Generalized Tensor Summation Compressive Sensing Network (GTSNET): An Easy to Learn Compressive Sensing Operation

Mehmet Yamaç, Ugur Akpinar, Erdem Sahin, *Member, IEEE*, Serkan Kiranyaz, *Senior Member, IEEE*, and Moncef Gabbouj, *Fellow, IEEE*

***Abstract*— The efforts in compressive sensing (CS) literature can be divided into two groups: finding a measurement matrix that preserves the compressed information at its maximum level, and finding a robust reconstruction algorithm. In the traditional CS setup, the measurement matrices are selected as random matrices, and optimization-based iterative solutions are used to recover the signals. Using random matrices when handling large or multi-dimensional signals is cumbersome especially when it comes to iterative optimizations. Recent deep learning-based solutions increase reconstruction accuracy while speeding up recovery, but jointly learning the whole measurement matrix remains challenging. For this reason, state-of-the-art deep learning CS solutions such as convolutional compressive sensing network (CSNET) use block-wise CS schemes to facilitate learning. In this work, we introduce a separable multi-linear learning of the CS matrix by representing the measurement signal as the summation of the arbitrary number of tensors. As compared to block-wise CS, tensorial learning eases blocking artifacts and improves performance, especially at low measurement rates (MRs), such as MRs < 0.1. The software implementation of the proposed network is publicly shared at https://github.com/mehmetyamac/GTSNET.**

***Index Terms*— Compressive sensing, deep reconstruction, tensorial compressive learning, separable compressive learning.**

I. Introduction

COMPRESSIVE sensing (CS) theory has attracted a lot of attention since its first appearance in 2005 [1]. CS theory claims that a signal can be sampled with far fewer measurements than the conventional Nyquist/Shannon-based sampling methods use. It has been applied in many fields such as CS-based MRI imaging [2], radar monitoring systems [3], [4], and ECG measurements in a health monitoring system [5]. Along with sampling, the technology has been adopted in many other fields. For instance, in a conventional CS system, random or pseudo-random measurement matrices are used, enabling a CS-based encryption mechanism [6], [7].

In Nyquist/Shannon based data acquisition systems, the reconstruction process is performed by sinc interpolation, which is a linear process and does not require expensive computations. The traditional CS-based data acquisition systems require advanced optimization-based iterative algorithms such as ℓ_1-minimization techniques [8], [9], [10]. Even if convex relaxation can bring a guarantee of sparse recovery with polynomial time, most solvers work in an iterative manner, and it makes them infeasible for real-time applications especially for large-scale signals, such as vectorized images. Moreover, ℓ_1 type estimators may lead to an unbiased estimation of the sparse signal [11]. There have been significant efforts spent to have faster recovery algorithms such as [12], [13], and [14] which are more feasible for a CS imaging system or similar multi-dimensional signals. However, the optimization-based recovery in a sparse domain can completely fail under some measurement rates, which are determined by the phase transition of the algorithms [15]. Moreover, the signal of interest in real applications rarely becomes strictly sparse in any sparsifying domain.

The first category of the deep learning-based CS approaches includes the works that use neural networks only for the reconstruction part [16], [17], [18]. They generally use conventional random matrices as the CS operators. To handle the images (2D signal), they apply the CS matrices to the vectorized smaller blocks of the image of interest. The well-known state-of-the art examples of this category of work can be listed as stacked denoising autoencoder (SDA) [16], non-iterative reconstruction of the compressively sensed images using CNN (ReconNet) [17], and learned version of iterative shrinkage thresholding algorithm for CS imaging (ISTA-Net) [18]. Among them, SDA uses the fully connected layers while the others adopt convolutional layers in their network. As a reconstruction part, ReconNet introduces fully convolutional layers, and this is why it is a non-iterative recovery framework that significantly reduces the computational time. ISTA-Net is based on iterative soft thresholding algorithms, and can be put into the category of deep unrolling techniques.

The second category of deep learning attempts can be enlisted as the ones that jointly learn CS matrices and reconstruction part instead of using conventional CS matrices. The recent state-of-the-art networks in this category are convolutional compressive sensing network (CSNET) [19] and scalable convolutional compressive sensing network (SCSNET) [20]. These works also handle CS of both

Manuscript received 2 October 2021; revised 6 October 2022; accepted 24 August 2023. Date of publication 29 September 2023; date of current version 17 October 2023. This work was supported in part by the EU H2020 OpenDR, in part by the Academy of Finland AWcHA, and in part by Business Finland AMALIA. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaolin Wu. *(Corresponding author: Mehmet Yamaç.)*

Mehmet Yamaç, Ugur Akpinar, Erdem Sahin, and Moncef Gabbouj are with the Faculty of Information Technology and Communication Sciences, Tampere University, 33720 Tampere, Finland (e-mail: mehmet.yamac@tuni.fi).

Serkan Kiranyaz is with the Department of Electrical Engineering, Qatar University, Doha, Qatar.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2023.3318946, provided by the authors.

Digital Object Identifier 10.1109/TIP.2023.3318946

gray-scale and RGB images in a block-by-block manner. However, they learn these CS matrices using convolutional kernels having the same size as the image blocks. In the reconstruction part they use convolutional layers to recover the full image as a whole. In this way, they can significantly improve the blocking artifacts.

In this study, we propose a novel network, Generalized Tensor Summation Networks (GTSNETs), that can *jointly* learn both CS matrix and reconstruction algorithm. With the proposed learned CS operator, conventional CS setup, matrix-vector multiplication, is approximated as T-number of measurement tensors' summation, where each measurement tensor is determined by the mode product of tensor and matrices (i.e. smaller sizes than full-sized CS matrices). As a result, GTSNETs can factorize and approximate unfactorized full-size CS matrices (i.e., matrices that can be applied to a vectorized image or multidimensional signal in the traditional CS setting) using Kronecker products [21] of smaller-size matrices. In contrast to previous attempts, a GTSNET is capable of replacing a wide range of compressive sensing systems, such as sub-Gaussian [22], separable [23], and structural CS matrices (e.g., sparsifying basis as a part of the CS matrix [6], [24], [25]). Some CS systems, however, in which hardware limitations impose special requirements over CS matrices (e.g., CS MRI [26], [27], CS Radar [3], [4] etc.), are outside the scope of this study.

In a GTSNET, CS operation can be performed directly over the spatial domain or in any other separable transformation basis like CS in the frequency domain using DCT. This is why GTSNET can generalize many CS systems, and thus we use the term Generalized Tensor Summation (GTS). When it comes to the performance comparison with a traditional deep learning approach, especially for lower measurement rates, the proposed system exhibits a superior performance in terms of PSNR and SSIM with a particular improvement over the fine details. At the same time, GTSNET performs signal reconstruction from compressively sensed measurements in a feed-forward manner and this significantly reduces the computational complexity compared to the iterative approaches.

The rest of the paper is organized as follows. In Section II, we shall make a brief introduction to compressive sensing, separable and multidimensional. Then, the proposed learnable compressive sensing operations will be presented in Section III. In Section IV extensive experimental results will be presented for the CS in both gray-scale and RGB images. We shall then present comparative evaluations in spatial and frequency domains. In addition, we shall investigate which information is more preserved when T is increased. Finally, the conclusions are drawn in Section VI.

II. Preliminaries and Prior Art

A. Compressive Sensing

CS [1], [28] theory has shown that a sparse signal can be recovered from far fewer measurements than traditional Shannon-Nyquist-based data acquisition methods use. Mathematically speaking, let a CS scheme linearly extracts m number of measurements of the signal, $\mathbf{s} \in \mathbb{R}^N$, i.e.,

$$\mathbf{y} = \Psi \mathbf{s}, \tag{1}$$

where the measurement matrix, $\Psi \in \mathbb{R}^{m \times N}$ represents the linear data acquisition with $m \ll N$. In the CS literature, the efforts of designing such a linear measurement system can be categorized into two groups: (i) Finding a measurement matrix, Ψ which maximally preserves the information of $\mathbf{s}$ while transforming it in a lower-dimensional subspace as in (1). (ii) Finding a robust reconstruction algorithm, which is able to recover $\mathbf{s}$ from $\mathbf{y}$ in a reasonable time with a tolerable reconstruction error.

From elementary linear algebra, one can easily say that (1) is an underdetermined linear system of equations where for a given Ψ and $\mathbf{y}$ pair, $\mathbf{s}$ has infinitely many solutions. Therefore, at least one more assumption is needed to have unique solution. For instance, if we know that the signal of interest, $\mathbf{s}$, is sparse in a proper sparsifying domain Φ, then (1) can be expressed as

$$\mathbf{y} = \Psi \Phi \mathbf{x} = \mathbf{A}\mathbf{x}, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^N$ is sparse or compressible coefficient vector (e.g., if it is k-sparse $\|\mathbf{x}\|_{\ell_0^N} \leq k$) and $\mathbf{A} = \Psi\Phi$, which can be named as equivalent dictionary [29]. Under the assumption that the coefficient vector is k-sparse, then the following sparse representation,

$$\min_{\mathbf{x}} \;\; \|\mathbf{x}\|_0 \;\; \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y} \tag{3}$$

is unique if $m \geq 2k$ and the minimum number of linearly dependent columns of $\mathbf{A}$ (see the definition of spark of a matrix [30]) is greater than $2k$ [30]. However, the problem in (3) is non-convex and known to be NP-hard. Fortunately, the most common approach will be the relaxation of it to an ℓ_1 minimization problem,

$$\arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \;\; s.t. \; \mathbf{x} \in \mho(\mathbf{y}) \tag{4}$$

where $\mho(\mathbf{y}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{y}\}$ in noisy-free case, which is known as Basis Pursuit (BP) [8]. To guarantee the equivalence of the solutions of (3) and (4), some properties of $\mathbf{A}$ are needed such as Null Space Property (NSP) [31], [32]. NSP can also be used to deal with approximately sparse signals. Moreover, if one should deal with approximately sparse signals in a noisy environment, a stronger property known as Restricted Isometry Property (RIP) [33], [34] can be borrowed from the CS literature. In this noisy case, the constraint in the optimization problem can be relaxed by setting $\mho(\mathbf{y}) = \left\{\mathbf{x} : \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon\right\}$ which is known as Basis Pursuit Denoising (BPDN) [9] or Dantzig Selector [35] if we set $\mho(\mathbf{y}) = \left\{\mathbf{x} : \left\|\mathbf{A}'(\mathbf{y} - \mathbf{A}\mathbf{x})\right\|_\infty \leq \lambda\right\}$. Although RIP can be used for both stability and uniqueness analysis, the calculation of Restricted Isometric Constant (RIC) of $\mathbf{A}$ (defined with RIP of $\mathbf{A}$) generally requires a combinatorial search. Therefore, instead of RIC of a matrix, another important measure of a measurement matrix is defined in the literature. This is a functional $\mu(\mathbf{A}) = \max_{i,j} \left|A_{i,j}\right|$, which is known as coherence. In CS literature, choosing the best measurement matrix Ψ according to sparsifying matrix Φ is well studied in terms of $\mu(\mathbf{A})$. The system in (2) is nothing but a linear dimensional reduction system. To be able to preserve enough information while transforming $\mathbf{x}$ to $\mathbf{y}$, we generally wish each row of matrix $\mathbf{A}$ to get enough information from each element of $\mathbf{x}$.

