Deep Learning for Block-level Compressive Video Sensing

Yifei Pei, Ying Liu, and Nam Ling
Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA

***Abstract*—Compressed sensing (CS) is a signal processing framework that effectively recovers a signal from a small number of samples. Traditional compressed sensing algorithms, such as basis pursuit (BP) and orthogonal matching pursuit (OMP) have several drawbacks, such as low reconstruction performance at small compressed sensing rates and high time complexity. Recently, researchers focus on deep learning to get compressed sensing matrix and reconstruction operations collectively. However, they failed to consider sparsity in their neural networks to compressed sensing recovery; thus, the reconstruction performances are still unsatisfied. In this paper, we use 2D-discrete cosine transform and 2D-discrete wavelet transform to impose sparsity of recovered signals to deep learning in video frame compressed sensing. We find the reconstruction performance is significantly enhanced.**

***Index Terms*—compressed sensing, block-based compressed sensing, deep learning, discrete cosine transform, discrete wavelet transform, fully-connected neural network**

I. Introduction

Compressed sensing (CS) is a mathematical framework defining the conditions and tools to recover a sparse signal from a small number of linear projections [1]. The measuring instrument acquires the signal in the domain of the linear projection in the compressed sensing structure, and the complete signal is reconstructed using convex optimization methods. CS has a variety of applications including image acquisition [2], magnetic resonance imaging [3], and image compression [4].

This paper's primary contributions are: (1) For the first moment it introduces the use of discrete cosine transformed images and discrete wavelet transformed images in deep learning for compressed sensing tasks. (2) By combining discrete cosine transform or discrete wavelet transform with deep learning, we propose a neural network architecture to achieve stronger reconstruction quality of compressed sensed video frames.

The rest of the paper is organized as follows. In section 2, we briefly review the concept of compressed sensing, the motivation of block-based image compression, and the state-of-the-art deep learning method for compressed sensing. In section 3, we introduce the proposed method. Section 4 presents the experiment results on the six datasets that support our algorithm developments. Finally, we conclude the paper and discuss future research directions.

II. Compressed Sensing Background

A. Compressed Sensing

Compressed sensing theory shows that an S-sparse signal $\mathbf{x} \in \mathbb{R}^{\mathbb{N}}$ is able to be compressed into a measurement vector $\mathbf{y} \in \mathbb{R}^{\mathbb{M}}$ by an over-complete matrix $\mathbf{A} \in \mathbb{R}^{\mathbb{M}\times\mathbb{N}}$, $M \ll N$ [1] and can be recovered if $\mathbf{A}$ satisfies the restricted isometry property (RIP). However, in images, pixels are not sparse. Thus, to recover $\mathbf{x}$ from the measurement $\mathbf{y}$, a certain transform (such as the discrete cosine transform or the discrete wavelet transform) is needed, so that $\mathbf{x}$ can be sparsely represented in the transform domain, that is, $\mathbf{x} = \mathbf{\Psi s}$, where $\mathbf{s}$ is the sparse transform coefficient vector [5]. The recovery of $\mathbf{x}$ is equivalent to solving the l_1-norm based convex optimization problem [6]:

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{\Phi\Psi s}. \end{aligned} \tag{1}$$

While basis pursuit can be efficiently implemented with linear programming to solve the above minimization problem, its computational complexity is often high, hence people resort to greedy techniques such as orthogonal matching pursuit [7] to reduce the computational complexity.

