runner	Drop	Aerial	Crash	Average	Running time
GAP-TV [41]	26.45 0.845	20.90 0.715	28.48 0.899	33.81 0.963	25.03 0.828	24.82 0.838	26.58 0.848	4.2
E2E-CNN [28]	27.79 0.807	24.62 0.840	34.12 0.947	36.56 0.949	27.18 0.869	26.43 0.882	29.45 0.882	0.0312
DeSCI [20]	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 [42]	30.47 0.926	24.08 0.833	32.88 0.938	40.87 0.988	24.02 0.814	24.32 0.836	29.44 0.889	3.0
BIRNAT [5]	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
Tensor-ADMM [23]	30.50 0.890	NA	NA	NA	25.42 0.780	25.27 0.860	NA	2.1
Tensor-FISTA [9]	31.41 0.920	NA	NA	NA	26.46 0.890	27.46 0.880	NA	1.7
GAP-Unet-S12 [25]	32.09 0.944	28.19 0.929	38.12 0.975	42.02 0.992	28.88 0.914	27.83 0.931	32.86 0.947	0.0072
MetaSCI [36]	30.12 0.907	26.95 0.888	37.02 0.967	40.61 0.985	28.31 0.904	27.33 0.906	31.72 0.926	0.025
RevSCI [4]	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
Ours	35.00 0.969	31.76 0.966	40.03 0.980	44.96 0.995	30.46 0.943	29.33 0.956	35.26 0.968	1.35

Algorithm 1 Dense Deep Unfolding Network with 3D-CNN Prior for SCI

Input: Sensing matrix $\mathbf{\Phi}$, observed measurement $\mathbf{y}$
Output: Reconstructed frames $\mathbf{x}^K$

1: Initialize $\mathbf{x}^0 = \mathbf{\Phi}^\top \mathbf{y}$, $\mathbf{r}^0 = \mathbf{0}$
2: **for** $k = 1, ..., K$ **do**
3: update $\mathbf{r}^k$ by Eq. (10)
4: update $\mathbf{v}^k$ by Eq. (11)
5: update $\mathbf{x}^k$ by Eq. (15)
6: **end for**

(p, q) in c-th channel can be computed by

$$\mathbf{S}(p,q,c) = \sum_{u=-z}^{z}\sum_{v=-z}^{z} \boldsymbol{\theta}(p,q,c,u,v) \times \mathbf{F}_m^{k-1}(p+u, q+v, c), \tag{18}$$

where $z = \lfloor n_f/2 \rfloor$ and $\mathbf{F}_m^{k-1}$ signifies the m-th ($m = 1, 2, 3$) entry of DFM $\mathbf{F}^{k-1}$. Then adapted dense feature map $\hat{\mathbf{F}}_m^{k-1}$ is obtained by

$$\hat{\mathbf{F}}_m^{k-1} = \sigma(\mathbf{S}) \otimes \mathbf{F}_m^{k-1}, \tag{19}$$

where σ denotes the sigmoid function to convert its value into the range $[0, 1]$. Then Eq. (17) can be modified to

$$\hat{\mathbf{F}}^{k-1} = [\hat{\mathbf{E}}_3^{k-1}, \hat{\mathbf{D}}_3^{k-1} \uparrow, \hat{\mathbf{D}}_2^{k-1} \uparrow]. \tag{20}$$

4.5. Training

The learnable parameters are denoted as Θ, including η^k in the data module and parameters of the deep network in the prior module $\mathcal{P}^k$. Thus $\Theta = \{\eta^k, \mathcal{P}^k\}_{k=1}^K$, where K is total number of phases ($K = 10$ in our proposed network).

Given the training pairs $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^{N_k}$, our network takes the measurement $\mathbf{y}_i$ as input and generates reconstructed frames $\mathbf{x}_i^K$. The mean square error (MSE) is chosen as our loss function, expressed as

$$\mathcal{L} = \frac{1}{N_k N_s} \sum_{i=1}^{N_k} ||\mathbf{x}_i^K - \mathbf{x}_i||_2^2, \tag{21}$$

where N_k, N_s are the total number of training blocks and size of each block $\mathbf{x}_i$.