In other words, the flatness of the rows of **A** is desired. This can be satisfied when the rows of the measurement matrix Ψ is not sparse in Φ. To describe this concept, the functional called "mutual coherence" is defined,

$$\mu\left(\Psi, \Phi\right) := \max_{1\leq k\leq m, 1\leq j\leq N} \left|\left\langle \psi_k, \phi_j \right\rangle\right| \tag{5}$$

which measures the coherence between Ψ and Φ, where ψ_k is the k'th row of Ψ and ϕ_j is the j'th column of Φ. It is clear that $\mu(\Psi, \Phi) \in \left[\frac{1}{\sqrt{N}}, 1\right]$ when Ψ has normalized rows and Φ has normalized columns. A theoretical lower bound to guarantee exact recovery for BP on the number of measurement in terms of defined mutual coherence can be found in [36] and [37] as,

$$m \geq \kappa.N.\mu^2(\Psi, \Phi).k.\log N, \tag{6}$$

where κ is a positive constant. In plain terms, the minimum number of required measurements is dictated by the mutual coherence, and one wishes to keep it a minimum in a CS system. For instance, the measurement matrices, which have random waveforms with *i.i.d* elements such as Gaussian, are well known to be incoherent with any fixed basis, i.e., $\mu\left(\Psi, \Phi\right) \approx \frac{\sqrt{2\log(N)}}{\sqrt{N}}$ [38].

B. Multi-Dimensional and Separable Compressive Sensing

As reviewed above, the mathematical foundation of the conventional CS scheme is well established. However, this traditional scheme, where dimensional reduction is performed via vector-matrix multiplication and recovery is represented via ℓ_1-minimization, may not be convenient in most multi-dimensional signal acquisition schemes such as compressive sensing of imaging systems. For instance, assume that the signal of interest is a 512×512 gray-scale image, **S**. Assume that we wish to build a sub-Gaussian CS system with a measurement rate of $\frac{m}{N} = 0.36$. In that CS scheme that samples the vectorized image, $\mathbf{s} \in \mathbb{R}^N$ with $N = 512^2$, the measurement matrix size will be $m \times N = 94372 \times 262144$. The conventional CS recovery algorithms such as ℓ_1-minimization techniques are iterative algorithms and in each iteration, they perform matrix-vector multiplications using CS matrix and the transpose of it. However, even saving alone such massive size matrices requires more than 80GB of storage. Therefore, the computational complexity of the iterative recovery algorithms becomes cumbersome. As a remedy, block-base CS and separable CS imaging [23] have become the most frequently used approaches. Among them, separable CS (also known as Kronecker CS) has the advantage of introducing fewer blocking artifacts. In a separable CS imaging introduced in [23], the CS sampling operator is separable over horizontal and vertical axes, i.e., $\mathbf{Y} = \Psi_1 \mathbf{S} \Psi_2'$, where $\mathbf{S} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ is the input image, $\mathbf{s} \in \mathbb{R}^N$, in its original matrix form, and $\Psi_1 \in \mathbb{R}^{\sqrt{m}\times\sqrt{N}}$ and $\Psi_2 \in \mathbb{R}^{\sqrt{m}\times\sqrt{N}}$ are the measurement matrices. In that way, the computational cost of the matrix multiplications is reduced from $2 \times m \times N$ flops to $4 \times \sqrt{m} \times N$ flops compared to conventional CS setup. Moreover, this separable CS setup can be easily formulated in a traditional CS setup, which makes the analysis and algorithms of CS theory still valid. For instance, consider that the sparsifying basis is also separable as in 2D DCT matrices, then CS in matrix-vector form is nothing but $vec(\mathbf{Y}) = \Psi_1 \otimes \Psi_2\ vec(\mathbf{S}) = \mathbf{A}_1 \otimes \mathbf{A}_2\ vec(\mathbf{X})$, where $\mathbf{A}_i = \Psi_i\Phi_i$, $\mathbf{X} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ is a sparse coefficient matrix and $\otimes$ is the Kronecker product. Let us assume the separable measurement matrices are Gaussian projection matrices, then mutual coherence between $\Psi_1 \otimes \Psi_2$ and $\Phi_1 \otimes \Phi_2$ can be easily calculated, i.e., $\mu\left(\Phi_1 \otimes \Phi_2, \Psi_1 \otimes \Psi_2\right) \approx \frac{log(N)}{\sqrt{N}}$. Hence, the mutual coherence increases $\sqrt{\frac{1}{2}log(N)}$ times and the number of the necessary measurement increases by the square of it compared to a conventional setup where CS matrix is unfactorized Gaussian projection matrix. *That is to say, in a separable CS setup, although computational complexity decreases, the minimum number of required measurements increases as a trade-off compared to conventional unfactorized CS scheme.* On the other hand, based on RIP or mutual coherence properties, these types of analysis for the CS reconstruction algorithm have been referred to by the term theoretical guarantee conditions in the worst-case scenario [39]. It is generally found, however, that algorithms perform much better than the performance bounds given by these types of worst-case scenario analyses, especially for recovering structurally sparse signals [27], [40].

In general multi dimensional and separable CS setup, the J-dimensional signal, $\mathcal{S} \in \mathbb{R}^{n_1\times n_2\ldots\times n_J}$ with $N = \prod_{j=1}^{J} n_j$, can be acquired by separable sensing operator:

$$\mathcal{Y} = \mathcal{S} \times_1 \Psi_1 \times_2 \Psi_2 \ldots \Psi_{J-1} \times_J \Psi_J, \tag{7}$$

where $\mathcal{S} \times_i \Psi_i$ is the i-mode product of tensor $\mathcal{S}$ and matrix $\Psi_i \in \mathbb{R}^{m_i\times n_i}$ and $\mathcal{Y} \in \mathbb{R}^{m_1\times m_2\ldots\times m_J}$ is CS tensor, with $m = \prod_{j=1}^{J} m_j$. Assuming that the sparsifying basis is also separable, then (7) can be re-cast as,

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \ldots \mathbf{A}_{J-1} \times_J \mathbf{A}_J \tag{8}$$

where $\mathbf{A}_i = \Psi_i\Phi_i$ and $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\ldots\times n_J}$ is the sparse representation tensor. (8) can be cast as a vector-matrix multiplication,

$$\mathbf{y} = \left(\mathbf{A}_1 \otimes \mathbf{A}_2 \otimes \ldots \otimes \mathbf{A}_J\right)\mathbf{x}, \tag{9}$$

where $\mathbf{y} = vec(\mathcal{Y})$ and $\mathbf{x} = vec(\mathcal{X})$. Therefore, the conventional CS recovery techniques defined in (4) can be used and this setup is also known as tensor compressive sensing or Kronecker compressive sensing [21].

III. Generalized Structural Tensor Sum Compressive Sensing

A. Tensor Sum as a Computationally Efficient Approximation of Non-Separable CS Matrix

Earlier, we discussed in detail the trade-off for computational complexity versus the minimum number of measurements when we move from conventional non-separable CS scheme to separable CS setup. In the sequel, to our knowledge for the first time in literature, we will demonstrate that non-separable or unfactorized CS matrix can be approximated with the summation of tensorial sum operation. By doing this, while preserving the "goodness" of the CS matrix (i.e., the probability of exact sparse signal recovery) as possible as close to unfactorized CS case, we can reduce the number of parameters to represent the CS matrix. This will enable us to have a feasible number of learnable parameters when we will attempt to jointly learn CS operation and recovery system using a neural network architecture. Mathematically speaking, let us sum T number of different separable CS tensor obtained from $\mathcal{S}$:

$$\mathcal{Y} = \sum_{t=1}^{T} \mathcal{S} \times_1 \Psi_1^{(t)} \times_2 \Psi_2^{(t)} \dots \Psi_{J-1}^{(t)} \times_J \Psi_J^{(t)}, \tag{10}$$

where $\Psi_i^{(t)}$ is the i^{th} dimension CS matrix of t^{th} operation. (10) can be re-formulated in a non-separable CS setup via

$$\mathbf{y} = \sum_{t=1}^{T} \left(\Psi_1^{(t)} \otimes \Psi_2^{(t)} \otimes \dots \otimes \Psi_J^{(t)} \right) \mathbf{s} = \sum_{t=1}^{T} \mathbf{P}^{(t)} \mathbf{s} = \mathbf{P}\mathbf{s}, \tag{11}$$

where $\mathbf{P}^{(t)} = \left(\Psi_1^{(t)} \otimes \Psi_2^{(t)} \otimes \dots \otimes \Psi_J^{(t)} \right)$ and $\mathbf{P} = \sum_{t=1}^{T} \mathbf{P}^{(t)}$. Note that (11) is nothing but conventional non-separable CS operation with measurement matrix $\mathbf{P}$. For special case where $T = 1$, (10) reduces to a separable CS scheme as in (7). Compared to the conventional CS, $\mathbf{y} = \Psi \mathbf{s}$ with unfactorized CS matrix Ψ, the number of parameters to represent the CS matrix is reduced from $mN = \prod_{j=1}^{J} m_j n_j$ to $T \sum_{j=1}^{J} m_j n_j$. We design an experiment to show how the goodness of the new CS matrix $\mathbf{P}$ is increased with T. For the goodness metric, we selected the probability of exact recovery of the k-sparse signal in our experimental results. As the CS matrices, we selected Gaussian random projection matrices; in the case of Ψ which is unfactorized, $\prod_{j=1}^{J} m_j \times \prod_{j=1}^{J} n_j$ size Gaussian matrix is produced with each element of it is randomly drawn from the Gaussian distribution. For the new case, CS matrix $\mathbf{P}$ is generated with the summation of Kronecker products of the separable random Gaussian matrices as shown in (11). Figure 1 shows us that when T increases the probability of exact recovery from $\mathbf{y} = \mathbf{P}\mathbf{x}$ also increases. The sparse signal length is set to 1024 and orthogonal matching pursuit algorithm [41] was used as the CS recovery algorithm. For $T = 5$, $\mathbf{P}$ can achieve similar performance in recovery when $k = 80$. The exact recovery probability is estimated over 250 trials.

B. Structural Tensor Sum or Transformation Basis as a Part of the CS Matrix

In the literature, adjusting the measurement matrix as the multiplication of two or more matrices is a common practice. For instance, in [42], structural compressive sensing matrices are constructed as the multiplication of random permutation, an orthonormal basis, and sub-sampling matrices. Thanks to such pseudo-random matrices, faster recovery can be possible compared to the CS system with full random matrices. Moreover, in [24] and [25], CS matrix is in the form of multiplication of a sparsifying basis and a random matrix (i.e., an ordinary CS sensing matrix such as Gaussian projection matrix) i.e., $\Psi = \Psi^* \Omega'$, where Ω' is transformation domain basis, and Ψ^* is ordinary random CS matrix. Even though this CS system was originally proposed for CS-based encryption in the frequency domain, in Section V we will discuss that the learned CS systems in the frequency domain may slightly carry more high-frequency details compared to the learned CS system in the spatial domain. If the transformation basis is also separable like DCT, such a system can also be injected in the proposed tensorial and sum of tensorial CS scheme, i.e., $\Psi_i^{(t)} = \Psi_i^{(t)*} \Omega_i^{(t)'}$, where $\Omega_i^{(t)'}$ is i^{th}-coordinate matrix of the separable transformation basis.

