# An Efficient Deep Quantized Compressed Sensing Coding Framework of Natural Images

Wenxue Cui†, Feng Jiang†, Xinwei Gao‡, Shengping Zhang†, Debin Zhao†
†Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
‡Wechat Business Group, Tencent, Shenzhen, China
wenxuecui@stu.hit.edu.cn, {fjiang, s.zhang, dbzhao}@hit.edu.cn, vitogao@tencent.com

## ABSTRACT

Traditional image compressed sensing (CS) coding frameworks solve an inverse problem that is based on the measurement coding tools (prediction, quantization, entropy coding, *etc.*) and the optimization based image reconstruction method. These CS coding frameworks face the challenges of improving the coding efficiency at the encoder, while also suffering from high computational complexity at the decoder. In this paper, we move forward a step and propose a novel deep network based CS coding framework of natural images, which consists of three sub-networks: sampling sub-network, offset sub-network and reconstruction sub-network that responsible for sampling, quantization and reconstruction, respectively. By cooperatively utilizing these sub-networks, it can be trained in the form of an end-to-end metric with a proposed rate-distortion optimization loss function. The proposed framework not only improves the coding performance, but also reduces the computational cost of the image reconstruction dramatically. Experimental results on benchmark datasets demonstrate that the proposed method is capable of achieving superior rate-distortion performance against state-of-the-art methods.

## KEYWORDS

Compressed sensing coding; image compression; deep neural network

**ACM Reference Format:**
Wenxue Cui†, Feng Jiang†, Xinwei Gao‡, Shengping Zhang†, Debin Zhao† . 2018. An Efficient Deep Quantized Compressed Sensing Coding Framework of Natural Images. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240706


*MM '18, October 22–26, 2018, Seoul, Republic of Korea*

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240706

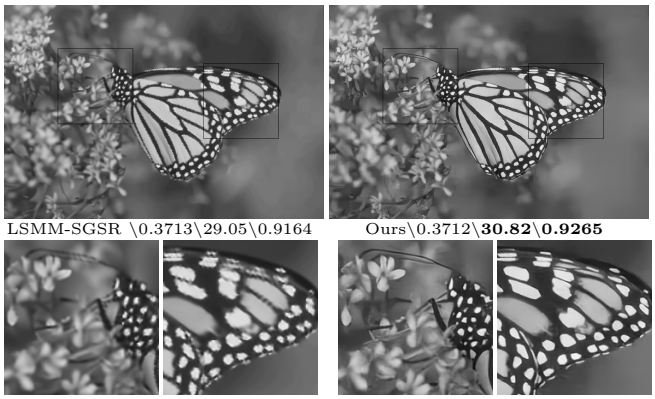
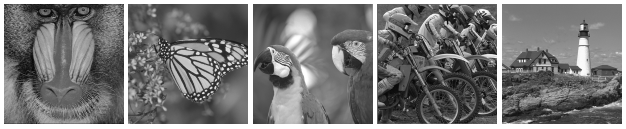



**Figure 1: Compared with the recent method LSMM-SGSR on *Monarch*. (Method\Bpp\PSNR\SSIM)**

## 1 INTRODUCTION

Recent years have seen significant interest in the paradigm of Compressed Sensing (CS) [5, 12], which is an emerging technique that samples a signal in a rate much lower than the conventional Nyquist/Shannon sampling rate for sparse signals. The possible reduction of sampling rate is attractive for diverse imaging applications such as radar imaging [13], Magnetic Resonance Imaging (MRI) [31], bio-signals acquisition [10], and sensor networks [30]. Faithful recovery of CS heavily depends on its recovery method from its linear projections (*i.e.* measurements). To achieving perfect reconstruction, many algorithms [26, 29, 36, 40, 42] have been proposed, which are based on the block-based CS (BCS) [15] that samples original images block by block. However, these frameworks are not real compression in the strict information theoretic sense, because they cannot directly produce a bitstream from the sensing device hardware. In fact, they can only be seen as a technology of dimensionality reduction in essence [35]. Besides, obtaining the measurements with infinite precision is infeasible, because the signal acquisition is usually performed by using analog-to-digital converters (ADC) that quantizes each measurement into a predefined value with a finite number of bits for efficient storage, transmission and manipulation in most image sensors or communication systems. Therefore, Quantized Compressed Sensing (QCS) coding is studied considering the quantization process and entropy coding.

Quantization is the process by which the measurement vector is represented by a vector of elements from a finite set. The



(a) *baboon* (b) *monarch* (c) *parrots* (d) *bikes* (e) *lighthouse*

**Figure 2: The test images**

finiteness allows its efficient digital storage and processing. Meanwhile, the quantization process makes recovering sparse signals difficult, as it might give rise to significant measurement errors. Numerous sparse signal recovery algorithms with quantized measurements have been proposed [4, 8, 18, 28, 38] . However, due to the random characteristic of the measurements generated by a random matrix, simple isometric scalar quantization does not perform well in rate-distortion performance at most cases.

In order to reduce the storage requirements, diverse rate control strategies are presented [17, 35, 43]. Inspired by the success of block-based hybrid video coding, such as MPEG-2 [19], H.264 [39], HEVC [37], the intra and inter prediction coding technology have the potential to be applied for CS measurement coding. Some hybrid coding based CS coding frameworks have been proposed [17, 35, 43]. However, all these above block-based hybrid CS coding frameworks still exist the following disadvantages:(1) highly computational complexity is required to solve a large-scale optimization problem; (2) simple scalar quantization operator is utilized, which restricts the efficiency of CS coding; (3) zero-order entropy is used to take place of entropy coding, which means no real bitstream is produced in these frameworks; (4) the sampling matrix is obtained randomly, which significantly limits the coding efficiency and reconstruction performance. Moreover, massive block artifacts and ringing effects are delivered at low bit rates.