B. Compressed Sensing with Deep Learning

Deep neural networks offer another way to perform compressive image sensing [8]. The benefit of such a strategy is that during training, the sensing matrix and nonlinear reconstruction operators can be jointly optimized, thus outperforming other existing CS algorithms for compressed-sensed images in terms of reconstruction accuracy and less reconstruction time. However, such reconstruction remains unsatisfying, particularly at very small sampling rates. At a large compressive sampling rate, the reconstruction ability tends to reach the upper limit due to the overfitting issue. Furthermore, this neural network is for images, not for videos. [9] develops a 6-layer convolutional neural network (32 or 64 feature maps in four convolutional layers) to reconstruct images from compressive sensing image signals. [10] uses a generative neural network to reconstruct compressive sensing MRI images. The neural network architecture consists of an 8-layer convolutional neural network (128 feature maps in each convolutional layer) with a ResNet for the generator and a 7-layer convolutional neural network (feature maps double from 8 to 64 in the first four layers and keep 64 until the last convolutional layer) for the discriminator. However, deep convolutional neural networks incur high computational complexity during training and the hyperparameters (e.g., depths and dimensions of feature maps) must be carefully tuned for specific datasets. Meanwhile, current deep learning strategies for compressive sensing seldom take the sparsity of

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

original signals into consideration as traditional compressive sensing methods do.

C. Block-based Compressed Sensing

In block-based compressed sensing (BCS), an image is split into small blocks of size $B \times B$ and compressed with a measuring matrix $\mathbf{\Phi}_B$ [11]. Assume that $\mathbf{X_i} \in \mathbb{R}^{\mathbb{B}\times\mathbb{B}}$ is an image block and the vectorized block is $\mathbf{x_i} \in \mathbb{R}^{\mathbb{B}^{\mathbb{2}}}$, where i is the block index. The corresponding CS measurement vector is $\mathbf{y_i} = \mathbf{\Phi_B x_i}$, where $\mathbf{\Phi}_B \in \mathbb{R}^{\lambda\times\mathbb{B}^{\mathbb{2}}}$ and $\lambda = \lfloor RB^2 \rfloor$($R$ is the sensing rate, $R \ll 1$). The use of BCS instead of sampling the whole image has several advantages:

1) Due to the small block size, the CS measurement vectors are conveniently collected and used;
2) The encoder does not have to wait until the whole image is compressed, instead, it can send the CS measurement vector of each block to the decoder after it is acquired;
3) Due to the small size of $\mathbf{\Phi}_B$, the memory is saved.

III. The Proposed Approach

A. Fully-connected Neural Network

In this paper, we propose a deep learning framework. The reason to choose this neural network is that it has a very simple structure, high computational efficiency, and it outputs high-quality reconstructed video frames. The architecture of the neural network (Fig. 1) consists of:

1) an input layer with B^2 nodes (frame block receptor);
2) a forward transform layer with B^2 nodes (forward transform operation);
3) a flatten layer with B^2 nodes (vectorization);
4) a compressed sensing layer with B^2R nodes, $R \ll 1$ (linear compressed sensing);
5) an expansion layer with B^2T nodes, each followed by the ReLU activation function, where $T > 1$ is the expansion factor;
6) a reconstruction layer of B^2 nodes (shape controller);
7) a reshape layer of B^2 nodes (vector to matrix conversion);
8) an inverse transform layer of B^2 nodes (inverse transform operation).

B. 2D-Discrete Cosine Transform and 2D-Discrete Wavelet Transform

We use discrete cosine transform (DCT) or discrete wavelet transform (DWT) to perform transformation on our image blocks to project them on to the sparse domain.

We use 2D-DCT and 2D-DWT. We denote the $B \times B$ transform matrix as $\mathbf{C}$. For 2D-transform, we use $\mathbf{CX_iC^T}$ to transform image block $\mathbf{X_i}$ to the frequency-domain sparse signal $\mathbf{S_i}$ and vectorize it as $\mathbf{s_i}$. For 2D-inverse-transform, we use $\mathbf{C^T S_i C}$ to transform $\mathbf{S_i}$ to the original image block $\mathbf{X_i}$ and denote the corresponding block vector as $\mathbf{x_i}$. Our algorithm jointly optimizes the sensing matrix $\mathbf{\Phi_B}$ and the non-linear reconstruction operator:

$$\hat{\mathbf{s}}_\mathbf{i} = \mathbf{W_2}(\mathrm{ReLU}(\mathbf{W_1}(\mathbf{\Phi_B s_i}))). \tag{2}$$

which is parameterized by coefficients matrices $\mathbf{W_1}$ and $\mathbf{W_2}$ with an activation function ReLU.