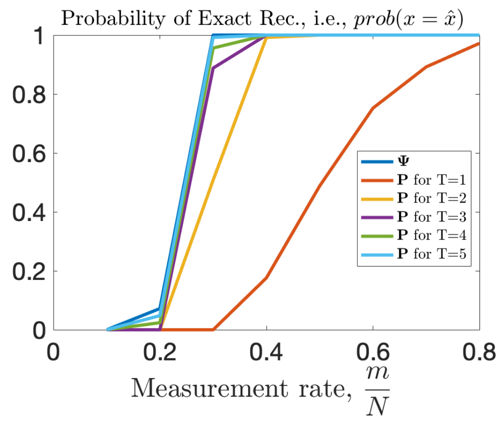


Fig. 1. Estimated probability of exact recovery over 250 trials for different realizations of CS matrix. An exactly sparse signal is synthetically produced for $N = 1024$ and $k = 80$.

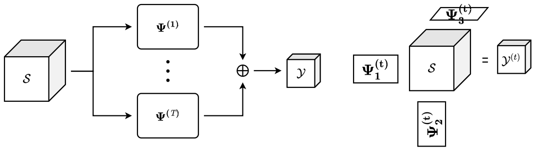


Fig. 2. CS Matrix configuration. Left: The CS operation modeled as the summation of tensor sums. Right: An individual tensor sum for the case of 3D tensors.

C. Generalized Tensor Summation Compressive Sensing Network (GTSNET)

In this section, we propose a neural network architecture that jointly learns the CS sensing mechanism (CS matrix), and the reconstruction of the signal. The proposed network is composed of three parts: i) A CS operation, ii) Adjoint of the CS operation (or coarse estimation of the signal), and iii) a refinement module.

1) Separable and Multi-Linear Learning of CS Operation (i.e., Learnable CS matrix): Our learnable CS matrix $\mathbf{P}$ is factorized as

$$\mathbf{P} = \sum_{t=1}^{T} \left(\Psi_1^{(t)} \otimes \Psi_2^{(t)} \otimes \dots \otimes \Psi_J^{(t)} \right), \tag{12}$$

where $\Psi_i^{(t)} = \Psi_i^{(t)*} \Omega_i^{(t)'}$, $\Omega_i^{(t)'}$ is i^{th}-coordinate matrix of the t^{th} separable transformation matrix such as the one that represents 8×8 size block-wise 2D DCT on the horizontal axis and $\Psi_i^{(t)*}$ learnable i^{th}-coordinate matrix of the t^{th} term in the summation. The CS operation can be factorized using a

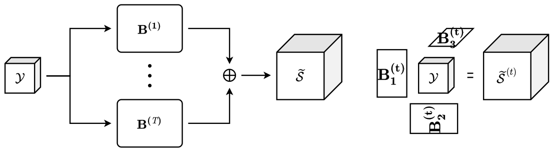


Fig. 3. Course estimation module. Left: The adjoint operation modeled as the summation of tensor sums. Right: An individual tensor sum with an input tensor in 3D.

reasonable number of training parameters thanks to the mode-j product:

$$\mathcal{Y} = \sum_{t=1}^{T} \mathcal{S} \times_1 \left(\Psi_1^{(t)*} \Omega_1^{(t)'}\right) \times_2 \ldots \times_J \left(\Psi_J^{(t)*} \Omega_J^{(t)'}\right). \quad (13)$$

A schematic explanation of (13) is given in Figure 2.

- For special case, $\Omega_J^{(t)'} = \mathbf{I}$, the CS system reduces to an un-structural tensor sum system, which is the learned version of the CS system defined in (10).
- For $T = 1$, the unfactorized system reduces to a separable CS system (e.g., the learned version of the separable CS imaging [23]).
- The system is valid for compressively sensing of any signal, $\mathcal{S}$.
- Thanks to the formulation in (13), the number of trainable parameters for unfactorized CS matrix is reduced from $\prod_{j=1}^{J} m_j n_j$ to $T \sum_{j=1}^{J} m_j n_j$, compared to conventional matrix-vector multiplication formula. Therefore, it makes the learning unfactorized CS sensing possible for large-scale and multi-dimensional signals.

Considering these properties, we call our learnable CS operation Generalized Tensor Summation Compressive Sensing.

2) A Coarse Estimation of the Signal: Separable Learning of Adjoint of CS Operation: In traditional iterative CS reconstruction algorithms, the transposition or the pseudo-inverse of the CS matrix is used in each iteration. On the other hand, in reconstruction-free inference tasks over CS signals [43], [44] or non-iterative deep learning-based recovery algorithms [17], a coarse estimation of the signal, also known as the proxy of the signal is first obtained, i.e., $\tilde{\mathbf{s}} = \Psi'\mathbf{y}$. Although, it is also possible to obtain such a proxy using the regularized pseudo inverse of CS matrix, i.e., $\tilde{\mathbf{s}} = \left(\Psi'\Psi + \lambda\mathbf{I}\right)^{-1} \Psi'\mathbf{y}$ [45], we follow the notation with the transpose or adjoint operator in general for simplicity. Eventually, the adjoint operator will be a learnable linear transformation. The adjoint of the $\mathbf{P}$ that is defined with the factorization in (12) can simply be expressed as,

$$\begin{aligned}\mathbf{P}' &= \sum_{t=1}^{T} \left(\Psi_1^{(t)} \otimes \Psi_2^{(t)} \otimes \ldots \otimes \Psi_J^{(t)}\right)' \\ &= \sum_{t=1}^{T} \left(\Psi_J^{(t)'} \otimes \Psi_{J-1}^{(t)'} \otimes \ldots \otimes \Psi_1^{(t)'}\right). \end{aligned} \quad (14)$$

where $\Psi_i^{(t)'} = \Omega_i^{(t)} \left(\Psi_i^{(t)*}\right)'$, $\Omega_i^{(t)}$ is i^{th}-coordinate matrix of the inverse of the t^{th} separable transformation matrix such as the one that represents the inverse operation of the 8 × 8 size block-wise 2D DCT transformation on the horizontal axis and $\left(\Psi_i^{(t)*}\right)'$ is the transpose of $\left(\Psi_i^{(t)*}\right)$. As stated above we introduce to learn the adjoint CS matrix from the training set. Mathematically speaking, we wish to have the proxy signal, $\tilde{\mathbf{s}} = \mathbf{B}\mathbf{y}$ where the operation $\mathbf{B}$ is learned by a neural network instead of directly applying $\mathbf{P}'$. In practice, there is no need to formulate it in vector-matrix multiplication formulation since the adjoint can be applied directly on the tensorized measurement. As shown in Figure 3, it can be expressed as,

$$\tilde{\mathcal{S}} = \sum_{t=1}^{T} \mathcal{Y} \times_1 \left(\Omega_J^{(t)'} \mathbf{B}_1^{(t)*}\right) \times_2 \ldots \times_J \left(\Omega_J^{(t)} \mathbf{B}_J^{(t)*}\right), \quad (15)$$

where $\mathbf{B}_i^{(t)*}$ is the i^{th}- coordinate learnable adjoint operation matrix for the t^{th} term and $\Omega_i^{(t)}$ is the corresponding fix (non-trainable) inverse transformation operation. As it was in the case of CS operation, the tensorial factorization in (15) makes the adjoint operation trainable instead of attempting directly to learn the elements of the unfactorized matrix $\mathbf{B}$.

3) Reconstruction Free Recovery With Deep Neural Network: Having the proxy signal, $\tilde{\mathcal{S}}$, we train a conventional CNN, $C(.)$, which takes the proxy signal, $\tilde{\mathcal{S}}$ as input and produce a finer estimation of the signal, i.e., $\widehat{\mathcal{S}} = C(\tilde{\mathcal{S}})$. In that way, a non-iterative reconstruction network and CS operation can jointly be optimized (learned). The final network is called Generalized Tensor Summation Compressive Sensing Network (GTSNET-T) which includes T tensor summation in its formula. Our solution introduces a generalized but flexible learning paradigm, it encapsulated many special cases which are set as the adjustable parameter of the network. For instance, one can set $T = 1$ to have a separable optimal CS operation and its reconstruction. Or alternatively, the transformation $\Omega_i^{(t)}$ can be set to the identity operator to sense in the spatial domain.

For the refinement module, $C(.)$, we incorporate a modified version of the Residual Dense Network (RDN) [46]. Such network takes advantage of the so-called residual dense blocks (RDBs), within which all the layer outputs are fully utilized via local feature fusion. The outputs of each RDB are further connected via a global feature fusion, where the information from each block is effectively preserved. The performance of the network is improved by both local and global residual learning. We modify the original RDN configuration by omitting the upscale layer as it was proposed for image super-resolution [46]. Furthermore, we adapt the overall RDN structure as a residual network, i.e., the output of the modified RDN is added to the input proxy signal, $\tilde{\mathcal{S}}$, to obtain $\widehat{\mathcal{S}}$.

The overall GTSNET-T structure is illustrated in Figure 4, including the learnable CS matrix and the adjoint operation, as well as the final refinement module. The adjoint operation matrices $\Omega_j^{(t)}\mathbf{B}_j^{(t)*}$ are denoted altogether as $\mathbf{B}_j^{(t)} = \Omega_j^{(t)}\mathbf{B}_j^{(t)*}$ for simplicity. Each branch t in the CS matrix (pink blocks in Figure 4) performs a single tensor product with the input tensor $\mathcal{S}$, while the final CS operation is the summation over the products as dictated by (10) and (11). Similarly, the adjoint operation is the summation over the individual tensor products with the compressed signal $\mathcal{Y}$ (green blocks in Figure 4), as given by (15). The refinement module composes of D RDBs, each of which having C convolution layers with

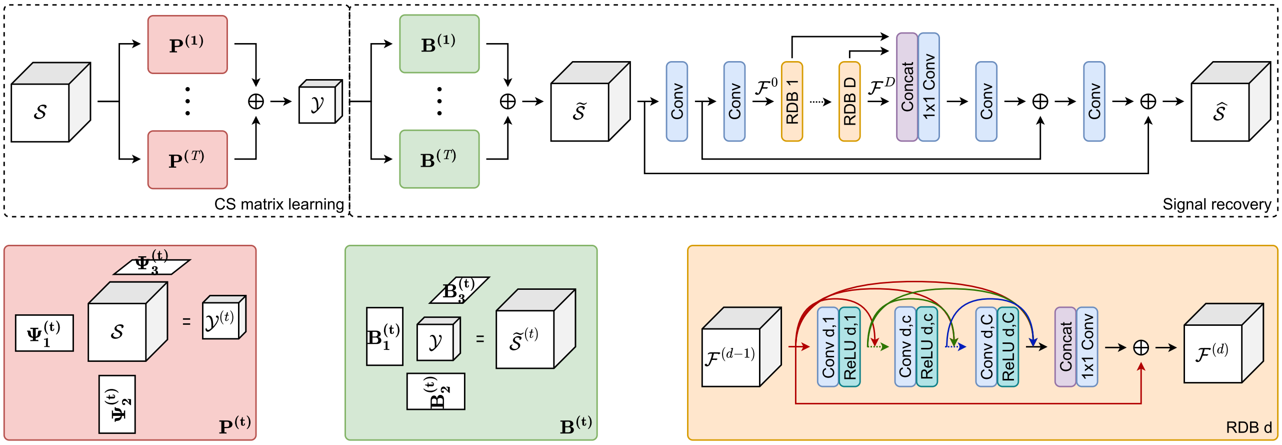


Fig. 4. Overall block diagram of the proposed end-to-end system.

rectified linear units (ReLUs) [47] as activation functions. The input feature map to each RDN, $\mathcal{F}^{(d-1)}$ has G_0 channels, while each convolution layer inside has G filters with a 3×3 filter size. The concatenated feature maps at the end of the RDB are processed through one 1×1 convolution layer to map the output channel size back to G_0. In our implementation, we set $\mathrm{D} = 4$, $\mathrm{C} = 3$, $\mathrm{G}_0 = 30$, and $\mathrm{G} = 12$.