To overcome the shortcomings of the aforementioned methods, we propose a Deep Quantized Block-based Compressed Sensing (DQBCS) coding framework as shown in Fig. 3, which includes five sub-modules: sampling , quantization ($Q$), entropy coding, enhanced inverse quantization ($\hat{Q}^{-1}$) and reconstruction. Specifically, the measurement is first obtained by the sampling sub-network that is used to mimic the sampling operator. Then the bit stream is produced by the entropy coding from the quantized measurement. In the inverse quantization process, an offset sub-network is introduced to counteract the quantization distortion for superior reconstruction performance. At last, a reconstruction sub-network is followed to recover the target images from the measurement domain to the image domain. It is worth emphasizing that the proposed framework can be optimized holistically except the entropy coding sub-module in the form of an end-to-end metric with the proposed loss function.

The contributions of this paper are summarized as follows:

- We propose an efficient Deep Neural Network based end-to-end quantized compressed sensing coding framework.
- In encoder, a learnable sampling matrix and an efficient entropy coding module are designed, which can save coding bits significantly.
- In decoder, an novel offset sub-network is introduced to counteract the quantization distortion. Besides, a reconstruction sub-network is proposed, in which several residual blocks are designed for boosting reconstruction performance.
- A novel rate-distortion optimization loss function is proposed, by which the proposed framework can be trained in form of an end-to-end metric.

The remainder of this paper is organized as follows: Section 2 reviews the related works. Section 3 elaborates the details of the proposed framework. Section 4 illustrates the experiments and discusses the results. Section 5 concludes the paper.

## 2 RELATED WORK

Compressed sensing (CS) has drawn quite an amount of attention as a joint sampling and compression methodology. The CS theory [5, 12] shows that if a signal is sparse in a certain domain $\boldsymbol{\Psi}$, it can be recovered with high probability from a small number of random linear measurements less than that of Nyquist sampling theorem. Mathematically, the measurements are obtained by the following linear formation $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{e}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is lexicographically stacked representations of the original signals and $\boldsymbol{y} \in \mathbb{R}^m$ is the CS measurements obtained by a $m \times n$ measurement matrix $\boldsymbol{\Phi}$, $(m \ll n)$. $\boldsymbol{e} \in \mathbb{R}^m$ indicates noise. CS aims to recover the signal $\boldsymbol{x}$ from its measurements $\boldsymbol{y}$ efficiently. It usually can be roughly divided into two categories: the ideal CS and the realistic CS coding based on whether the quantization operator is considered.

### 2.1 Ideal Compressed Sensing

In the study of CS, there are two most challenging issues including $(a)$ the design of the sampling operator $\boldsymbol{\Phi}$; $(b)$ the development of an efficient nonlinear reconstruction algorithm [15]. In the past decade, both of them have been extensively studied.

*2.1.1 Sampling Process.* In most works, the sampling matrix is a random matrix, for example, a Gaussian or Bernoulli matrix [41], which meets the Restricted Isometry Property (RIP) with a large probability [6]. However, such a random sampling always suffer form some problems such as high computation cost, vast storage and uncertain reconstruction qualities. For the Block-based Compressed Sensing (BCS) [15], Dinh *et al.* [9] propose a structural sampling matrix to balance the conflict between the compressed ratio and reconstructed quality. In [17], Gao *et al.* design a local structural sampling matrix by utilizing the local smooth property of neural images. Recently, several Deep Neural Network (DNN) based methods are proposed to learn more accurate sampling matrices. In [1], a fully connected layer is utilized to obtain the final sampling matrix. Shi *et al.* [36] propose to use a convolutional layer to mimic sampling process.

