Class-Specific Neural Network for Video Compressed Sensing

Yifei Pei[1], Ying Liu[1], Nam Ling[1], Lingzhi Liu[2], Yongxiong Ren[2]
[1]Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA
[2]Heterogenous Computing Group, Kwai Inc., Palo Alto, CA, USA

2021 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-7281-9201-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISCAS51556.2021.9401450

***Abstract*—Compressed sensing is an effective solution for signal acquisition and signal reconstruction at a much lower rate than the Nyquist rate. Traditional methods, such as orthogonal matching pursuit and basis pursuit, for image compressed sensing reconstruction have unsatisfying reconstruction quality and long reconstruction time. Researchers now focus on neural network and deep learning methods for the better reconstruction of compressed-sensed signals at a very low sampling rate and a fast speed. However, current neural network approaches for image compressed sensing do not consider the similarities between images or within images, or the types of image blocks; thus, performing poorly in images with complex contents. In this paper, we develop a novel neural network framework that utilizes the similarities between image blocks through Gaussian-mixture models without recording the similarity information to achieve better reconstruction quality than the state-of-the-art neural network methods for block-level image compressed sensing.**

Keywords—block-based compressed sensing, compressed sensing, discrete cosine transform, Gaussian-mixture model, logistic regression classifier, neural network.

I. Introduction

The Shannon-Nyquist sampling theory indicates that to recover a signal accurately, a signal needs to be sampled at least twice the highest frequency present in the signals [1], resulting in large samples with redundant information. Compressed sensing is proposed to achieve sampling and compression steps at one time [2]. Traditional methods, such as orthogonal matching pursuits [3] and basis pursuit [4], for image compressed sensing reconstruction requires huge reconstruction time, and bring low qualities of reconstructed images. Researchers now focus on deep learning methods to achieve high reconstruction quality of images at a fast speed [5]. However, deep learning for image compressed sensing has not yet considered the similarities between images or the contents within images; thus, poorly performing as the images become complex. One way to know the types of images or image blocks is through the Gaussian-mixture model [6]. However, if we tailor neural networks to different classes of images or image blocks, we need to record the class label information to inform the decoder, which requires extra bits. In this work, we develop a neural network architecture utilizing the class labels of video frame blocks for image compressed sensing without recording the class label information. Our contributions are:

- We use Gaussian-mixture models to classify video frame blocks for optimizing our end-to-end neural network architectures with different clusters.
- We use a logistic regression classifier to provide class labels of compressed-sensed video frame block vectors for decoder without recording extra clustering information.
- We design a hashmap data structure to accelerate compressed sensing and reconstruction speed significantly.

II. Background

A. Compressed Sensing

Compressed sensing theory shows that an *S*-sparse signal $\mathbf{x} \in \mathbb{R}^N$ is able to be compressed into a sampled vector $\boldsymbol{y} \in \mathbb{R}^M$ by a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$, $M \ll N$ and can be recovered if $\boldsymbol{\Phi}$ satisfies the restricted isometry property (RIP) of the order of 2k [7], that is:

$$(1-\delta_{2k})||\mathbf{x}||_{l_2}^2 \leq ||\boldsymbol{\Phi}\mathbf{x}|| \leq (1+\delta_{2k})||\mathbf{x}||_{l_2}^2, \quad (1)$$

where δ_{2k} is the isometry constant. However, in images, pixels are not sparse, and some transforms are needed to represent $\mathbf{x}$ in sparse frequency $\mathbf{s}$ through $\mathbf{x} = \boldsymbol{\psi}\mathbf{s}$. The recovery of $\mathbf{x}$ is equivalent to solving the l_1-norm based convex optimization problem:

$$\hat{\mathbf{s}} = argmin_s||\mathbf{s}||_1 \; s.t. \; \boldsymbol{y} = \boldsymbol{\phi}\boldsymbol{\psi}\mathbf{s}. \quad (2)$$

B. Compressed Sensing with Neural Network

The basic idea of neural network for image compressed sensing is to train a measurement matrix $\boldsymbol{\Phi}$ with the inverse transform matrix $\mathbf{W}$ collaboratively, so that, $\hat{\mathbf{x}} = \mathbf{W}(\boldsymbol{\Phi}\mathbf{x})$. A method has been proposed in [5] where the researchers use *n* fully-connected neural network layers with weights $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, …, $\mathbf{W}^{(n)}$ of these layers, and the non-linear activation function, *Relu*, in a layer with more nodes for reconstruction. It achieved better reconstruction results than traditional methods and with a fast speed. One most recent research [8] applied the sparse transform matrix, such as discrete cosine transform (DCT) matrix and the inverse sparse transform matrix to video frames such that the neural network utilizes the properties of compressed sensing, that is, $\mathbf{s}$ must be sparse so to achieve satisfying results. However, the models are relatively simple and cannot perform well if images become complex. Both [5] and

This research was supported in part by Kwai Inc., USA under Kwai, Inc. grant.