We train the network with an overall loss function $L(\tilde{\mathcal{S}}, \hat{\mathcal{S}}, \mathcal{S})$, which is the combination of two loss functions defined over the proxy signal, $\tilde{\mathcal{S}}$, and the final output, $\hat{\mathcal{S}}$, respectively. The loss over the proxy signal, $\tilde{L}(\tilde{\mathcal{S}}, \mathcal{S})$, is set to be a simple L_1-loss, as it was previously demonstrated to achieve better performance compared to the L_2-loss in various image processing problems [48]. Assuming each training batch contains K input tensors $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K\}$,

$$\tilde{L}(\tilde{\mathcal{S}}, \mathcal{S}) = \frac{1}{K} \sum_{k=1}^{K} \|\tilde{\mathbf{s}}_k - \mathbf{s}_k\|_1 \,. \tag{16}$$

The loss over the final output, $\hat{L}(\hat{\mathcal{S}}, \mathcal{S})$, is also an L_1-loss with an additional regularization term, $\hat{R}(\hat{\mathcal{S}}, \mathcal{S})$,

$$\hat{L}(\hat{\mathcal{S}}, \mathcal{S}) = \frac{1}{K} \sum_{k=1}^{K} \left(\|\hat{\mathbf{s}}_k - \mathbf{s}_k\|_1 + \alpha \hat{R}(\hat{\mathcal{S}}_k, \mathcal{S}_k) \right), \tag{17}$$

where α is a hyperparameter. Our regularization term is a modified sparse gradient prior [49] applied on the spatial domain, which has been proposed for image debluring as it provides sharper details compared to, e.g., Gaussian prior. The mathematical description of the regularization is expressed as follows:

$$\begin{aligned} \hat{R}(\hat{\mathcal{S}}, \mathcal{S}) = & \sum_{n_1,\ldots,n_J} \exp\left(-\beta |\nabla_{n_1} \mathcal{S}|^{\gamma}\right) |\nabla_{n_1} \hat{\mathcal{S}}|^{\gamma} \\ & + \sum_{n_1,\ldots,n_J} \exp\left(-\beta |\nabla_{n_2} \mathcal{S}|^{\gamma}\right) |\nabla_{n_2} \hat{\mathcal{S}}|^{\gamma}, \end{aligned} \tag{18}$$

where ∇_{n_1} and ∇_{n_2} are the discrete differential operators over the first and second dimensions, respectively. The exponential weights $\exp\left(-\beta |\nabla_{n_1} \mathcal{S}|^{\gamma}\right)$ and $\exp\left(-\beta |\nabla_{n_2} \mathcal{S}|^{\gamma}\right)$ are introduced to decrease the prior term over the edges of the original tensor $\mathcal{S}$, as proposed in [49]. We empirically set, $\alpha = 0.005$, $\beta = 10$, and $\gamma = 0.9$. Finally, the overall loss function is $L(\tilde{\mathcal{S}}, \hat{\mathcal{S}}, \mathcal{S}) = \tilde{L}(\tilde{\mathcal{S}}, \mathcal{S}) + \hat{L}(\hat{\mathcal{S}}, \mathcal{S})$.

IV. Experimental Setup and Results

A. Training Setup

We prepare the training dataset in the following manner: Div2K image dataset was used and 256×256 image patches were selected with stride 512 and they were cropped. Data augmentation was applied during the data generation with rotations in four different degrees; 0, 90, 180, and 270, flipping and downsampling with scale factors; 1, 0.8, and 0.6. Hence, by using the training set of DIV2K total of 89272 image patches were obtained to be used as the training set. Similarly, as the validation set, we obtained 1512 images from the validation set of DIV2K. All the images are normalized to range $[0, 1]$. The batch size was selected as 16 and the networks were trained with 100 epochs. During training, the learning rates are scheduled to be 10^{-3} for the first 50 epoch, 10^{-4} for the later 30 epoch, and 10^{-5} for the last 20 epoch. The network of the 100. epoch was chosen as final. The implementation of the GTSNET was done using MatConvNet package [50].

B. Comparative Evaluations

As traditional CS reconstruction methods, which are well-known state-of-the-art sparse recovery methods, comparative evaluations are performed against the following three methods; Gradient Projection for Sparse Reconstruction (GPSR) [14], TV Minimization by Augmented Lagrangian and Alternating Direction Algorithms (TVAL3) [12] and Denoising-based AMP (D-AMP) [13]. GPSR is a sparse recovery algorithm that was specifically proposed as computationally more efficient and feasible to apply for any image CS framework. As the CS matrix, a randomly selected subset of the rows of noiselet basis [51] was used. As the sparsifying transform, wavelet "Coiflet 2" was used with the toolbox WaveLab850 [52]. TVAL3 is one of the state-of-the-art TV minimization solvers. Walsh Hadamard Transform whose fast implementation available in the TVAL3 toolbox was used as the CS matrix. The parameters on TVAL3 toolbox were set as follows: $\mu = 2^{13}$,

TABLE I

Performance Metrics (PSNR and SSIM) Obtained by the Competing and Proposed Methods Over Three Benchmark Datasets

Measurement Rates (MRs)	Datasets	GPSR		TVAL3		DAMP		CSNET+		SCSNET		GTSNET-1		GTSNET-3	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MR = 0.01	SET5	16.25	0.378	17.08	0.552	8.30	0.066	24.18	0.669	24.21	0.669	24.61	0.696	**24.66**	**0.697**
	SET14	16.61	0.349	16.46	0.474	7.69	0.041	22.93	0.588	22.97	0.588	**23.08**	**0.600**	**23.08**	**0.600**
	SET11	14.07	0.289	13.94	0.441	5.61	0.024	21.02	0.589	21.04	0.589	**21.39**	**0.609**	21.38	**0.609**
	Avg.	15.64	0.339	15.83	0.489	7.20	0.044	22.71	0.615	22.74	0.615	23.03	**0.635**	**23.04**	**0.635**
MR = 0.05	SET5	20.58	0.413	23.44	0.661	26.56	0.766	29.75	0.848	29.74	0.847	**30.25**	**0.861**	30.24	**0.861**
	SET14	20.17	0.361	19.95	0.555	24.70	0.658	27.04	0.739	27.04	0.739	**27.16**	**0.747**	27.15	**0.747**
	SET11	17.69	0.307	17.27	0.576	21.77	0.684	25.86	0.788	25.86	0.787	**26.27**	**0.806**	26.25	**0.806**
	Avg.	19.48	0.360	20.22	0.597	24.34	0.703	27.55	0.792	27.55	0.791	**27.89**	**0.805**	27.88	**0.805**
MR = 0.1	SET5	23.18	0.505	26.00	0.743	31.42	0.872	32.60	0.906	32.78	0.908	33.03	0.912	**33.11**	**0.913**
	SET14	22.31	0.438	20.76	0.620	28.77	0.769	29.24	0.820	**29.32**	**0.821**	29.17	**0.821**	29.24	**0.821**
	SET11	20.05	0.396	18.59	0.670	26.17	0.852	28.34	0.859	28.52	0.862	28.79	0.871	**28.83**	**0.872**
	Avg.	21.85	0.446	21.78	0.678	28.79	0.831	30.06	0.862	30.21	0.864	30.33	0.868	**30.39**	**0.869**
MR = 0.2	SET5	26.76	0.659	27.92	0.823	35.26	0.926	36.07	0.949	36.17	0.949	**36.29**	**0.950**	36.27	**0.950**
	SET14	25.23	0.578	23.04	0.717	32.05	0.851	32.26	0.896	**32.30**	**0.897**	31.88	0.893	31.94	0.894
	SET11	23.49	0.561	20.85	0.789	27.89	0.913	31.67	0.921	31.83	0.922	31.81	**0.924**	**31.86**	0.923
	Avg.	25.16	0.599	23.94	0.776	31.73	0.897	33.33	0.922	**33.43**	**0.923**	33.33	0.922	33.36	0.922
MR = 0.3	SET5	29.55	0.763	26.27	0.843	36.86	0.946	38.29	0.965	**38.49**	**0.966**	37.95	0.964	38.00	0.964
	SET14	27.61	0.690	23.70	0.745	33.33	0.894	34.46	0.931	**34.64**	**0.933**	33.64	0.926	33.74	0.927
	SET11	26.48	0.692	23.21	0.854	27.11	0.943	34.32	0.950	**34.66**	**0.952**	34.09	0.951	33.98	0.950
	Avg.	27.88	0.715	24.39	0.814	32.43	0.928	35.69	0.949	**35.93**	**0.950**	35.23	0.947	35.24	0.947

$\beta = 2^6$, $\mu_0 = 2^2$, $\beta_0 = 2^{-2}$, $tol = 10^{-6}$, $maxit = 300$. D-AMP was proposed to improve the performance of CS recovery for the natural signals by employing off-the-shelf denoising algorithms. We test the algorithm with default settings, where the elements of the CS matrix are picked from *i.i.d.* Gaussian distribution and BM3D [53] is used as the denoiser. The number of iterations and the image block size are empirically set as 30 and 128 × 128, respectively.

As the akin state-of-the-art deep learning methods, we selected CSNET [19] and SCSNET [20]. The algorithms and the trained models were taken from the competing algorithms' web pages. Both methods jointly learn the CS matrix and reconstruction of the image from the measurement as proposed in this study. However, these methods learn the block-wise CS matrix using convolution operation in a non-overlapping manner. In that sense, when the kernel size is increased to full image size, the method turns out to be the classical unfactorized CS setup with an infeasible increase in the number of parameters to train.

We trained two GTSNET versions; GTSNET-1 and GTSNET-3. Among them GTSNET-1 learns tensorial representation of CS matrix, therefore suitable for both separable and unfactorized CS schemes. For this network, separable transformation matrices Ω_1' and Ω_2' were chosen as 8 × 8 DCT transformation matrices in the horizontal and vertical directions, respectively. GTSNET-3 includes the three-tensor summation as the CS operation and represents an unfactorized CS setup. As the sparsifying matrices, $\Omega_i^{(t)'}$, we selected 8 × 8, 16 × 16 and 32 × 32 2D DCT transformations for $t = 1, t = 2, t = 3$, respectively. All the competing algorithms were tested on three commonly-used datasets: Set14 [54], Set5 [55], and Set11 [17]. The results on five different measurement rates (MRs) are presented in Table I. Against the competing traditional methods, GPRS, TVAL3 and DAMP, a significant gap on the average performance is observed. In particular, we achieve 7.21 dB, 4.55 dB, 1.6 dB, 1.63 dB, and 2.81 dB improvements in PSNRs compared to the closest performance, for MRs of 0.01, 0.05, 0.1, 0.2 and 0.3, respectively. When we compare against the deep learning-based competing methods, CSNET+ and SCSNET, GTSNET-T shows superiority for the lower MRs (< 0.2), i.e., 0.3 dB, 0.3 dB, 0.18 dB PSNR improvement over the best competing method, for the MRs of 0.01, 0.05 and 0.1, respectively. Figure 1 presents visual comparisons over the state-of-the-art CS methods. Although there is no significant gap between the PSNR and SSIM of GTSNET-1 and GTSNET-3 results, one can observe GTSNET-3 outputs preserve high frequency details better, e.g., see Parrot and Flinstone images in Figure 5 and Figure 6. The performance gap in both PSNR and SSIM measures becomes significant in RGB images while the visual quality of the GTSNET-3 outputs especially at the fine details noticeably improves.