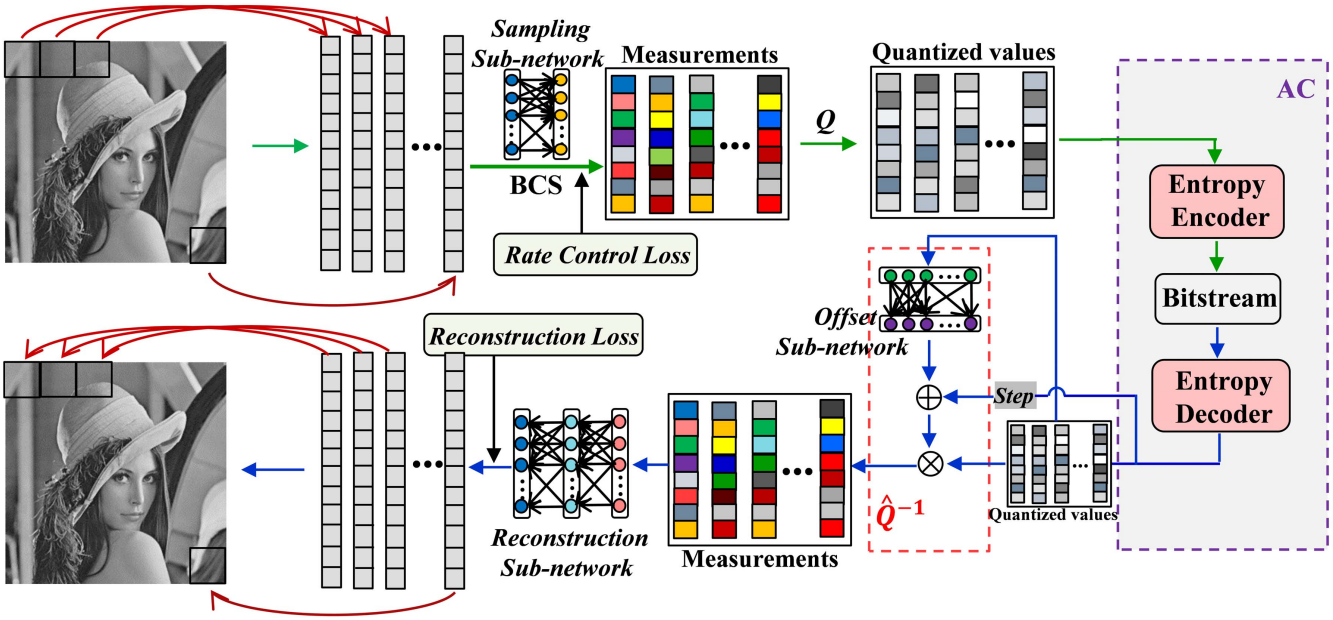



**Figure 3: The proposed Deep Quantization Block-based Compressed Sensing (DQBCS) framework**

*2.1.2 Reconstruction Process.* For CS reconstruction, many algorithms have been proposed, which can be generally divided into two categories: traditional optimization-based CS methods and recent network-based CS methods.

**Optimization-based CS methods.** Given the linear measurements $\boldsymbol{y}$ and according to CS theory, traditional image CS methods usually reconstruct the original image $\boldsymbol{x}$ by solving the following convex optimization problem

$$\tilde{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2} \| \boldsymbol{\Phi}\, \boldsymbol{x} - \boldsymbol{y} \|_2^2 + \lambda \| \boldsymbol{\Psi}\, \boldsymbol{x} \|_{l_p} \tag{1}$$

where $\lambda$ is the regularization parameter to control the tradeoff of fidelity term and sparsity term. The sparsity of the vector $\boldsymbol{\Psi x}$ is characterized by the $l_p$ norm. Many classic domains (*e.g.* DCT, wavelet [34], and gradient domain [29]) have been applied in modeling Eq. 1. Besides, many works incorporate additional prior knowledge about transform coefficients (*e.g.* statistical dependencies [23], structure [22], *etc.*) into the CS recovery framework. Furthermore, some elaborate priors exploiting the non-local self-similarity properties of natural images, such as the collaborative sparsity prior [44] or low-rank prior [11], have been proposed to improve CS reconstruction performance. However, all these traditional image CS reconstruction algorithms require hundreds of iterations to solve Eq. 1 by means of some iterative solvers (*e.g.* ISTA [3], ADMM [2], or AMP [33]), which inevitably gives rise to high computational cost and restricts the application of CS.

**Neural Network based CS methods.** Recently, several neural network based algorithms have been proposed for image CS reconstruction. Kulkarni *et al.* [26] propose a convolutional neural network based CS reconstruction algorithm, dubbed ReconNet, in which a nonlinear mapping from measurement domain to image domain is learned directly. In [40], a Deep Residual Reconstruction Network (DR$^2$-Net) is proposed for image compressed sensing, which introduces the residual learning block to promote the reconstruction performance. In [7], a deep convolutional laplacian pyramid architecture is proposed to reconstruct the target images progressively. Obviously, network-based image CS algorithms are non-iterative, which dramatically reduces the computational complexity compared with their optimization-based counterparts.

However, the aforementioned methods are not the real compression in the strict information theoretic sense, because it cannot directly produce a bitstream from the sensing device hardware, which can be only seen as a technology of dimensionality reduction in essence.

## 2.2 Realistic Compressed Sensing Coding

In order to achieve efficient storage and manipulation, a quantization operation is introduced in CS, named Quantized Compressed Sensing (QCS). Mathematically, a practical

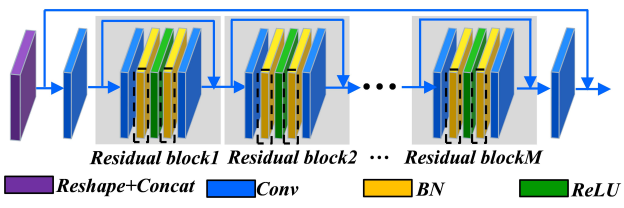



**Figure 4: The network structure of Offset sub-network (without purple block) and Reconstruction sub-network (without blocks in dashed).**

quantization is of the form: $\mathcal{Q}: \mathbb{R}^m \rightarrow \mathcal{A}^m$. where $\mathcal{A} \subset \mathbb{R}$ is a finite set, called the quantization alphabet. In BCS, both the memoryless scalar quantization (MSQ) [27] and the Sigma-Delta ($\Sigma\Delta$) quantization [18, 25] have been extensively studied. Furthermore, inspired by the technology of the intra and inter prediction in the block-based hybrid video coding (*e.g.* MPEG-2 [19], H.264 [39], HEVC [37]), several hybrid coding based CS coding frameworks have been presented. Mun and Fowler [35] propose the block-based quantized compressed sensing coding of natural images with different pulse-code modulation (DPCM) and uniform scalar quantization. Zhang *et al.* [43] extend the DPCM based CS coding and propose the spatially directional predictive coding (SDPC), in which the intrinsic spatial correlation between neighbouring measurements of natural images are further explored. In the BCS measurement coding, Khanh *et al.* [9] point out that, the spatial correlation among neighboring blocks becomes higher as block size decreases and the CS recovery of a small block is less efficient than that of a large block. In order to balance the conflict between the compressed ratio and reconstructed quality, a structural measurement matrix (SMM) is proposed to achieve a better rate-distortion performance, in which the image is sampled by some small blocks, and reconstructed with large blocks spliced by the small blocks. In considering the local smooth property of natural images, Gao *et al.* [17] propose a local structural measurement matrix (LSMM), in which an auxiliary bound constraint for quantization interval is integrated into the final optimization problem. However, little gains are received because of its rough modeling for quantization errors.