The architecture is implemented in PyTorch [26] with 4 NVIDIA Tesla V100 GPUs. To optimize the parameters, we adopt the Adam optimizer [17] with a mini-batch size of 4 and the number of total epoch is 200. The initial learning rate is set to be 1.28×10^{-4}, but we find it is too large to be converged. To overcome early optimization difficulties, we use the strategy of warmup [10] with the first five epochs, *i.e.*, using lower learning rates at the start of training. After the first five epochs, the learning rate decays with a factor of 0.9 every ten epochs.

5. Experiments

5.1. Datasets

We choose **DAVIS** [27] which has 90 scenes with 480p resolution as our training dataset. 25600 video clips are sampled with the block size of 128×128 and the temporal size of 8. Data is augmented with random cropping and rotation. The synthetic data with size of $256 \times 256 \times 8$ follows the same setup in [20], including **Kobe, Traffic, Runner, Drop, Crash,** and **Aerial** [23]. The real data includes **Water Balloon** and **Dominoes** with size of $512 \times 512 \times 10$ [28]. Both PSNR and SSIM [35] are selected as the evaluation metrics.

5.2. Results on Simulation Data

We compare our proposed method with existing algorithms including model-based methods (GAP-TV [41] and DeSCI [20]), learning-based methods (E2E-CNN [28], BIRNAT [5], MetaSCI [36] and RevSCI [4]) and DUN-based methods (PnP-FFDNet [42], Tensor-ADMM [23], Tensor-FISTA [9] and GAP-Unet-S12 [25]) on synthetic data. It should be noted that for Tensor-ADMM and Tensor-FISTA only 3 results out of 6 scenes are provided because they only trained models on 3 scenes. Quantitative comparison is shown in Table 1.

It is obvious that our proposed network has surpassed

Table 2: Ablation study on convolution: 3D-CNN versus 2D-CNN.

Sequence	2DCN	3DCN
Kobe	32.07 0.940	34.09 0.962
Traffic	28.26 0.931	30.72 0.958
Runner	36.43 0.967	38.81 0.976
Drop	42.38 0.992	44.20 0.994
Aerial	29.18 0.921	30.04 0.936
Crash	28.29 0.934	29.06 0.950
Average	32.77 0.948	34.49 0.963

all the existing algorithms on six benchmarks by a large margin. Especially our performance exceeds the existing best algorithm RevSCI by 1.34dB on average. Fig. 5 and Fig. 6 plots some selected reconstructed results on simulation data. Note that the model-based method GAP-TV has worse reconstruction results. Built on the plug-and-play framework, PnP-FFDNet obtains decent results but still lacks details, especially in texture-rich areas. The results of DeSCI are a little bit over smooth with the cost of high computation load. Compared with BIRNAT, in zooming areas, our method obtains sharper edges and more abundant details, demonstrating the effectiveness of utilization of 3D-CNN and the success of declining information loss.

5.3. Results on Real Data

We apply our network to real data **WaterBalloon** and **Dominoes** [28] where masks are controlled by a digital micromirror device. It is more challenging to reconstruct the real measurements because of noise. As illustrated in Fig. 7, our method can generate more apparent contours with fewer artifacts and ghosting. In general, our method achieves the best performance with competing inference speed. When facing a more challenging situation, our method's advantages are more pronounced, which indicates that our algorithm is more applicable under the actual scenario.

5.4. Ablation Study

5.4.1 Effect of 3D-CNN over 2D-CNN

To validate the effectiveness of the application of 3D-CNN, we trained a plain 3D modified UNet ('3DCN') [30] and corresponding 2D version ('2DCN') without integration of the DFM fusion and DFMA module. As shown in Table 2, 3DCN can achieve an increment of 1.72dB on PSNR, which shows not just sliding in the dimension of height and width, sliding in temporal dimension is necessary for extracting spatial-temporal correlation.

Table 3: Ablation study on various versions in terms of the average results of PSNR in dB.

3DCN	DFM_1	DFM_2	DFM_3	DFMA	PSNR
✓					34.49
✓	✓				34.54
✓	✓			✓	34.81
✓	✓	✓			34.80
✓	✓	✓		✓	35.03
✓	✓	✓	✓		34.94
✓	✓	✓	✓	✓	35.26