978-1-7281-9201-7/21/$31.00 ©2021 IEEE

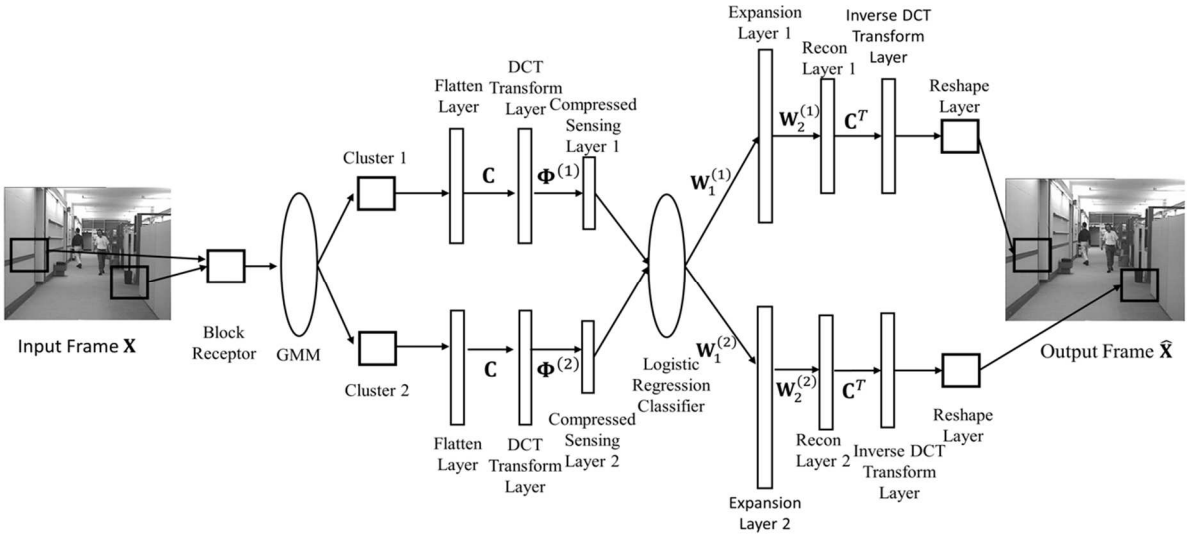

Fig.1 Example of class-specific neural network framework for two clusters.

[8] are performed at block-level. Block-level compressed sensing brings several benefits: (1) it saves the space of transform matrices given that transform matrices are applied to image blocks with small magnitudes rather than the whole images with hundreds of times of magnitudes; (2) it solves the issues when training data are not sufficient as splitting the whole images to blocks increases the number of data points. Some other methods applied to whole images use deep convolutional neural networks, which typically have a few convolutional layers with hundreds of feature maps in each convolutional layer [9], [10]. These approaches require large amount of training data, and the complicated tuning of network hyperparameters.

III. Our Approach

We design a class-specific neural network framework (Fig.1). A trained Gaussian-mixture model predicts the blocks labels and sends the blocks to their belonged encoders to get the compressive-sensed vectors. A logistic regression classifier predicts the labels of compressive-sensed vectors and sends these vectors to their belonged decoders to complete the block reconstruction.