We minimize the mean-squared-error (MSE) loss function in the training process:

$$\underset{\mathbf{\Phi}_B,\mathbf{W_1},\mathbf{W_2}}{\text{minimize}} \quad E\{\|\hat{\mathbf{s}}_\mathbf{i} - \mathbf{s_i}\|^\mathbf{2}\}. \tag{3}$$

IV. Experiment Results

This section provides experimental details and the performance evaluation of the proposed neural network. We use the Foreman and the Container datasets (SIF format). Each dataset has 300 frames and each frame is of dimension $352 \times 288 \times 1$. To simplify our experiment, we only use the luminance component of each dataset. We divide our images into $B \times B$ blocks. In our experiment, we set $B = 16$. Instead of using the AdaGrad optimization algorithm [8], as we find in practice, that has the local minima problem as the learning rates vanish, we use the Adam optimization algorithm in the training process to achieve fast convergence speed and to overcome the local minima issue [13]. In our experiments, we find 150 epochs suitable for our training process in most cases.

We evaluate the reconstruction performance by the peak signal-to-noise ratios (PSNRs) with 3 expansion factor values (T = 8, 10, and 12) in FCN+DCT, FCN+DWT, and FCN at 7 compressed sensing ratios (R = 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, and 0.40). PSNR is calculated through mean-squared-error (MSE) by (4). The MSE is defined as $E\{\|\hat{\mathbf{x}}_\mathbf{i} - \mathbf{x_i}\|\}$. The maximum pixel intensity value (MAX) is 255. We compare the reconstruction PSNR values and the total processing time of our proposed compressed sensing deep learning algorithms with those of the traditional algorithms, such as basis pursuit (BP), orthogonal matching pursuit (OMP) and total variation minimization [12]. Deep learning algorithms are implemented with Python by using Keras 2.3.0 and accelerated by NVIDIA RTX 2080 Ti GPU. Orthogonal matching pursuit, basis pursuit and total-variation minimization are implemented with Matlab. Gaussian sensing matrices with random entries of 0 mean and standard deviation $\sqrt{M}$ are used to compress the original image blocks (M is the length of the CS measurement vectors) in orthogonal matching pursuit and basis pursuit. Random partial Walsh Hadamard matrix is used to compress the original image block in total-variation minimization.

$$\text{PSNR} = 10 \cdot \log_{10} \frac{\mathbf{MAX^2}}{\mathbf{MSE}} \text{ (dB)}. \tag{4}$$

TABLE I and TABLE II show that our proposed FCN+DCT and FCN+DWT perform better than the pure FCN and traditional compressed sensing recovery algorithms such as the basis pursuit (BP), orthogonal matching pursuit(OMP), and total-variation minimization (TV) in terms of the reconstruction quality at 7 sensing rates. Further, FCN+DCT outperforms FCN+DWT. For each testing dataset, we calculate the average reconstruction PSNR values of each deep learning method across testing frames of each expansion factor value. For the Foreman dataset, the proposed FCN+DWT improves the