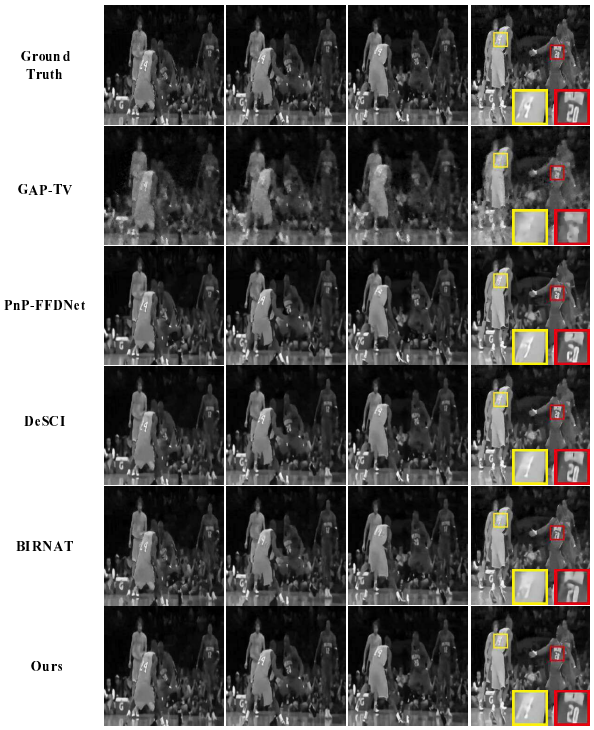

Figure 5: Selected multiple reconstruction frames of simulated data **Kobe**.

5.4.2 Effect of DFM and DFMA

To verify the proposed DFM fusion and DFMA, we integrate diverse branches based on 3DCN in Sec. 5.4.1 with three branches in $\mathbf{F}^k$ denoted as 'DFM_1', 'DFM_2' and 'DFM_3' respectively. As manifested in Table 3, combining the second branch improves the performance most (0.26dB) contrast with the first (0.05dB) and third branch (0.14dB). The whole branches bring an improvement of 0.45dB, proving that merging dense feature map from adjacent phases can reduce the information loss to a great extent. Incorporating DFMA into each branch brings additional enhancement, and the whole branch of DFMA leads to a boost of 0.32dB on PSNR.

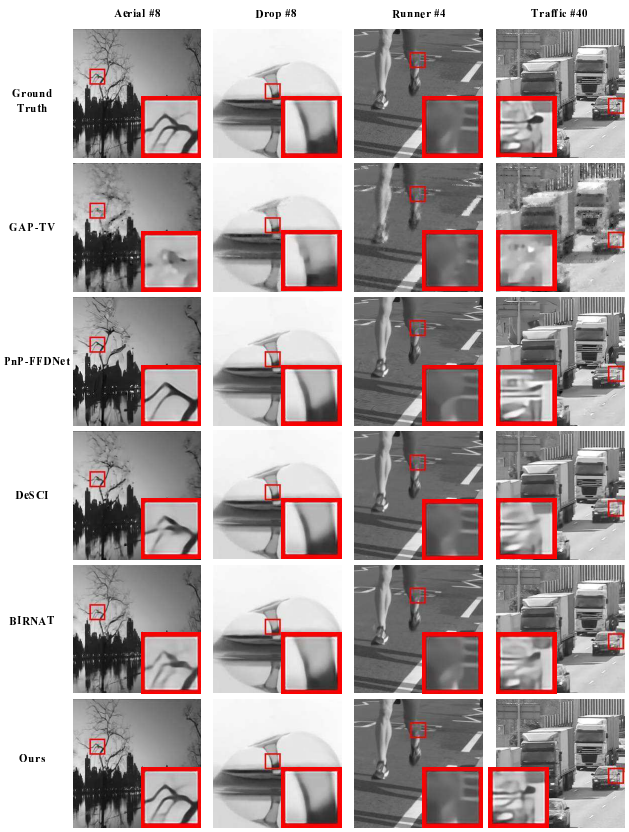

Figure 6: Selected reconstruction frames of simulated data **Aerial, Drop, Runner and Traffic**.

5.4.3 Effect of Phase Number K

To ensure the phase number K being set as an appropriate value, we retrain the network with K being designated to $2, 4, 6, 8$ and 10. Fig. 8 plots the performance with various phase numbers. Note that the performance with $K = 4$ is 0.26dB higher than RevSCI. Although reconstruction accuracy will increase with more phases, considering the memory cost and inference time, we assign K to be 10.