A. Gaussian-mixture Model for Block Classification

The Gaussian-mixture model (GMM) assumes that each data point from class k is generated by a mixture of K multivariate Gaussian distributions with the model parameter $\boldsymbol{\theta}$ of a weight π_k, a mean vector $\boldsymbol{\mu}_k$ and a covariance matrix $\boldsymbol{\Sigma}_k$ for each cluster $k = 1{:}K$ [11]. The GMM has constraints of $\sum_{k=1}^{K} \pi_k = 1$ and $0 \le \pi_k \le 1$. The probability density of the i^{th} vectorized frame block $\mathbf{x}_i$ under the class k is expressed as:

$$\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{1/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right). \quad (3)$$

The linear super-position of Gaussians is:

$$p(\mathbf{x_i}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (4)$$

We denote the predicted class label of $\mathbf{x_i}$ as z_i. We compute z_i using the maximum a posteriori probability (MAP) estimate [11]:

$$z_i = \underset{k}{argmax}\ log\ p\,(\mathbf{x}_i|z_i = k, \boldsymbol{\theta}) + log\ p\,(z_i = k|\boldsymbol{\theta}). \quad (5)$$

B. Class-Specific Neural Network for Block Compressed Sensing

We denote the sampling ratio as R ($R = M/N$). The main parts of the proposed class-specific neural network consist of:

1. an input layer with B^2 nodes;
2. a flatten layer;
3. a DCT transform layer with B^2 nodes using fixed parameters, that is, Kronecker product form of sparse transform matrix $\mathbf{C} = \mathbf{D} \otimes \mathbf{D}^T$ applied to block vectors, where $\mathbf{D}$ is the discrete cosine transform (DCT) matrix applied to block matrices [12];
4. a trainable compressed sensing layer with B^2R nodes, $R \ll 1$;
5. an expansion layer with B^2T nodes, each followed by *Relu* activation function, $T \gg 1$;
6. a trainable reconstruction layer with B^2 nodes;
7. an inverse DCT transform layer with B^2 nodes using fixed parameters, $\mathbf{C}^T$;
8. a reshape layer to convert predicted block vectors to predicted block matrices.

For K classes, we train K neural networks. For the k^{th} class, we denote the block of size $B \times B$ as $\mathbf{X}_i^{(k)}$ and the corresponding neural network operation as $f^{(k)}(\cdot)$. We update

the parameters for $\mathbf{\Phi}^{(k)}$, $\mathbf{W}_1^{(k)}$, and $\mathbf{W}_2^{(k)}$ by minimizing the mean-squared-error (MSE):

$$\mathbf{\Phi}^{(k)}, \mathbf{W}_1^{(k)}, \mathbf{W}_2^{(k)} = argmin\, E||f^{(k)}(\mathbf{X}_i{}^{(k)}) - \mathbf{X}_i{}^{(k)}||_F^2. \quad (6)$$

C. Logistic Regression for the Classification of Compressed-Sensed Vector

Without recording the extra information for class labels of compressed-sensed vectors, we use a trained logistic regression to predict the class labels of compressed-sensed vectors.

For the i^{th}compressed-sensed vector $\mathbf{y}_i \in \mathbb{R}^{\mathbf{M}\times\mathbf{1}}$, we denote the class label as l_i and the parameter vector of the trained logistic regression for the k^{th} class as $\mathbf{w}_k\,(k = 1{:}\,K)$. The probability of the compressed-sensed vector belonged to class k is:

$$p(l_i = k|\mathbf{y}_i) = exp(\mathbf{w}_k^T\mathbf{y}_i)/\, exp\left(\sum_{k'=1}^{K} \mathbf{w}_{k'}^T\mathbf{y}_i\right), \quad (7)$$

where exp(·) is the exponential function [13]. We maximize $p(l_i = k|\mathbf{y}_i)$ to get the predicted label l_i of $\mathbf{y}_i$.

D. Hashmap Data Structure to Accelerate Compressed Sensing Speed and Reconstruction Speed

For a compressed-sensed vector **y**, the decoder needs to assign it to the corresponding k^{th} decoder based on the label predicted by logistic regression, which takes time to switch between decoders. Hence, we use a hashmap data structure so that we can decode a group of **y** vectors all at once. We group all compressed-sensed vectors into a matrix $\mathbf{Y} \in \mathbb{R}^{n\times B^2R}$, where n is the number of compressed-sensed vectors and B^2R is the length of these vectors. The k^{th} decoder is denoted as $F^{(k)}(\cdot)$, where $k = 1{:}\,K$. The labels of the compressed-sensed vector elements of **Y** predicted by the trained logistic regression are saved in an 1D array **l**. According to the class labels in **l**, we extract the submatrix $\mathbf{Y}^{(k)}$, which is formed by the compressed-sensed vectors from class k, k = 1:*K*. We decode the blocks from class k all-together through $\widehat{\mathbf{X}}^{(k)} = F^{(k)}(\mathbf{Y}^{(k)})$ for $k = 1{:}\,K$, where $\widehat{\mathbf{X}}^{(k)} \in \mathbb{R}^{n_k\times B\times B}$ is the decoded blocks from class k**,** and n_k is the number of blocks from class k. In this way, the reconstruction speed is significantly accelerated.