# 3 PROPOSED METHOD

In this section, we describe the methodology of the proposed DQBCS framework, which includes five sub-modules: sampling sub-network, quantization, entropy coding, enhanced inverse quantization, reconstruction sub-network.

## 3.1 Sampling Sub-network

In the traditional block-based compressed sensing (BCS) [15], the sampling operation is actually a series of convolution operations and each row of the sampling matrix $\mathbf{\Phi}$ can be considered as a filter. Therefore, the sampling process can be mimicked by using a convolutional layer [1, 36]. In our model, we use a convolutional layer with $B \times B$ filters and set stride as B for non-overlapping $B \times B$ blocks to mimic the non-overlapping sampling operation. Specifically, given an image $I$ with size $w \times h$, there are a total of $L = \lfloor \frac{w}{B} \rfloor \times \lfloor \frac{h}{B} \rfloor$ non-overlapping blocks $\{b_1, b_2, ..., b_L\}$ with size $B \times B$. The dimensions of measurements for each block is $n_B = \lfloor \frac{m}{n} B^2 \rfloor$, where $\frac{m}{n}$ is named for subrate. Therefore, the dimensions of measurements for the current image is $L \times n_B$. For traditional random sampling, regardless of the characteristics of signal the generated measurements are also random, which results in low efficiency of the measurement coding. Besides, the traditional sampling matrix is fixed for various reconstruction algorithms, which cuts down its flexibility for diverse reconstruction methods. In our work, the sampling matrix is learned jointly with the reconstruction process from large amount of data. In other words, the optimization of the sampling matrix is guided by the reconstruction process for superior reconstruction performance.

## 3.2 Quantization

Given the measurement $m_j$ of the $j^{th}$ block $b_j$ for image $I$, like most literatures [17, 35, 43], we utilize the following quantization criterion

$$m_j^q = \mathcal{Q}(m_j) = ROUND(m_j / Step) \tag{2}$$

where $Step$ is the step size of quantization, $ROUND(\cdot)$ indicates the round operation and $m_j^q$ is the quantized measurement for the current $j^{th}$ block. While unlike the aforementioned works, we quantize the measurement directly, instead of quantizing the residual between the measurement and its prediction. In other words, our method can be implemented easily and requires less time compared with the prediction-based algorithms mentioned above. Unfortunately, the round operation is not differentiable when performing the back propagation algorithm. Therefore, we regard it as a differentiable function: $m_j^q = (m_j / Step)$ approximately, whose derivative is $\frac{1}{Step}$, which can be used for back-propagation during optimization. At last, the quantized measurements are sent to the entropy coding module for generating bitstreams.

## 3.3 Entropy Coding

After obtaining the quantized measurements, the entropy coding is required to remove the statistical redundancies and generate the bitstreams. Inspired by [16], we utilize an arithmetic coding scheme (AC) for our CS coding framework. Given the quantized measurements $m_j^q = [m_j^1, ..., m_j^k, ..., m_j^{n_B}]^T$, we find that the occurrence probability of the significant components is independent of their positions within the quantized measurements. From the statistical information on the quantized measurements, we also find that most of the quantized measurement values are zeros and the significant component is randomly distributed throughout the quantized measurements. Therefore, we use a syntax element namely *significant_map* to indicate the significance of each component within the quantized measurements. For each significant component, the syntax elements *abs_coeff_level* and *sign_flag* are signaled to indicate its absolute magnitude and sign, respectively.

Based on the designed syntax elements, the coding of a quantized measurement is summarized as follows. First, the *significant_map* is transmitted for each component in the quantized measurements to indicate the locations of the significant components. If the *significant_map* for component $m_j^k$ is equal to one, it means that $m_j^k$ is a significant component. Then, the *abs_coeff_level* and *sign_flag* are signaled for each significant component to indicate its magnitude and sign, respectively. In order to efficiently code these syntax elements, the binary arithmetic coding engine M-coder in CABAC [32] is adopted in the proposed arithmetic coding

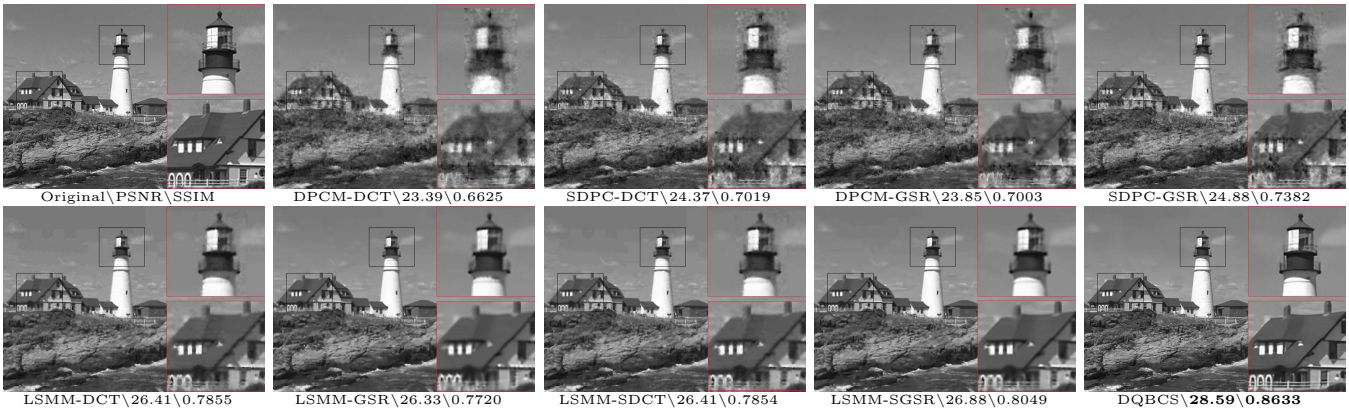



**Figure 5: Visual quality comparison of different CS coding frameworks on image *Lighthouse* from LIVE1 in the case of bpp = 0.4**