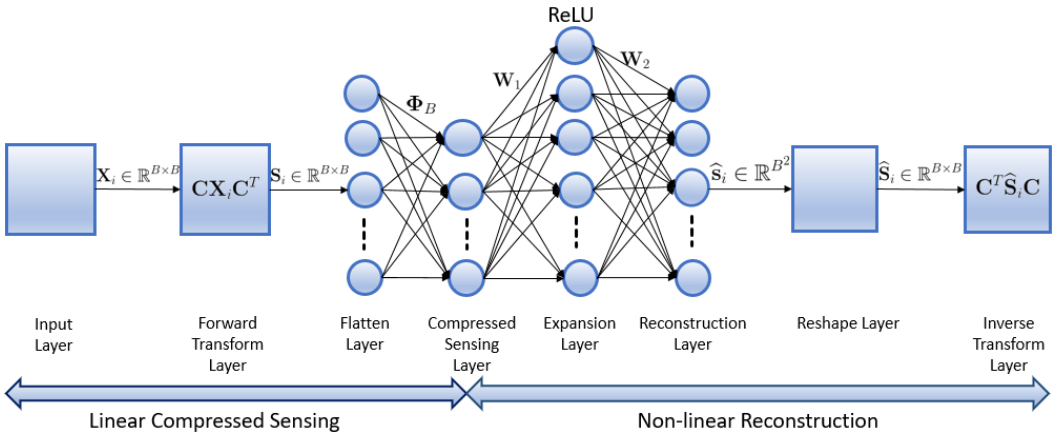

Fig. 1: Fully-connected neural network for compressed sensing.

TABLE I: The average reconstruction PSNR [dB] versus the sensing rate ($R = M/N$) of the Foreman dataset.

Method	$R = 0.10$	$R = 0.15$	$R = 0.20$	$R = 0.25$	$R = 0.30$	$R = 0.35$	$R = 0.40$
FCN+DCT ($T = 8$)	31.63	32.87	33.87	34.65	35.65	36.34	37.12
FCN+DCT ($T = 10$)	31.55	32.85	34.90	35.52	36.13	36.13	37.33
FCN+DCT ($T = 12$)	31.67	32.80	34.01	34.97	35.70	36.44	37.31
FCN+DWT ($T = 8$)	31.50	32.84	33.79	34.72	35.53	36.10	36.67
FCN+DWT ($T = 10$)	31.49	32.85	33.84	34.61	35.27	36.06	37.11
FCN+DWT ($T = 12$)	31.57	32.84	33.81	34.66	35.49	36.42	36.49
FCN ($T = 8$)	31.22	32.56	33.28	34.18	35.15	35.62	36.65
FCN ($T = 10$)	31.29	32.66	33.35	34.22	35.13	35.75	35.81
FCN ($T = 12$)	31.00	32.39	33.53	34.24	35.02	35.79	36.00
OMP	19.08	20.64	21.85	23.67	24.07	25.11	25.78
BP	20.08	21.60	23.94	25.28	26.55	27.74	28.73
TV	23.91	25.43	27.56	28.83	30.25	31.40	32.24

TABLE II: The average reconstruction PSNR [dB] versus the sensing rate ($R = M/N$) of the Container dataset.

Method	$R = 0.10$	$R = 0.15$	$R = 0.20$	$R = 0.25$	$R = 0.30$	$R = 0.35$	$R = 0.40$
FCN+DCT ($T = 8$)	34.15	35.43	36.56	37.58	38.20	38.87	39.73
FCN+DCT ($T = 10$)	34.20	35.73	36.31	37.23	38.33	39.15	40.27
FCN+DCT ($T = 12$)	34.48	35.64	36.91	37.14	38.44	39.30	39.64
FCN+DWT ($T = 8$)	33.81	35.50	36.33	36.86	37.10	38.06	38.82
FCN+DWT ($T = 10$)	33.92	35.31	36.20	37.05	37.86	37.95	38.83
FCN+DWT ($T = 12$)	34.06	35.29	36.47	37.38	37.51	38.15	38.29
FCN ($T = 8$)	33.70	35.02	35.71	36.24	36.90	37.94	38.78
FCN ($T = 10$)	33.63	34.86	35.36	36.61	36.83	37.94	38.00
FCN ($T = 12$)	34.00	34.74	34.94	36.74	36.75	37.57	38.03
OMP	17.47	18.32	19.22	20.18	20.98	21.77	22.49
BP	18.78	20.19	21.56	22.73	23.73	24.66	25.56
TV	22.33	23.21	24.44	25.47	26.49	27.44	28.28