C. Comparative Evaluations Against Deep Learning-Based CS Methods

As the competing deep learning-based solutions,

(i) the stacked denoising autoencoder (SDA) [16], which is the pioneer method,

(ii) non-iterative reconstruction of the compressively sensed images using CNN (ReconNet) [17]

(iii) the learned version of iterative shrinkage thresholding algorithm for CS imaging (ISTA-Net),

(iv) akin state of the art techniques convolutional compressive sensing network (CSNET) [19]

(v) scalable convolutional compressive sensing network (SCSNET) [20],

(vi) memory augmented cascading Network (MAC-Net) [56],

(vii) dual-path attention network for compressed sensing (DPA-Net) [57], and the most recent deep unrolling techniques,

(viii) OPINE-NET [58],

(ix) AMP-Net+ [59], and

(x) COAST [60]

are selected as the most recent techniques. This article was updated to include a few more deep learning-based solutions [61], [62] while it was being prepared. Through self-attention mechanisms, these algorithms enhance the performance of recovery refinement parts of end-to-end CS networks based on vision transformers [63], which have recently become an active research area. While learning compressively sensing operator part, however, they use a strided convolution (block-by-block sensing) based module similar (or identical) to the ones used by CSNET, SCSNET, OPINE-NET, and AMP-Net.

For ISTA-Net, CSNET, and AMPNet we choose their improved versions ISTA-Net+, CSNET+, and AMP-Net+, respectively. The comparative evaluations are conducted on the benchmark SET11 dataset. The results for different measurement rates are presented in Table II. All the algorithms and the trained models were downloaded from authors' web pages and run over SET11 except SDA and DPA-Net, whose source codes are not available online. The results of SDA were taken from [17] and the results of DPA-Net were taken from [57]. The average PSNR values show the superiority of the proposed network over all competing methods especially for the case of lower sampling rates, e.g., for MR < 0.1. Figure 6 shows samples for the qualitative performance comparison where it is clear that the outputs of SDA, ReconNet, and ISTA-Net+ may exhibit strong blocking artifacts. The reason is that they use block-by-block sampling strategy to compressively sense the signal, and then apply block-by-block recovery strategy. On the other hand, CSNET, MAC-Net, and SCSNet algorithms have block-by-block compressive sensing setup, but their reconstruction step recovers the image as a whole by using convolutional layers. Therefore, their outputs show fewer blocking artifacts. On the other hand, GTSNET-1 CS module is convenient for both separable and unfactorized (conventional vector-matrix CS system) CS setup. When it comes to reconstruction, it uses a CNN similar to CSNET and SCSNET and recovers the image as a whole. For the use case, where one wants to use a traditional sampling setup with a better approximation of unfactorized CS matrices, the GTSNET-T ($T > 1$) can be used. The sampling strategies of deep learning methods are summarized in Table I. Although there is no significant gap in PSNR and SSIM values on average, GTSNET-3 can recover more high-frequency details as seen in the Parrot image in Figure 6. In Section V, we will discuss the effects of the tensor sum in the frequency domain.

D. Comparative Evaluations Over RGB Images

Unfortunately, most aforementioned competing methods except CSNET were designed only for gray-scale images. Therefore, we compare GTSNET with CSNET. An extensive set of comparative evaluations was conducted on the following benchmark RGB image datasets: Set5, Set11, Manga109 [64], and Urban100 [65]. The results are reported in Table III. For GTSNET-5, as the sparsifying matrices, we selected, 8×8, 16×16, 32×32, 64×64 and 128×128 2D DCT transformations for $\Omega^{(1)'}$, $\Omega^{(2)'}$, $\Omega^{(3)'}$, $\Omega^{(4)'}$ and $\Omega^{(5)'}$, respectively. As clearly observable from the table that the performance gap between GTSNET-1 and GTSNET-T ($T > 1$) widens in terms of PSNR and SSIM. Compared to CSNET+, a comparable performance with the separable CS setup (GTSNET-1) is achieved. For the unfactorized CS matrix setup ($T > 1$), the performance gap between CSNET+ and the best operating GTSNET configuration becomes significant, i.e., 1.15 dB, 0.92 dB and 2.91 dB for sampling rates of 0.05, 0.1 and 0.2, respectively. Moreover, some samples for visual comparison of the recovered images are shown in Figure 5. The outputs of CSNET+ exhibit certain level of blocking artifacts that are entirely absent in any of the outputs of the proposed GTSNET-T networks.

TABLE II

SAMPLING AND RECOVERY STRATEGIES OF THE DEEP LEARNING-BASED ALGORITHMS. GTSNET-T CAN BE USED FOR BOTH CLASSICAL (UNFACTORIZED) CS AND SEPARABLE CS SYSTEMS

Algorithm	**Sampling Strategy**	**Reconstruction Strategy**
SDA	Block-by-block CS	Block-by-block Auto-encoder
ReconNet	Block-by-block CS	Block-by-block CNN
ISTA-Net+	Block-by-block CS	Block-by-block CNN
MAC-Net	Block-by-block CS	CNN
CSNET+	Block-by-block CS	CNN
SCSNET	Block-by-block CS	CNN
GTSNET-1	Unfactorized or separable CS	CNN
GTSNET-T ($T > 1$)	Unfactorized CS	CNN

TABLE III

PSNR LEVELS OBTAINED BY THE COMPETING AND PROPOSED METHODS OVER SET11 DATASET

Algorithm	Measurement Rate			
	0.25	0.1	0.04	0.01
SDA	25.34	22.65	20.12	17.29
ReconNet	25.60	24.28	20.63	17.27
ISTA-Net+	32.44	26.49	21.56	17.45
DPA-Net	31.74	26.99	23.50	18.05
MAC-Net	32.91	27.68	24.22	18.26
CSNET+	-	28.34	-	21.02
SCSNET	-	28.52	-	21.04
OPINE-NET	**34.44**	29.33	25.04	19.87
AMP-Net+	-	29.40	25.26	20.20
COAST	-	**30.03**	-	-
GTSNET-1	32.47	28.79	25.44	**21.39**
GTSNET-3	32.36	28.83	**25.45**	21.38

V. DISCUSSION

A. Tensor Vs Tensor Sum for CS Matrix Learning

In this section, we perform an ablation study concerning the effects of the number of tensor sums, T, over the final reconstruction quality. As a starting point, we plot the PSNR values of each image in Set5, sensed and reconstructed via three different setups, for $T = 1, T = 3$, and $T = 5$. We perform analysis on both gray-scale and RGB images,

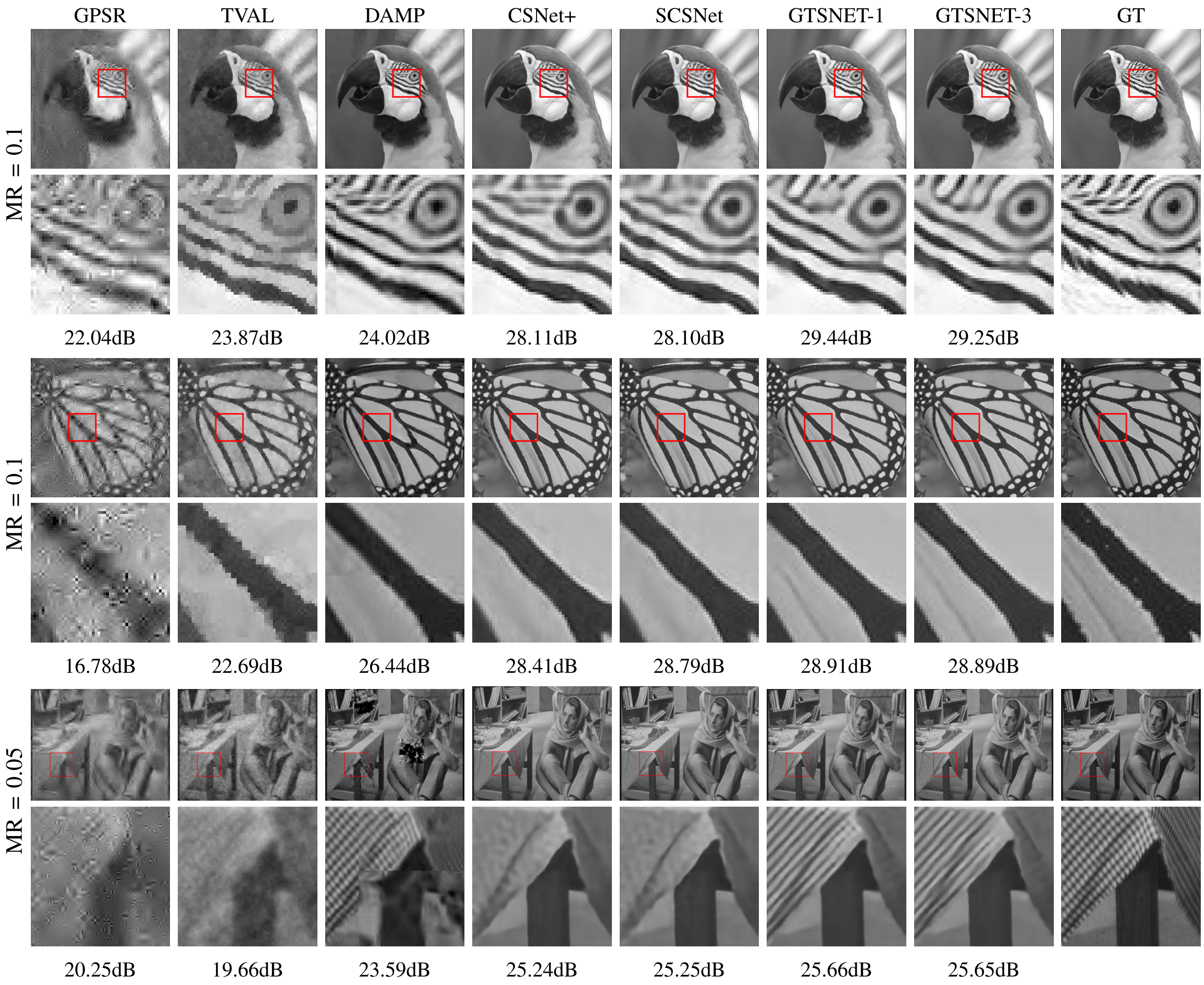


Fig. 5. Visual comparison with the state-of-the-art on grayscale images with varying measurement rates.