6. Conclusions and Future Work

Inspired by the half quadratic splitting (HQS) algorithm, we put forward a novel dense deep unfolding network with 3D-CNN prior for Snapshot compressive imaging. Merging the merits of both model-based methods and learning-based methods, our method has strong interpretability and high-quality reconstructed results. To enhance the ability to exploit spatial-temporal correlation, we assemble a deep network with 3D-CNN prior. To reduce the information loss, we propose a strategy of dense feature map (DFM) fusion, and we also design a dense feature map adaption (DFMA) module to make information optionally transmitting between phases.

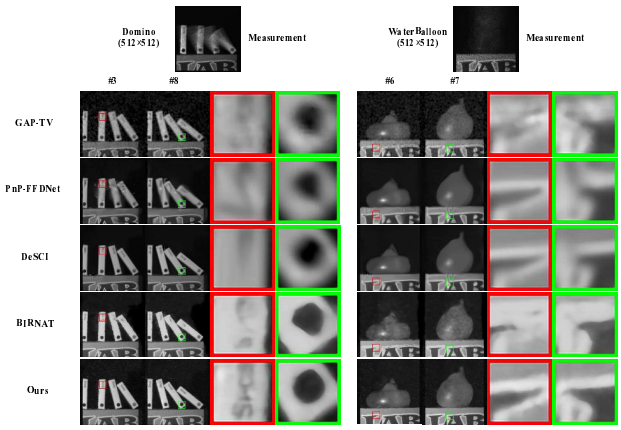

Figure 7: Selected reconstruction frames of real data.

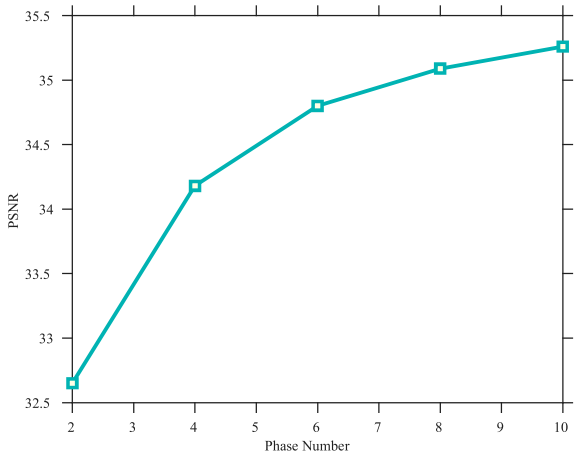

Figure 8: PSNR comparison on reconstruction accuracy with various number of phase.

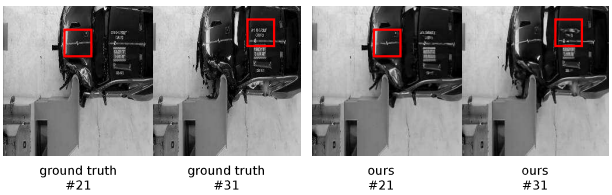

Figure 9: A case of unsatisfactory reconstruction results due to large motion.

Despite the fact that our method has obtained the best performance so far, when it comes to large motion within frames, our method can not attain decent results, as shown in Fig. 9. How to properly process large motion is thorny trouble for existing SCI recovery methods. In our future work, we might insert motion estimation techniques such as optical flow to solve this trouble. In addition, we will consider the application in other computational imaging problems, not just video SCI.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 3

[2] Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016. 3

[3] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9066–9075, 2019. 2

[4] Ziheng Cheng, Bo Chen, Guanliang Liu, Hao Zhang, Ruiying Lu, Zhengjue Wang, and Xin Yuan. Memory-efficient network for large-scale video compressive sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16246–16255, June 2021. 2, 6

[5] 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 *Proceedings of the IEEE conference on European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 6

[6] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Proceedings of the International conference on Medical Image Computing and Computer-assisted Intervention*, pages 424–432. Springer, 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. 1

[8] Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE transactions on Image Processing*, 4(7):932–946, 1995. 1

[9] Xiaochen Han, Bo Wu, Zheng Shou, Xiao-Yang Liu, Yimeng Zhang, and Linghe Kong. Tensor fista-net for real-time snapshot compressive imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10933–10940, 2020. 1, 2, 6

[10] 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. 6

[11] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9–18, 2018. 1, 2

[12] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *Proceedings of the European Conference on Computer Vision*, pages 645–660. Springer, 2020. 5

[13] Shirin Jalali and Xin Yuan. Snapshot compressed sensing: Performance bounds and algorithms. *IEEE Transactions on Information Theory*, 65(12):8005–8024, 2019. 2