IV. Experimental Results

We evaluate the performance of our proposed Gaussian-awareness neural network using four CIF format video sequences. Each CIF video sequence has 300 frames and each frame has the size of $352 \times 288 \times 3$. To simplify our experiments, we only use luminance channel, resulting in each frame of size of $352 \times 288 \times 1$. For each dataset, we randomly select 150 frames as the training dataset, 60 frames as the validation dataset, and 90 frames as the testing dataset. We split the our video sequences in training dataset and validation dataset into 16×16 overlapped blocks with a split step of 8. For the testing dataset, we split the frames into 16×16 nonoverlapped blocks. We vectorize all blocks in training and testing sets to train the Gaussian-mixture model as we assume we have them. For the neural network training, we set the epoch as 150, the batch size as 64, and use the Adam optimizer [14]. For each training epoch, we use our validation dataset to do validation and save the best model till that training epoch. We compare our model with the block-based CSNet [5] and SparseNet [8] at sampling rates (*R=M/N*) of 0.05, 0.10, 0.15, 0.20 and 0.25. We use peak signal-to-noise ratio (PSNR) as the quality evaluation metric. It is defined as $\text{PSNR} = 20 \cdot \log_{10}(\text{MAX}) - 10 \cdot \log_{10}(\text{MSE})$ [dB], where MAX is the maximum pixel intensity 255, and MSE is mean-squared-error between original frame pixel intensity and reconstructed frame pixel intensity.

TABLE I shows the total reconstruction time (without GPU) for 90 testing frames of Hall Monitor. Although our approach is slower than CSNet and SparseNet because our approach needs to complete the reconstruction of one class of frame blocks before reconstructing another class of frame blocks using one processor, it has higher PSNR than the CSNet. Note that the reconstruction procedures for different classes of frame blocks can be paralleled to reduce the total reconstruction time.

Fig. 2 and Fig. 3 show the visual quality of selected reconstructed frame of Hall Monitor and Akiyo video sequences at the sampling rates at 0.10 and 0.20 under different methods. Our approach with *k*=6 achieves the best visual quality.

TABLE II shows the average reconstruction PSNR values in the four video sequences at four sampling rates. As we increase cluster number *k*, PSNR is improved by 1.5 dB in Hall monitor, 0.5 dB in Foreman, 2.5 dB in Container, and 2.0 dB in Akiyo, compared to SparseNet.

TABLE I: The total reconstruction time at R=0.10 for 90 testing frames (352×288) of Hall Monitor.

Method	Time [seconds]
CSNet	7.79
SparseNet	9.22
GMMNet(*k*=2)	12.69
GMMNet(*k*=3)	15.62
GMMNet(*k*=4)	19.44
GMMNet(*k*=5)	23.13
GMMNet(*k*=6)	26.37

V. Conclustions

We propose a block-level deep learning compressed sensing framework that utilizes the types of frame blocks predicted by the Gaussian-mixture model. The model achieves enhanced reconstruction quality for video frames but still at a fast speed. Our contribution to this paper will help future researchers to consider the similarities between blocks in their neural network design for compressed sensing and even for some general image processing purposes. Future research will be focused on using complex convolutional neural networks to improve the reconstruction quality of video sequences and considering other visual tasks [15].

Fig. 2 PSNR for Hall Monitor under R=0.10: Left to right: original; CSNet, 34.57 dB; SparseNet, 35.17 dB; Ours (k=2), 35.18 dB; Ours (k=6), 37.16 dB.

Fig. 3 PSNR for Akiyo under R=0.20: Left to right: original; CSNet, 38.63 dB; SparseNet, 39.31 dB; Ours (k=2), 39.93 dB; Ours (k=6), 41.44 dB.

TABLE II: The average reconstruction PSNR [dB] versus the sampling rate (R=M/N) of testing datasets.