TABLE IV

Performance Metrics (PSNR and SSIM) Obtained by the Competing and Proposed Methods Over Four Benchmark RGB Image Datasets

Ratio	MR = 0.01								MR = 0.05							
Method / *Dataset*	CSNET+		GTSNET-1		GTSNET-3		GTSNET-5		CSNET+		GTSNET-1		GTSNET-3		GTSNET-5	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SET5	**24.35**	0.858	23.17	0.843	24.23	**0.865**	24.16	**0.865**	29.18	0.940	28.53	0.935	29.82	0.948	**30.48**	**0.955**
SET14	**22.83**	0.734	21.55	0.707	22.62	**0.736**	22.49	0.732	26.71	0.869	25.43	0.818	26.41	0.849	**27.15**	**0.872**
Manga109	21.18	0.821	20.43	0.825	**21.38**	0.842	21.35	**0.843**	25.41	0.914	26.04	0.929	27.48	0.946	**28.36**	**0.953**
Urban100	**20.93**	**0.697**	19.51	0.649	20.61	0.692	20.31	0.686	**25.05**	0.857	23.01	0.794	24.40	0.843	24.97	**0.858**
Avg.	**22.32**	0.778	21.17	0.756	22.21	**0.784**	22.08	0.782	26.59	0.895	25.75	0.869	27.03	0.897	**27.74**	**0.910**
	MR = 0.1								MR=0.2							
SET5	32.07	0.966	31.16	0.959	**32.97**	**0.971**	32.45	0.968	35.15	0.979	34.13	0.975	35.33	0.981	**37.28**	**0.986**
SET14	29.31	**0.926**	27.30	0.870	**29.35**	0.922	28.53	0.898	32.26	0.960	29.79	0.922	31.35	0.944	**34.15**	**0.972**
Manga109	28.82	0.957	29.49	0.961	**31.63**	**0.976**	31.04	0.973	30.96	0.969	33.62	0.983	34.67	0.987	**36.11**	**0.990**
Urban100	**27.78**	**0.918**	25.13	0.861	27.74	**0.918**	26.88	0.903	30.64	0.955	27.94	0.920	29.94	0.949	**33.11**	**0.972**
Avg.	29.50	0.942	28.27	0.913	**30.42**	**0.947**	29.73	0.936	32.25	0.966	31.37	0.950	32.82	0.965	**35.16**	**0.980**

where the measurement rate is set as 0.1. The gray-scale images are constructed via taking the luminance channel of each image in YCbCr color space. The results are shown in Figure 6. While the performance gap is negligible difference on the gray-scale images, we observe a significant performance improvement in reconstructing the RGB images as T increases from 1 to 3, e.g., up to 2.74 dB PSNR improvement on the "woman" image. An interesting observation worth mentioning is that GTSNET-T with $T = 3$ outperforms the one with $T = 5$, both for each individual image in Set5, and for the average of each dataset presented in Table III. This might seem at first contradictory to our derivations within the theoretical discussions, where we demonstrate in Section III-A and Figure 1 that the mutual coherence decreases as T increases.

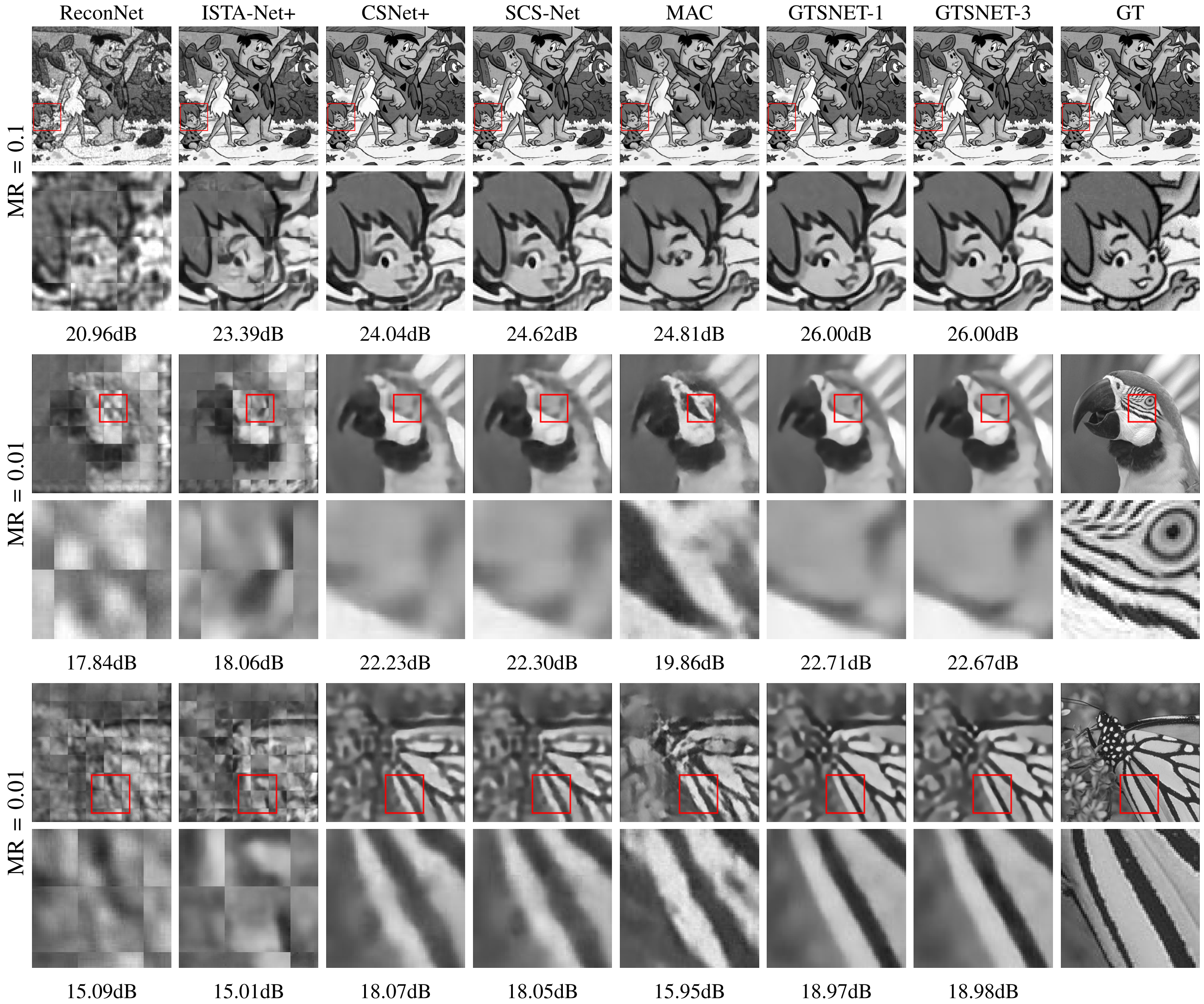


Fig. 6. The recovered images of the competing and proposed methods with the GT image on the left.

However, within such analysis each tensor is chosen to be composed of Gaussian random matrices. As in practice we further learn the CS matrices to improve performance over the random matrices, an inherent trade-off occurs, where the number of learned parameters increases linearly with increasing T. The experimental analysis shows that $T = 3$ provides best of both worlds with a consistently superior image quality for MR $= 0.1$.

To enrich the discussion above, we visually compare the three methods over a rather tricky color image: "zebra" from Set14. In particular, we examine the frequency responses of both the proxy signals, $\widetilde{\mathcal{S}}$, as the immediate reconstruction, and the final outputs, $\widehat{\mathcal{S}}$. The results are shown in Figure 9. The differences between each method are clearly visible over the frequency responses of the proxy signals (middle row), where the cut-off frequency of GTSNET-3 is higher compared to that of GTSNET-1 and GTSNET-5. Subsequently, the final output of GTSNET-3 can preserve the higher frequency information while providing better quantitative result in terms of PSNR value.

Having shown the improvement in image quality with $T > 1$, we now provide the information flowing from each branch of the adjoint operator. Figure 10 illustrate the tentative reconstruction results, where one of the branches ($\mathbf{B}^{(3)}$) performs the majority of the reconstruction over the lower frequency region and the residual high-frequency details are recovered through branches $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$. In addition, the first and second branches carry information regarding the different regions of the spectrum; the support of $\mathbf{B}^{(2)}$ is more concentrated towards the low-frequency region, whereas the frequency response of the first branch contains higher frequencies. The proximal signal, $\widetilde{\mathcal{S}}$ (Figure 10, fourth column), is the summation of each output, having a wider response than each individual branch.

B. Tensor Sum vs Structural Tensor for CS Matrix Learning

In Section III-C1, we discuss that the proposed method is suitable for designs of both structured and unstructured tensor summations, whereas we mainly demonstrate our results via the learned structured matrices. In this subsection, we compare a network trained for unstructured tensor sums, i.e., $\Omega_J^{(t)'} = \mathbf{I}$, with the previously discussed structured tensors. We train both setups for GTSNET-3, where the measurement rate is set as

TABLE V

PERFORMANCE COMPARISON OF TIED VS. UNTIED LEARNING OF CS MATRICES

Algorithm	GTSNET-T										ℓ_1-minimization
Measurement M. / Adjoint	A, B are learned with constraint $B = A^T$ (Tied Learning)					A, B are learned without constraint					A is Gaussian matrix, $B = A^T$
Measurement Rates (MRs)	T=1	T=2	T=3	T=4	T=5	T=1	T=2	T=3	T=4	T=5	Unfactorized
MR=0.01	16.81	17.28	17.15	17.40	17.54	17.33	17.44	17.50	17.50	**17.80**	9.85
MR=0.04	23.62	23.70	23.86	23.84	23.81	23.95	23.86	23.98	**24.42**	24.35	9.96
MR=0.05	25.30	25.30	25.29	25.55	25.51	25.62	25.46	25.68	25.66	**25.71**	10.00
MR=0.1	28.11	28.12	28.24	28.28	28.08	28.55	28.39	28.24	28.50	**28.56**	10.25

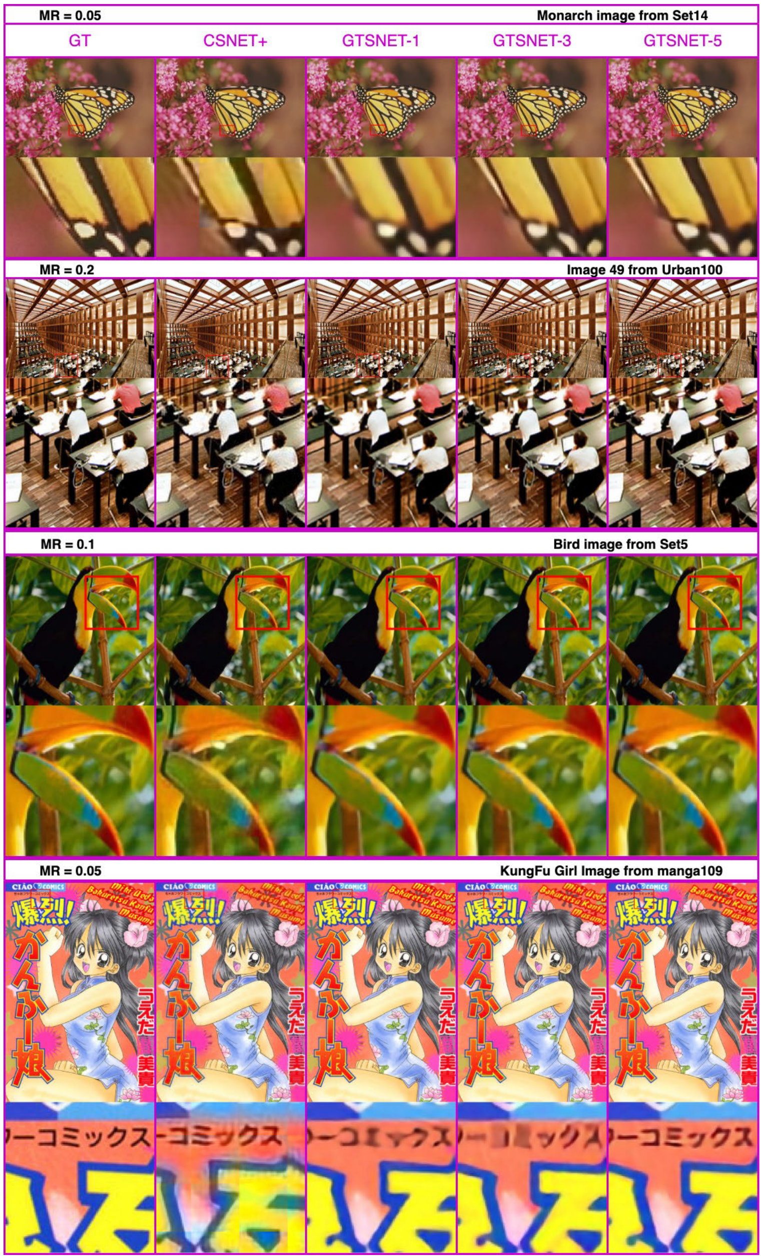


Fig. 7. The recovered images of the competing and proposed methods with the GT image on the left.