PSNR of FCN by 0.35 dB for low sensing rate ($R = 0.10$) and 0.60 dB for high sensing rate ($R = 0.40$). FCN+DCT further improves these results by 0.10 dB and 0.50 dB. For the Container dataset, the FCN+DWT improves the PSNR of FCN by 0.15 dB and 0.38 dB for low sensing rate ($R = 0.10$) and high sensing rate ($R = 0.40$), respectively. The FCN+DCT further improves these results by 0.35 dB and 1.24 dB. We also observe that neural network for compressed sensing signal recovery performs better in the Container dataset than in the Foreman dataset. It is because the motion in the Foreman dataset is faster than that in the Container dataset. Figs. 2-3 demonstrate the visual quality improvements by the FCN+DCT and the FCN+DWT compared to the FCN on two testing images at two sensing rates. In Fig.4, we analyze the validation loss in 150 epochs. We find the FCN+DCT and the FCN+DWT smooth the validation loss curves compared to the validation loss curve of the pure FCN. In particular, the FCN+DCT smoothes the validation loss curve better as compared to the FCN+DWT. We also use another four CIF format datasets (Monitor Hall, News, Akiyo and Silent) to train and test the neural network models. Each dataset has 300 frames and each frame has a dimension size of $352 \times 288 \times 1$. We use the same method as the one used for the Foreman and Container datasets to train neural network models except that we set the training epochs for Akiyo to be 25 instead of 150 because overtraining issues occur after 25 epochs of training [14]. The results are shown in TABLE III, indicating that the proposed FCN+DCT achieves higher quality for the recovered video frames compared to FCN+DWT and FCN. TABLE IV shows the total processing time (DCT/DWT transform

Fig. 2: Foreman for $M/N = 0.4$. Left to right: original; FCN+DCT ($T = 10$), PSNR = 38.44 dB; FCN+DWT ($T = 10$), PSNR = 38.21 dB; FCN ($T = 10$), PSNR = 36.73 dB; OMP, PSNR = 26.50 dB; BP, PSNR = 29.15 dB; TV, PSNR = 32.90 dB.

Fig. 3: Container for $M/N = 0.2$. Left to right: original; FCN+DCT ($T = 10$), PSNR = 36.97 dB; FCN+DWT ($T = 10$), PSNR = 36.90 dB; FCN ($T = 10$), PSNR = 35.83 dB; OMP, PSNR =19.47 dB; BP, PSNR = 21.48 dB; TV, PSNR = 24.41 dB.

TABLE III: The Average reconstruction PSNR [dB] versus the sensing rate ($R = M/N$) of other datasets by neural networks ($T = 10$).

Dataset	$R = 0.10$			$R = 0.25$			$R = 0.40$		
	FCN+DCT	FCN+DWT	FCN	FCN+DCT	FCN+DWT	FCN	FCN+DCT	FCN+DWT	FCN
Monitor Hall	33.94	33.76	33.50	38.26	37.89	37.71	41.82	41.19	41.09
News	32.02	32.01	31.57	36.22	35.77	34.74	40.02	38.97	38.90
Akiyo	34.41	34.25	33.61	38.07	37.10	36.83	40.04	39.21	39.11
Silent	34.59	34.17	33.78	38.59	37.93	36.39	41.28	40.52	38.38

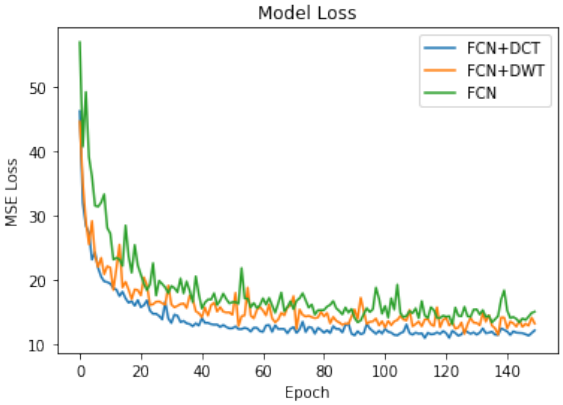

Fig. 4: Validation loss of Container for $M/N = 0.25$ ($T = 10$).