scheme, in which the probability update module can adaptively update the probability according to the previously coded symbols.

### 3.4 Enhanced Inverse Quantization

Inverse quantization is the inverse process of the quantization. In most works, the simplest operation

$$\tilde{m}_j = \mathcal{Q}^{-1}(m_j^q) = (m_j^q \times Step) \tag{3}$$

is adopted to implement $Q^{-1}$ for the current $j^{th}$ block. However, the unstable quantization distortion between $m_j$ and $\tilde{m}_j$ is introduced, which restricts the reconstruction performance significantly. In order to minimize the quantization distortion and according to the MSE distortion metric, optimization for $Q^{-1}$ is performed by solving

$$\tilde{\mathcal{Q}}^{-1} = \arg\min_{\mathcal{Q}^{-1}} E\{\|m_j - \mathcal{Q}^{-1}(m_j^q)\|_2^2\} \tag{4}$$

In our method, in order to obtain an excellent $\mathcal{Q}^{-1}$, a learnable *Offset* for *Step* is introduced to counteract the quantization distortion

$$\tilde{m}_j = \hat{\mathcal{Q}}^{-1}(m_j^q) = m_j^q \times (Step + Offset) \tag{5}$$

Therefore, the quantization distortion is modeled as

$$\mathcal{D} = m_j^q \times Offset \tag{6}$$

In other side, according to Eq. 3, the quantization distortion can also be modeled as

$$\mathcal{D} = m_j - m_j^q \times Step \tag{7}$$

We set $\mathcal{Q}^{\#}$ as the ideal inverse operation of $\mathcal{Q}$ in Eq. 2. *e.g.* $m_j = \mathcal{Q}^{\#}(m_j^q)$. Combining Eq. 6 and Eq. 7, we can obtain that

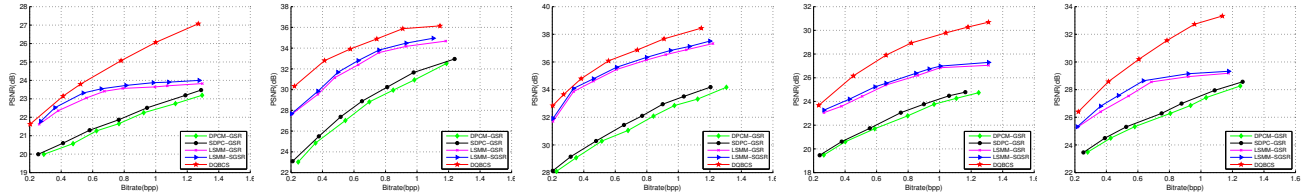
$$m_j^q \times Offset = \mathcal{Q}^{\#}(m_j^q) - m_j^q \times Step \tag{8}$$

Hence, *Offset* is a function about $m_j^q$, *e.g.* $Offset = f_q(m_j^q)$. Then, by combining Eq. 4 and Eq. 5 and according to the MSE distortion metric, optimization for $f_q$ can be performed by solving

$$\tilde{f}_q = \arg\min_{f_q} E\{\|m_j - m_j^q \times (Step + f_q(m_j^q))\|_2^2\} \tag{9}$$

In order to model the function $f_q$, an offset sub-network is designed to enhance the ability of nonlinear mapping as shown in Fig. 4, in which the residual learning and Batch Normalization (BN) are applied to speed up the training process and boost the performance. However, the sampling matrix is variable in our framework, which results in the difficulty of optimization for $f_q$. Furthermore, for the QCS framework, finding the quantizer that minimizes MSE between $m_j$ and $\hat{\mathcal{Q}}^{-1}(m_j^q)$ is not necessarily equivalent to minimizing MSE between the original image block $b_j$ and its CS reconstruction $\hat{b}_j$ from quantized measurements $m_j^q$. More specifically, $\hat{b}_j = Rec(m_j^q \times (Step + f_q(m_j^q)))$, where $Rec(\cdot)$ indicates the reconstruction algorithm. Hence, instead of solving 9, it is more interesting to solve

$$\tilde{f}_q = \arg\min_{f_q} E\{\|b_j - Rec(m_j^q \times (Step + f_q(m_j^q)))\|_2^2\} \tag{10}$$