0.1. A test image is picked from the Urban100 dataset for the experiment, carrying high-frequency components with fixed patterns. Figure 11 shows the results. While the differences are not prominent through visual inspection on the spatial domain at first, the proxy output of the structured tensor summation is observed to contain a wider frequency response. We also notice a decrease in the high-frequency region of the final output with the unstructured tensor sum, visible as a box in the middle of the frequency response (Figure 11, bottom left), and a decrease of 0.43 dB in PSNR. For such

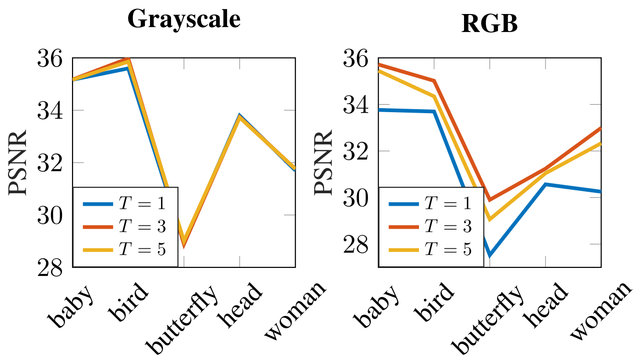


Fig. 8. Quantitative reconstruction results of Set5 dataset, with varying number of tensor sums T. Left: Gray-scale (luminance) images. Right: Original color (RGB) images. The measurement rate is set to be 0.1 in both scenarios.

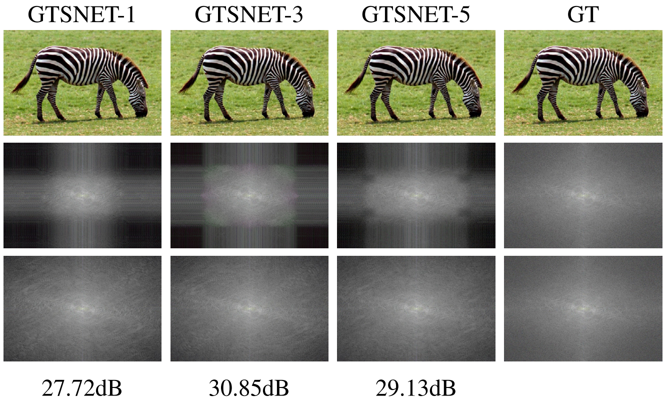


Fig. 9. Frequency analysis with varying number of tensor sums. Top: Final outputs $\widehat{S}$ in spatial domain. Middle: Proxy signals $\widetilde{S}$ in frequency domain. Bottom: Final outputs $\widehat{S}$ in frequency domain. Measurement rate is set as 0.1.

purposes, we proceed with the structured matrices. Nevertheless, it is important to demonstrate that the proposed method has flexibility generalizing various types of CS matrix designs.

C. Performance on Sparse Signals

The proposed GTSNET-T scheme is able to sense the images in sparsifying domain e.g., DCT basis directly. The adjoint operator consists of two parts; one is a learned adjoint matrix and the other is the fixed inverse transform basis of the sparsifying transform. Therefore, the coarse estimation of the image is also done in sparsifying, firstly. However, the natural images do not exhibit exact sparsity but approximate sparsity in any sparsifying domain. This is why in the sequel, we will investigate the the performance of the GTSNET-T in the recovery of exact sparse signals.

In this section, we use the MNIST dataset for our experiments. Images in this dataset have resolutions of 28×28 pixels, and intensities ranging from 0 to 1. The background of each image covers a larger area than the foreground, making it a sparse signal in the canonical basis. Non-zero coefficients to vectorized signal dimension ratios (i.e., $\frac{k}{N}$) range from 0.05 to 0.4 [45]. Therefore it is actually a challenging sparse signal dataset for the traditional CS setup since it includes a large number of less sparse samples. The dataset includes 70000 samples, among them 50000 were used to train the networks, and 20000 were used for the test.

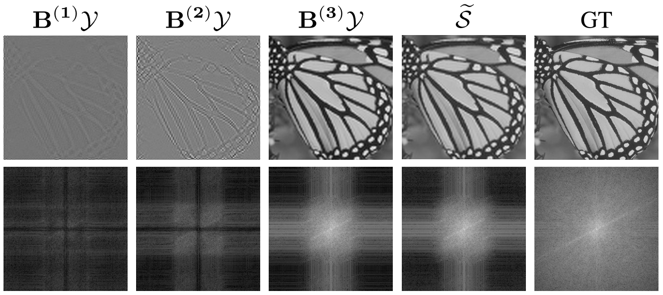

$\mathbf{B}^{(1)}\mathcal{Y}$
$\mathbf{B}^{(2)}\mathcal{Y}$
$\mathbf{B}^{(3)}\mathcal{Y}$
$\widetilde{S}$
GT

Fig. 10. Tentative reconstruction result of each individual adjoint operation in GTSNET-3, as well as the proxy signal $\widehat{S}$ as the summation of each branch.

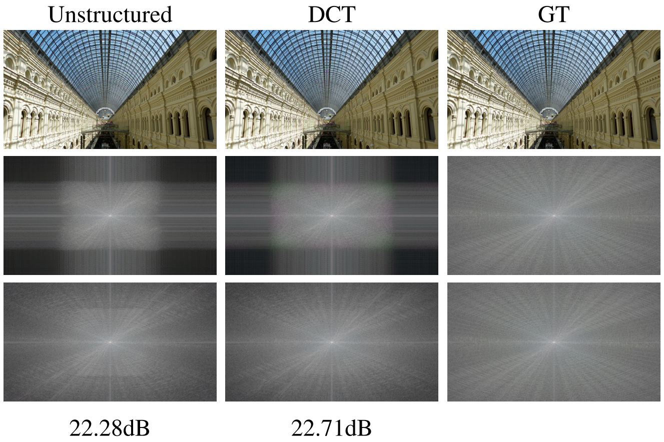


Fig. 11. Visual comparison between the unstructured and structured (DCT) Tensor sums. Top: Final outputs $\widehat{S}$ in the spatial domain. Middle: Proxy signals $\widetilde{S}$ in the frequency domain. Bottom: Final outputs $\widehat{S}$ in the frequency domain. The results are shown for GTSNET-3 with MR = 0.1.

1) Tied Learning ($\mathbf{B} = \mathbf{A}'$) *vs Untied Learning* $\mathbf{B} \neq \mathbf{A}'$: In conventional solutions for linear inverse problems, the adjoint operator is taken as the transpose of the linear degradation matrix $\mathbf{A}$, i.e., $\mathbf{B} = \mathbf{A}^T$. However, the adjoint $\mathbf{B}$ is also learned without any constraint depending on $\mathbf{A}$. In this section, we will investigate the advantages/disadvantages of such a learning scheme instead of tied learning where $\mathbf{A}$ and $\mathbf{B}$ are learned with a constraint $\mathbf{B} = \mathbf{A}^T$. For the MNIST dataset, we assume that the i^{th} vectorized sparse signal $\mathbf{x}_i \in \mathbb{R}^{N=784}$ is compressively sensed, $\mathbf{y_i} = \mathbf{Ax_i}$, where $\mathbf{A} \in \mathbb{R}^{m \times N}$ for different m values. The measurement matrix $\mathbf{A}$ is selected in three different ways: a) Unfactorized full-size Gaussian matrix. b) Learned CS matrix with GTSNET-T with constraint $\mathbf{B} = \mathbf{A}^T$. c) Learned CS matrix with GTSNET-T without such constraint.

As an example let us consider that one wants to take $m = 81$ measurements. In this setup, the number of parameters to represent the unfactorized conventional Gaussian matrix can be calculated as $81 \times 728 = 58968$. On the contrary for the learning matrix with $T = 3$-tensor sum scheme will be $2 \times T \times 9 \times 28 = 1512$ since each Kronecker product is represented by left and right matrix multiplications with the matrices of size 9×28. If there is no constraint in the adjoint operator then the number of parameters to represent the abjoint B will be the same. In the meanwhile, when there is a such constraint then, there is no additional parameter in order to represent the adjoint matrix.

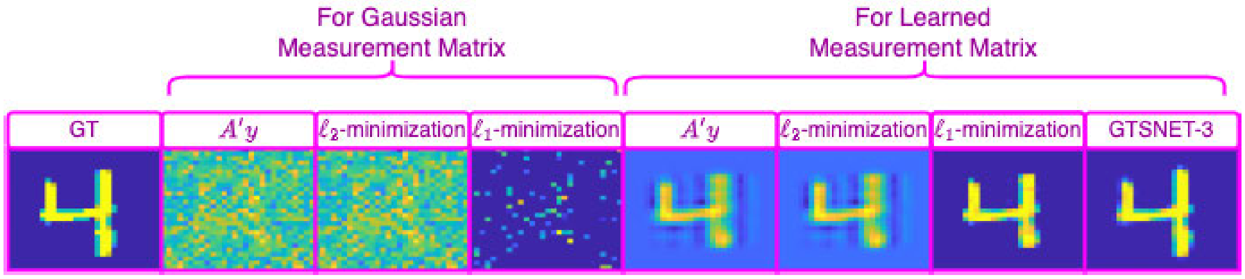


Fig. 12. Comparison of Gaussian measurement matrix and learned measurement matrix: Proxy, minimum norm solution, ℓ_1-minimization results from the measurement vector **As** when **A** is Gaussian matrix (on the left) and when **A** is learned with proposed tensorial sum approach (on the right).

Table V shows the comparison of the reconstruction performance of these three setups with different measurement rates and with different T values. For the conventional CS setup, we used ℓ_1-minimization to solve the recovery problem: $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.t $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$. There is a clear pattern that shows that untied learning can achieve superior performance with increasing margin, especially when MR gets smaller. On the other hand, conventional CS setup performs very poorly due to the fact that the data sparsity ratio makes the sparse recovery very challenging.

2) Learned Measurement Matrix Within ℓ_1-Minimization Based Sparse Recovery: In order to compare the goodness of the learned measurement matrices to conventional sub-Gaussian matrices, the following experimental setup is designed: Learned measurement matrices are obtained via $\mathbf{A} = \sum_{t=1}^{T} \mathbf{A}_1^{(t)} \otimes \mathbf{A}_2^{(t)}$, where $\mathbf{A}_1^{(t)}$ and $\mathbf{A}_2^{(t)}$ are left and right multiplication matrices for t^{th} Kronecker product. Then, compressive sensing of vectorized sparse signal, $\mathbf{x}$, is done, i.e., $\mathbf{y} = \mathbf{A}\mathbf{x}$. In order to recover recover sparse signal, ℓ_1-minimization is used: $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.t $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$. ADMM [66] is used to solve this optimization problem. Figure 7 shows an example visual comparison of the recovered outputs in this vector CS setup for conventional Gaussian CS matrix vs. learned CS matrix. Even coarse estimations of $\mathbf{x}$ via minimum norm solution and simple matrix vector multiplication with adjoint matrix (tied learning is used in this experiment, i.e., $\mathbf{B} = \mathbf{A}^{\mathbf{T}}$) are significantly improved compared to the ones from conventional Gaussian projection CS setup.