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2012. 2

[15] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Proceedings of the Advances in Neural Information Processing Systems*, 29:667–675, 2016. 5

[16] Soo Ye Kim, Jeongyeon Lim, Taeyoung Na, and Munchurl Kim. 3dsrnet: Video super-resolution using 3d convolutional neural networks. *arXiv preprint arXiv:1812.09079*, 2018. 2

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

[18] Yuqi Li, Miao Qi, Rahul Gulve, Mian Wei, Roman Genov, Kiriakos N Kutulakos, and Wolfgang Heidrich. End-to-end video compressive sensing using anderson-accelerated unrolled networks. In *Proceedings of the 2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 2

[19] Xuejun Liao, Hui Li, and Lawrence Carin. Generalized alternating projection for weighted-2,1 minimization with applications to model-based compressive sensing. *SIAM Journal on Imaging Sciences*, 7(2):797–823, 2014. 3, 4

[20] Yang Liu, Xin Yuan, Jinli Suo, David J Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2990–3006, 2018. 2, 6

[21] 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

[22] Jianping Luo, Shaofei Huang, and Yuan Yuan. Video super-resolution using multi-scale pyramid 3d convolutional networks. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1882–1890, 2020. 2

[23] Jiawei Ma, Xiao-Yang Liu, Zheng Shou, and Xin Yuan. Deep tensor admm-net for snapshot compressive imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10223–10232, 2019. 1, 2, 6

[24] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, 2012. 1, 2

[25] Ziyi Meng, Shirin Jalali, and Xin Yuan. Gap-net for snapshot compressive imaging. *arXiv preprint arXiv:2012.08364*, 2020. 1, 2, 6

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Proceedings of the Advances in Neural Information Processing Systems*, 32:8026–8037, 2019. 6

[27] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 6

[28] Mu Qiao, Ziyi Meng, Jiawei Ma, and Xin Yuan. Deep learning for video compressive sensing. *APL Photonics*, 5(3):030801, 2020. 6, 7

[29] Dikpal Reddy, Ashok Veeraraghavan, and Rama Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 329–336. IEEE, 2011. 1, 2

[30] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1354–1363, 2020. 4, 7

[31] Iikka Tapio Teivas. Video event classification using 3d convolutional neural networks. Master's thesis, 2017. 2

[32] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 2

[33] 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

[34] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5232–5239, 2019. 2

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

[36] Zhengjue Wang, Hao Zhang, Ziheng Cheng, Bo Chen, and Xin Yuan. Metasci: Scalable and adaptive reconstruction for video compressive sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2083–2092, 2021. 2, 6

[37] Zhuoyuan Wu, Zhenyu Zhang, Jiechong Song, and Jian Zhang. Spatial-temporal synergic prior driven unfolding network for snapshot compressive imaging. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2021. 2

[38] Jianbo Yang, Xin Yuan, Xuejun Liao, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin. Video compressive sensing using gaussian mixture models. *IEEE Transactions on Image Processing*, 23(11):4863–4878, 2014. 1, 2

[39] Michitaka Yoshida, Akihiko Torii, Masatoshi Okutomi, Kenta Endo, Yukinobu Sugiyama, Rin-ichiro Taniguchi, and Hajime Nagahara. Joint optimization for compressive video sensing and reconstruction under hardware constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 634–649, 2018. 1, 2

[40] Di You, Jian Zhang, Jingfen Xie, Bin Chen, and Siwe Ma. COAST: COntrollable Arbitrary-Sampling NeTwork for Compressive Sensing. *IEEE Transactions on Image Processing*, 30:6066–6080, 2021. 2

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

[42] Xin Yuan, Yang Liu, Jinli Suo, and Qionghai Dai. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1447–1457, 2020. 1, 2, 4, 6

[43] 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. 2

[44] Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):765–774, 2020. 2

[45] Chen Zhao, Siwei Ma, and Wen Gao. Image compressive-sensing recovery using structured laplacian sparsity in dct domain and multi-hypothesis prediction. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2014. 1

[46] Chen Zhao, Siwei Ma, Jian Zhang, Ruiqin Xiong, and Wen Gao. Video compressive sensing reconstruction via reweighted residual sparsity. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6):1182–1195, 2016. 2

[47] Chen Zhao, Jian Zhang, Ronggang Wang, and Wen Gao. CREAM: CNN-REgularized admm framework for compressive-sensed image reconstruction. *IEEE Access*, 6:76838–76853, 2018. 1