Dataset	Method	R=0.05	R=0.10	R=0.15	R=0.20	R=0.25
Hall monitor	CSNet	31.20	34.37	35.99	37.34	38.25
	SparseNet	31.98	34.89	36.56	37.92	39.07
	GMMNet(k=2)	32.54	35.15	36.82	38.21	39.36
	GMMNet(k=3)	32.36	35.71	37.34	38.68	39.92
	GMMNet(k=4)	33.19	35.86	37.52	38.81	40.07
	GMMNet(k=5)	33.52	36.15	37.73	39.02	40.25
	GMMNet(k=6)	33.75	36.26	37.80	39.06	40.25
Foreman	CSNet	29.11	31.23	32.59	32.96	34.16
	SparseNet	29.21	31.46	32.78	33.86	34.78
	GMMNet(k=2)	29.35	31.58	32.91	33.99	34.98
	GMMNet(k=3)	29.48	31.66	32.97	34.03	34.98
	GMMNet(k=4)	29.58	31.70	32.94	34.09	35.08
	GMMNet(k=5)	29.74	31.82	33.10	34.18	35.16
	GMMNet(k=6)	29.81	31.84	33.13	34.22	35.15
Container	CSNet	28.74	30.85	31.43	33.46	34.16
	SparseNet	29.21	31.46	32.78	33.86	34.78
	GMMNet(k=2)	30.04	32.68	34.35	35.65	36.93
	GMMNet(k=3)	30.49	32.93	34.47	35.85	37.04
	GMMNet(k=4)	30.98	33.42	34.95	36.31	37.46
	GMMNet(k=5)	31.26	33.59	35.05	36.35	37.55
	GMMNet(k=6)	31.33	33.64	35.38	36.43	37.63
Akiyo	CSNet	34.35	37.43	39.03	40.04	40.98
	SparseNet	34.64	37.83	39.59	40.96	41.83
	GMMNet(k=2)	35.64	38.74	40.47	41.82	42.97
	GMMNet(k=3)	36.91	39.70	41.19	42.32	43.58
	GMMNet(k=4)	37.24	39.97	41.45	42.61	43.80
	GMMNet(k=5)	37.70	40.14	41.65	42.84	43.90
	GMMNet(k=6)	37.89	40.26	41.73	42.97	44.02

REFERENCES

[1] M. Vetterli, P. Marziliano and T. Blu, "Sampling signals with finite rate of innovation," in IEEE Transactions on Signal Processing, vol. 50, no. 6, pp. 1417-1428, June 2002, doi: 10.1109/TSP.2002.1003065.

[2] D. L. Donoho, "Compressed sensing," in IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006, doi: 10.1109/TIT.2006.871582.

[3] Y. C. Pati, R. Rezaiifar and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1993, pp. 40-44 vol.1, doi: 10.1109/ACSSC.1993.342465.

[4] 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, doi: 10.1109/ACSSC.1994.471413.

[5] 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, doi: 10.1109/MMSP.2017.8122281.

[6] Y. Jiang and H. F. Frank Leung, "Gaussian Mixture Model and Gaussian Supervector for Image Classification," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631558.

[7] Y. Liu, M. Li and D. A. Pados, "Motion-Aware Decoding of Compressed-Sensed Video," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 3, pp. 438-444, March 2013, doi: 10.1109/TCSVT.2012.2207269.

[8] Y. Pei, Y. Liu and N. Ling, "Deep Learning for Block-Level Compressive Video Sensing," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Sevilla, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9181254.

[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, doi: 10.1109/CVPR.2016.55.

[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, doi: 10.1109/TMI.2018.2858752.

[11] K.Murphy, “Machine Learning: a probabilistic perspective,” MIT Press, 2012.

[12] M. H. Lee and M. H. A. Khan, "Big Data ‘Fork’: Tensor Product for DCT-II/DST-II/ DFT/HWT," 2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Beijing, 2014, pp. 1-6, doi: 10.1109/BMSB.2014.6873510.

[13] P. Guccione, L. Mascolo and A. Appice, "Iterative Hyperspectral Image Classification Using Spectral–Spatial Relational Features," in IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 7, pp. 3615-3627, July 2015, doi: 10.1109/TGRS.2014.2380475.

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

[15] B. Hou, Y. Liu and N. Ling, "A Super-Fast Deep Network for Moving Object Detection," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Sevilla, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9181053.