VI. Conclusion and Future Work

We propose generalized tensor summation networks for fast and high-quality CS. Our framework incorporates end-to-end learning where the parameters of both the CS matrix and the signal recovery are jointly optimized. On the sensing part, the CS matrices are modeled as the summation of T tensors, which has certain critical advantages. On one hand, the complexity and the number of parameters are greatly reduced thanks to the separability of tensors. By keeping $T = 1$, for instance, we can reduce the system to a Kronecker CS. On the other hand, unfactorized CS matrices can be approximated well enough by increasing T. In addition, we can design structured matrices by incorporating any separable basis into our framework, such as DCT.

The reconstruction step of the proposed algorithm takes advantage of an adjoint operator learned similarly in tensorial sum representation to perform a tentative reconstruction. In this proof of concept work, a non-iterative, CNN-based deep learning architecture is used as the refinement module. To further increase the performance, one can use the most advance deep backbone structures such as transformers or optimization-inspired deep neural networks in future works. We further note that increasing T improves the recovery until it reaches a maximum value at a certain T value (generally at $T = 3$ or 4). However, contrary to the expectation of staying in saturation, after this point, the performance metric starts decreasing with a further increase in T. Future works can be devoted to analyzing the reason for such behavior which may lead to better training strategies and increased performance.

References

[1] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Sep. 2004.

[2] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[3] A. C. Gurbuz, J. H. McClellan, and W. R. Scott, “A compressive sensing data acquisition and imaging method for stepped frequency GPRs,” *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2640–2650, Jul. 2009.

[4] M. Yamaç, M. Orhan, B. Sankur, A. S. Turk, and M. Gabbouj, “Through the wall target detection/monitoring from compressively sensed signals via structural sparsity,” in *Proc. 5th Int. Workshop Compressed Sens. Appl. Radar, Multimodal Sens., Imag.*, 2018. [Online]. Available: https://new.eurasip.org/Proceedings/Ext/CoSeRa2018/papers/p-yamac.pdf

[5] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, “Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.

[6] Y. Zhang, Y. Xiang, L. Y. Zhang, Y. Rong, and S. Guo, “Secure wireless communications based on compressive sensing: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1093–1111, 2nd Quart., 2019.

[7] M. Yamaç, M. Ahishali, N. Passalis, J. Raitoharju, B. Sankur, and M. Gabbouj, “Multi-level reversible data anonymization via compressive sensing and data hiding,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1014–1028, 2021.

[8] S. Chen and D. Donoho, “Basis pursuit,” in *Proc. 28th Asilomar Conf. Signals, Syst. Comput.*, 1994, pp. 41–44.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.

[10] M. S. Asif and J. Romberg, “On the LASSO and Dantzig selector equivalence,” in *Proc. 44th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2010, pp. 1–6.

[11] C. Olsson, M. Carlsson, and D. Gerosa, “Bias reduction in compressed sensing,” 2018, *arXiv:1812.11329*.

[12] C. Li, W. Yin, H. Jiang, and Y. Zhang, “An efficient augmented Lagrangian method with applications to total variation minimization,” *Comput. Optim. Appl.*, vol. 56, no. 3, pp. 507–530, Dec. 2013.

[13] C. A. Metzler, A. Maleki, and R. G. Baraniuk, “From denoising to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[14] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[15] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 367, no. 1906, pp. 4273–4293, Nov. 2009.

[16] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1336–1343.

[17] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 449–458.

[18] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1828–1837.

[19] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 375–388, 2020.

[20] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12282–12291.

[21] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 494–504, Feb. 2012.

[22] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[23] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 449–452, Jun. 2009.

[24] M. Impiö, M. Yamaç, and J. Raitoharju, "Multi-level reversible encryption for ECG signals using compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1005–1009.

[25] L. Y. Zhang, K.-W. Wong, Y. Zhang, and J. Zhou, "Bi-level protected compressive sampling," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1720–1732, Sep. 2016.

[26] M. Lustig, L. D. Donoho, M. Juan Santos, and M. J. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2007.

[27] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, "Breaking the coherence barrier: A new theory for compressed sensing," in *Forum of Mathematics, Sigma*, vol. 5. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[28] E. J. Candès, "Compressive sampling," in *Proc. Int. Congr. Math.*, vol. 3, Aug. 2006, pp. 1433–1452.

[29] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization for compressive sensing systems," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2887–2898, Jun. 2013.

[30] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.

[31] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 211–231, Jan. 2009.

[32] H. Rauhut, "Compressive sensing and structured random matrices," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9. Berlin, Germany: de Gruyter, 2010, pp. 1–92.

[33] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[34] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comp. Rendus Mathematique*, vol. 346, nos. 9–10, pp. 589–592, May 2008.

[35] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Statist.*, vol. 35, no. 6, pp. 2313–2351, Dec. 2007.

[36] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Jun. 2007.

[37] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.

[38] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.

[39] K. Schnass and P. Vandergheynst, "Average performance analysis for thresholding," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 828–831, Nov. 2007.

[40] I. Y. Chun and B. Adcock, "Compressed sensing and parallel acquisition," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4860–4882, Aug. 2017.

[41] S. K. Sahoo and A. Makur, "Signal recovery from random measurements via extended orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2572–2581, May 2015.

[42] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 139–154, Jan. 2012.

[43] A. Degerli, S. Aslan, M. Yamac, B. Sankur, and M. Gabbouj, "Compressively sensed image recognition," in *Proc. 7th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Nov. 2018, pp. 1–6.

[44] S. Lohit, K. Kulkarni, and P. Turaga, "Direct inference on compressive measurements using convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1913–1917.

[45] M. Yamac, M. Ahishali, S. Kiranyaz, and M. Gabbouj, "Convolutional sparse support estimator network (CSEN) from energy efficient support estimation to learning-aided compressive sensing," 2020, *arXiv:2003.00768*.

[46] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[47] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[48] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[49] H. Son and S. Lee, "Fast non-blind deconvolution via regularized residual networks with long/short skip-connections," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, May 2017, pp. 1–10.

[50] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 689–692.

[51] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008.

[52] D. Donoho, A. Maleki, and M. Shahram, "WaveLab 850," in *Software Toolkit for Time-Frequency Analysis*. Stanford, CA, USA: Stanford Univ. Press, 2006.

[53] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[54] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.* Cham, Switzerland: Springer, 2010, pp. 711–730.

[55] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. 23rd Brit. Mach. Vis. Conf. (BMVC)*. BMVA Press, 2012, pp. 135.1–135.10.

[56] J. Chen, Y. Sun, Q. Liu, and R. Huang, "Learning memory augmented cascading network for compressed sensing of images," in *Proc. ECCV*, 2020, pp. 513–529.

[57] Y. Sun, J. Chen, Q. Liu, B. Liu, and G. Guo, "Dual-path attention network for compressed sensing image reconstruction," *IEEE Trans. Image Process.*, vol. 29, pp. 9482–9495, 2020.

[58] J. Zhang, C. Zhao, and W. Gao, "Optimization-inspired compact deep compressive sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 4, pp. 765–774, May 2020.

[59] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, "AMP-Net: Denoising-based deep unfolding for compressive image sensing," *IEEE Trans. Image Process.*, vol. 30, pp. 1487–1500, 2021.

[60] D. You, J. Zhang, J. Xie, B. Chen, and S. Ma, "COAST: Controllable arbitrary-sampling network for compressive sensing," *IEEE Trans. Image Process.*, vol. 30, pp. 6066–6080, 2021.

[61] W. Cui, S. Liu, F. Jiang, and D. Zhao, "Image compressed sensing using non-local neural network," *IEEE Trans. Multimedia*, vol. 25, pp. 816–830, 2023.

[62] D. Ye, Z. Ni, H. Wang, J. Zhang, S. Wang, and S. Kwong, "CSformer: Bridging convolution and transformer for compressive sensing," 2021, *arXiv:2112.15299*.

[63] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

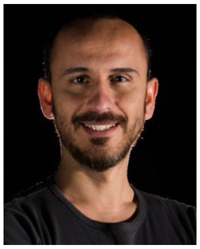
[64] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, “Manga109 dataset and creation of metadata,” in *Proc. 1st Int. Workshop coMics Anal., Process. Understand.*, Dec. 2016, pp. 1–5.

[65] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.

[66] S. Boyd, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

Mehmet Yamaç received the B.S. degree in electrical and electronics engineering from Anadolu University, Eskisehir, Turkey, in 2009, and the M.S. degree in electrical and electronics engineering from Bogaziçi University, Istanbul, Turkey, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computing Sciences, Tampere University, Tampere, Finland. He was a Research and Teaching Assistant with Bogazici University from 2012 to 2017 and a Researcher with Tampere University from 2017 to 2020. He has been a Senior Researcher with Huawei Technologies Oy, Tampere, from 2020 to 2023. He has coauthored more than 35 papers, two of them nominated for the “Best (or Student Best) Paper Award” in EUVIP 2018 and EUSIPCO 2019. His research interests are computer and machine vision, machine learning, and compressive sensing.

Ugur Akpinar received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University in 2015 and the M.Sc. degree in information technology from the Tampere University of Technology in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology and Communication Sciences, Tampere University. His research interests include multidimensional signal processing, optics, 3D imaging, and display technologies.



Erdem Sahin (Member, IEEE) received the Ph.D. degree from the Electrical and Electronics Engineering Department, Bilkent University, in 2013. In 2014, he joined the 3D Media Group, Faculty of Information Technology and Communication Sciences, Tampere University, as a Marie Curie Experienced Researcher, where he has been a Senior Research Fellow since 2019. His current research interests include development of computational light field and holographic imaging systems and algorithms.

Serkan Kiranyaz (Senior Member, IEEE) was born in Turkey in 1972. He received the B.S. and M.S. degrees from the Electrical and Electronics Department, Bilkent University, Ankara, Turkey, in 1994 and 1996, respectively, and the Ph.D. and Docency degrees from the Institute of Signal Processing, Tampere University of Technology, in 2005 and 2007, respectively.

He was a Professor with the Signal Processing Department, Tampere University of Technology, from 2009 to 2015. He is currently a Professor with Qatar University, Doha, Qatar. He has noteworthy expertise and background in various signal processing domains. He has published two books, seven book chapters, ten patents/applications, more than 100 journal articles in several IEEE TRANSACTIONS and other high impact journals, and more than 120 papers in international conferences. He served as a PI and a LPI in several national and international projects. His principal research field is machine learning and signal processing. He is rigorously aiming for reinventing the ways in novel signal processing paradigms, enriching it with new approaches especially in machine intelligence and revolutionizing the means of “learn-to-process” signals. He made significant contributions on bio-signal analysis, particularly EEG and ECG analysis and processing, classification and segmentation, computer vision with applications to recognition, classification, multimedia retrieval, evolving systems and evolutionary machine learning, swarm intelligence, and evolutionary optimization.

Moncef Gabbouj (Fellow, IEEE) is currently a Professor in information technology with the Department of Computing Sciences, Tampere University, Finland. He was an Academy of Finland Professor. His research interests include big data analytics, multimedia analysis, artificial intelligence, machine learning, pattern recognition, nonlinear signal processing, video processing, and coding. He is a fellow of the Asia–Pacific Artificial Intelligence Association. He is a member of the Academia Europaea, the Finnish Academy of Science and Letters, and the Finnish Academy of Engineering Sciences. He is also the Finland Site Director of the NSF IUCRC funded Center for Big Learning.