(a) *Baboon* (b) *Monarch* (c) *Parrots* (d) *Bikes* (e) *Lighthouse*

**Figure 6: The rate-distortion performance on test images**

**Table 1: The results (PSNR\SSIM) on test images for different CS coding frameworks**

| Methods | bpp=0.2 | bpp=0.4 | bpp=0.6 | bpp=0.8 | bpp=1.0 | Avg. |
|---|---|---|---|---|---|---|
| DPCM-DCT [34, 35] | 21.80\0.4666 | 23.34\0.6239 | 24.52\0.6716 | 25.58\0.7148 | 26.66\0.7447 | 24.38\0.6443 |
| SDPC-DCT [34, 43] | 22.56\0.4994 | 23.87\0.6267 | 24.95\0.6750 | 25.96\0.7208 | 26.95\0.7560 | 24.85\0.6556 |
| DPCM-GSR [35, 42] | 22.03\0.4919 | 23.89\0.6700 | 25.26\0.7109 | 26.42\0.7604 | 27.61\0.7927 | 25.04\0.6852 |
| SDPC-GSR [42, 43] | 22.64\0.5219 | 24.35\0.6707 | 25.51\0.7239 | 26.87\0.7679 | 28.00\0.8059 | 25.47\0.6981 |
| LSMM-DCT [17, 34] | 25.35\0.7006 | 26.97\0.7609 | 27.95\0.7955 | 28.78\0.8118 | 29.43\0.8318 | 27.70\0.7801 |
| LSMM-GSR [17, 42] | 25.59\0.7152 | 27.57\0.7874 | 28.67\0.8182 | 29.52\0.8370 | 30.25\0.8581 | 28.32\0.8032 |
| LSMM-SDCT [17] | 25.61\0.7086 | 27.16\0.7763 | 28.09\0.8043 | 28.95\0.8222 | 29.64\0.8365 | 27.89\0.7896 |
| LSMM-SGSR [17] | 25.76\0.7206 | 27.74\0.7961 | 28.88\0.8259 | 29.77\0.8462 | 30.42\0.8698 | 28.51\0.8117 |
| DQBCS | **26.47\0.7585** | **28.93\0.8336** | **30.41\0.8705** | **31.44\0.8906** | **32.65\0.9162** | **29.98\0.8539** |

Inferentially, for the current image $I$, the optimization can be formula as

$$\tilde{f}_q = \arg\min_{f_q} E\{\|I - \hat{I}\|_2^2\} \tag{11}$$

where $\hat{I} = Rec(I^q \times (Step + f_q(I^q)))$, in which $I^q$ is the quantized measurements for current image $I$. *e.g.* $I^q = \{m_1^q, m_2^q, ..., m_L^q\}$

## 3.5 Reconstruction Sub-network

For the CS reconstruction, several deep network based models have been proposed [26, 40]. While these methods omit the quantization process, which reconstruct target images from the measurements directly. Besides, these methods are implemented independently for each block, which ignore the relationship between blocks and therefore results in serious blocking artifacts in most cases. To address these problems, we utilize a "reshape+concat" layer [36] to concatenate all blocks to obtain an initial reconstruction, which is then refined to obtain a further superior reconstruction. The topology of proposed reconstruction sub-network applies the residual learning technology as shown in Fig. 4

*3.5.1 Initial reconstruction.* Given the compressed measurements $\tilde{m}_j$ for the $j^{th}$ block, the initial reconstruction block $\tilde{x}_j$ can be obtained by

$$\tilde{x}_j = \mathcal{F}^f(\tilde{m}_j, \tilde{\Phi}_B) = \tilde{\Phi}_B \tilde{m}_j \tag{12}$$

where $\tilde{\Phi}_B$ is a $B^2 \times n_B$ mapping matrix. The traditional BCS [15] methods use the MMSE linear estimation to obtain $\tilde{\Phi}_B$

$$\tilde{\Phi}_B = R_{xx}\Phi_B^T(\Phi_B R_{xx}\Phi_B^T)^{-1} \tag{13}$$

where $R_{xx}$ is the autocorrelation function of the input signal. In fact, $\tilde{\Phi}_B$ can be optimized using a convolutional network. Specifically, $\tilde{m}_j$ is a $n_B \times 1$ vector, so the size of the convolution filter in the initial reconstruction layer is $1 \times 1 \times n_B$. We use $1 \times 1$ stride convolution to reconstruct each block. Since the dimension of target block is $B \times B$, $B^2$ convolution filters of size $1 \times 1 \times n_B$ are used. However, the reconstructed output of each block is still a vector ($\tilde{x}_j$). To obtain the initial reconstructed image, a "reshape+concat" layer [36] is adopted. This layer first reshapes each $B^2$ reconstructed vector $\tilde{x}_j$ into a $B \times B$ block, then concatenates all the blocks to get the reconstructed image

$$\tilde{I} = CatR(\tilde{x}_j) \tag{14}$$

where $CatR(\cdot)$ indicates the reshape and concatenation operation.

*3.5.2 Further reconstruction.* To further narrow down the gap between $\tilde{I}$ and $I$, a further reconstruction sub-network is designed to estimate the gap between them. In other words, the residual $d$ is inferred based on $\tilde{I}$, *i.e.*,

$$d = I - \tilde{I} \tag{15}$$

Specifically, we implement the residual learning [21] with residual learning blocks shown in Fig. 4. Each block contains two convolution layers and a ReLU layer between them. We donate the residual network as $f_d(\tilde{I}, \theta_r)$, where $\theta_r$ is its parameters. The procedure can be expressed as

$$\tilde{d} = f_d(\tilde{I}, \theta_r) \tag{16}$$

The reconstruction sub-network takes $\tilde{m}_j (j = 1, 2...L)$ as input. It first obtains the preliminary reconstructed image $\tilde{I}$, then fuse it with the estimated residual $\tilde{d}$ from the residual