TABLE IV: Total processing time at $R = 0.20$ for 90 testing images (352 $\times$288).

Method	Time [seconds]
FCN+DCT ($T = 8$)	4.90
FCN+DCT ($T = 10$)	5.12
FCN+DCT ($T = 12$)	5.38
FCN+DWT ($T = 8$)	4.80
FCN+DWT ($T = 10$)	5.01
FCN+DWT ($T = 12$)	5.24
FCN ($T = 8$)	4.13
FCN ($T = 10$)	4.53
FCN ($T = 12$)	4.99
OMP	642.53
BP	543.86
TV	2717.13

time, compressed sensing time, recovery time and DCT/DWT inverse-transform time) for 90 testing images. The DCT/DWT slightly increases the processing time compared to the pure FCN, but the overall methods are approximately 542 times faster than total-variation minimization.

V. Conclusions

This paper proposed a deep learning framework that utilizes the sparse property of images to enhance the reconstruction quality of compressed-sensed video frames through a fully-connected neural network. This paper demonstrated that sparse transforms such as DCT and DWT, which are widely used in traditional compressed sensing recovery algorithms, can also be applied to neural networks to recover compressed-sensed video frames. However, performance improvement differs in 2D-DCT and 2D-DWT, where 2D-DCT outperforms 2D-DWT in the fully-connected neural network reconstruction of compressed-sensed images. The future research will focus on the mathematical explanations of sparse transform in deep learning for compressed sensing recovery and use new activation functions to move the study forward [15].

REFERENCES

[1] D. L. Donoho, "Compressed sensing," in IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006.

[2] S. Rouabah, M. Ouarzeddine and B. Souissi, "SAR Images Compressed Sensing Based on Recovery Algorithms," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, 2018, pp. 8897-8900.

[3] D. Lee, J. Yoo and J. C. Ye, "Deep residual learning for compressed sensing MRI," 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, 2017, pp. 15-18.

[4] J. Li, Y. Fu, G. Li and Z. Liu, "Remote Sensing Image Compression in Visible/Near-Infrared Range Using Heterogeneous Compressive Sensing," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 12, pp. 4932-4938, Dec. 2018.

[5] R. G. Baraniuk, "Compressive Sensing [Lecture Notes]," in IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 118-121, July 2007.

[6] Shaobing Chen and D. Donoho, "Basis pursuit," Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1994, pp. 41-44 vol.1.

[7] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," in IEEE Transactions on Information Theory, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[8] A. Adler, D. Boublil and M. Zibulevsky, "Block-based compressed sensing of images via deep learning," 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, 2017, pp. 1-6.

[9] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche and A. Ashok, "ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 449-458.

[10] M. Mardani et al., "Deep Generative Adversarial Neural Networks for Compressive Sensing MRI," in IEEE Transactions on Medical Imaging, vol. 38, no. 1, pp. 167-179, Jan. 2019.

[11] Lu Gan, "Block Compressed Sensing of Natural Images," 2007 15th International Conference on Digital Signal Processing, Cardiff, 2007, pp. 403-406.

[12] J. Romberg, "Imaging via Compressive Sampling," in IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 14-20, March 2008.

[13] D. P. Kingma and J. Ba, "Adam : A method for stochastic optimization," arXiv:1412.6980 [cs], Dec. 2014.

[14] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks," 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, 2017, pp. 173-177.

[15] L. Xiao, H. Wang and N. Ling, "Image Compression with Deeper Learned Transformer," Proceedings of the APSIPA Annual Summit and Conference 2019, pp.53-57, Lanzhou, China, Nov 18-21, 2019.