**Table 2: Running time for different CS coding frameworks on test images in case of $\frac{m}{n} = 0.1$**

| Methods | DPCM-DCT | SDPC-DCT | DPCM-GSR | SDPC-GSR | LSMM-DCT | LSMM-GSR | LSMM-SDCT | LSMM-SGSR | DQBCS-CPU | DQBCS-GPU |
|---|---|---|---|---|---|---|---|---|---|---|
| Sampling(s) | 0.069 | 0.069 | 0.069 | 0.069 | 0.052 | 0.052 | 0.052 | 0.052 | 0.067 | **0.022** |
| Coding(s) | 0.062 | 0.074 | 0.062 | 0.074 | 1.304 | 1.304 | 1.304 | 1.304 | 0.011 | **0.002** |
| Decoding(s) | 80.300 | 72.020 | 1702.300 | 1694.300 | 70.150 | 1664.800 | 97.066 | 1657.000 | 0.454 | **0.121** |
| Totals(s) | 80.431 | 72.163 | 1702.431 | 1694.443 | 71.506 | 1666.156 | 98.422 | 1658.356 | 0.532 | **0.145** |

**Table 3: The results (PSNR\SSIM) on Set14**

| Bpp | 0.2 | 0.4 | 0.6 |
|---|---|---|---|
| DPCM-GSR | 21.43\0.4474 | 23.97\0.6516 | 25.51\0.7020 |
| SDPC-GSR | 21.69\0.4631 | 24.20\0.6571 | 25.69\0.7716 |
| LSMM-GSR | 25.19\0.6930 | 26.58\0.7630 | 28.43\0.8079 |
| LSMM-SGSR | 25.36\0.7014 | 26.74\0.7687 | 28.68\0.8120 |
| DQBCS | **25.98\0.7404** | **28.30\0.8204** | **29.99\0.8644** |

**Table 4: The results (PSNR\SSIM) on LIVE1**

| Bpp | 0.2 | 0.4 | 0.6 |
|---|---|---|---|
| DPCM-GSR | 21.59\0.4478 | 23.74\0.6405 | 25.00\0.6945 |
| SDPC-GSR | 21.83\0.4625 | 24.01\0.6463 | 25.20\0.6956 |
| LSMM-GSR | 24.96\0.6912 | 26.63\0.7584 | 27.79\0.8006 |
| LSMM-SGSR | 25.16\0.7038 | 26.84\0.7639 | 27.96\0.8103 |
| DQBCS | **25.78\0.7360** | **28.05\0.8244** | **29.71\0.8700** |

network as the final result, *i.e.*,

$$I^* = \tilde{I} + \tilde{d} \tag{17}$$

By replacing the $\tilde{I}$ and $\tilde{d}$ with Eq. 14 and Eq. 16, the final $I^*$ is obtained by

$$I^* = CatR(\mathcal{F}^f(\tilde{m}_j, \tilde{\Phi}_B)) + f_d(CatR(\mathcal{F}^f(\tilde{m}_j, \tilde{\Phi}_B)), \theta_r) \tag{18}$$

In order to minimize the gap between $I$ and $I^*$, and according to the MSE distortion metric, optimization is performed by solving

$$\theta_r^\Phi = \tilde{\Phi}_B, \theta_r = \arg\min_{\tilde{\Phi}_B, \theta_r} E\{\|I - I^*\|_2^2\} \tag{19}$$

where $\theta_r^\Phi$ is the mixture of $\tilde{\Phi}_B$ and $\theta_r$.

# 4 EXPERIMENTS AND ANALYSIS

## 4.1 Implementation and Training Details

*4.1.1 Loss Function.* Let $x$ be the input image, $\theta_s$ is the parameter of the sampling sub-network, $\theta_q$ is the parameter of the offset sub-network and $\theta_r^\Phi$ is the parameter of the reconstruction sub-network. Three mapping functions $f_s$, $f_q$ and $f_r$ are desired for sampling, inverse quantization and reconstruction respectively.

In fact, we have two main missions in our framework for facilitating rate-distortion performance. One is rate controlling, the other is reconstruction performance promoting. Therefore, in order to check and balance the cooperation between them, the complete loss function can be expressed as

$$L(\theta_s, \theta_q, \theta_r^\Phi) = \lambda L_{rate}(\theta_s) + L_{rec}(\theta_s, \theta_q, \theta_r^\Phi) \tag{20}$$

where $L_{rate}(\theta_s)$ is the rate control loss for efficient rate controlling and $L_{rec}(\theta_s, \theta_q, \theta_r^\Phi)$ is the reconstruction loss for superior reconstruction performance. The $\lambda$ is the regularization parameter to control the tradeoff of them. Specifically, the rate control loss is defined as

$$L_{rate}(\theta_s) = \frac{1}{N}\sum_{i=1}^{N} ||f_s(x_i; \theta_s) - R_{(0,\sigma)}||_2^2 \tag{21}$$

where $R_{(0,\sigma)}$ indicates a group of random numbers, which conform to a normal distribution with mean value 0 and variance $\sigma$. In order to ensure the reconstruction performance, another reconstruction loss is introduced

$$L_{rec}(\theta_s, \theta_q, \theta_r^\Phi) = \frac{1}{N}\sum_{i=1}^{N} ||f_r(f_q(\mathcal{Q}(f_s(x_i; \theta_s)), \theta_q), \theta_r^\Phi) - x_i||_2^2 \tag{22}$$

In our model, we set $\sigma = 0.1$ and $\lambda = 0.01$.

*4.1.2 Training Details.* In the training process, the proposed framework takes the original image blocks as the inputs as well as the labels during the optimization. Four sub modules are activated successively, namely the sampling sub-network, quantization, enhanced inverse quantization and the reconstruction sub-network. It is worth emphasizing that the entropy coding do not participate in the training process because of its lossless characteristics between the encoder and the decoder. In order to train the end-to-end framework, the derivative of the quantization operation is estimated as depicted in Subsection 3.2. The derivatives of the remaining three sub modules can be easily calculated because only linear operations are involved. Therefore, the back propagation of gradients can be implemented between each sub module.

*4.1.3 Experimental Setup.* In the sampling sub-network, we set B=32 and the number of filters is $n_B$. In the offset sub-network, all convolutional layers have $n_B$ kernels with size of $3 \times 3 \times n_B$ and the number of residual blocks is set as 4. In the reconstruction sub-network, the first convolutional layer has 64 kernels with size of $7 \times 7 \times 1$ and the last convolutional layer has a single kernel with size of $7 \times 7 \times 64$. The other convolutional layers in all residual blocks consist of 64 kernels with size of $3 \times 3 \times 64$ and there are 5 residual blocks. We initialize the convolutional filters using the same method as [20]. We pad zero values around the boundaries before applying convolution to keep the size of all feature maps the same as the input.

We use the training set of the VOC2012 database [14] for training, and the validation set of VOC2012 for validation. For testing, we use 5 images as shown in Fig. 2, which are widely used in the literatures. The patch's size is set as $128 \times 128$, which is cropped from training dataset randomly and we set batch size $N$ as 16. We train our model with the python toolbox Pytorch on a Titan X GPU. Adaptive moment estimation (Adam) [24] is used to optimize all network parameters. The learning rate is initialized to $1e-4$ for all layers and decreased by a factor of 3 for every 50 epochs. We train our model for 200 epochs.

## 4.2 Comparison with Existing Methods

*4.2.1 Rate-Distortion Performance.* To evaluate the performance of the proposed DQBCS framework, we conduct experimental comparisons against eight representative methods including DPCM-DCT [34, 35], SDPC-DCT [34, 43], DPCM-GSR [35, 42], SDPC-GSR [42, 43], LSMM-DCT [17, 34], LSMM-GSR [17, 42], LSMM-SDCT [17] and LSMM-SGSR [17]. It is worth emphasizing that the effectiveness of

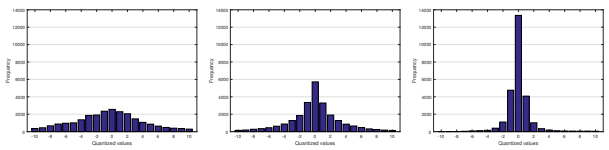
GSR based methods are more superior than that of DCT based methods. Therefore, only the GSR based methods are concerned in some tables and figures. We investigate several different bit rates with assessment criteria PSNR and SSIM as shown in Tables. 1, 3, 4, which proves that the proposed DQBCS achieves more than 1dB grains in PSNR compared against Gao's [17] that is the state-of-the-art in the compared methods. Besides, Fig. 6 shows the rate-distortion curve for test images, which demonstrates the rate-distortion performance intuitively. Obviously, the proposed framework significantly outperforms the compared methods by a large margin on all test bit-rates for all test images, which fully demonstrates the effectiveness of the proposed framework. The subjective performances for different bit rates in Fig. 1, 5 show that the proposed DQBCS is able to reconstruct more details and sharper edges.

*4.2.2 Running Time.* For running time, Table. 2 provides a strong proof of its high efficiency for test images. Specifically, the time for the sampling process is similar with others. While for the coding time, the proposed method is more faster than others, because of the pruning for prediction procedure. Due to the introduction of deep network technology, the decoding time is more efficient than the traditional optimization based algorithms. All the test experiments are implemented in Matlab 2015b on a Windows 7 system, and runs on a desktop computer with 8 cores CPU at 3.30 GHz and 32 GB RAM.

## 4.3 Further Analysis

The experimental results demonstrate that the proposed DQBCS significantly outperforms other state-of-the-art methods. However, the contributions of each sub-network are indistinct. In this sub-section, the performance of each sub-network is evaluated and analysed separately.

**Sampling sub-network.** In order to evaluate the performance of our sampling sub-network, we design a counterpart version of DQBCS, dubbed DQBCS(w/o)-S, in which the filters of sampling sub-network are replaced by the Gaussian Random Matrix (GRM) and only the other two sub-networks are updated during training. Compared DQBCS with DQBCS(w/o)-S in Table. 5, we found that the learned sampling matrix outperforms the random version significantly. Besides, because the distribution of measurements is closely related to the sampling matrix, the sampling matrix has an important contribution for rate controlling. Fig. 7 shows the



(a) GRM (b) CSNet (c) Ours

**Figure 7: Frequency distribution histogram of quantized measurements for different sampling matrices of image *Baby* in Set5 in case of $\frac{m}{n}$=0.1.**

**Table 5: The PSNR on Set5 for different methods**

| Methods | bpp=0.4 | bpp=0.6 | bpp=0.8 | Avg. |
|---|---|---|---|---|
| DPCM-GSR | 25.79 | 27.51 | 28.85 | 27.38 |
| SDPC-GSR | 26.04 | 27.71 | 29.17 | 27.64 |
| LSMM-GSR | 29.53 | 31.12 | 31.96 | 30.87 |
| LSMM-SGSR | 29.69 | 31.27 | 32.04 | 31.00 |
| DQBCS(w/o)-S | 26.69 | 27.99 | 29.03 | 27.90 |
| DQBCS(w/o)-O | 30.50 | 31.91 | 32.79 | 31.73 |
| DQBCS | **30.76** | **32.27** | **33.20** | **32.08** |

frequency distribution histogram of the quantized measurements based on image *Baby* in Set5 for three different sampling matrices: Gaussian Random Matrix (GRM), CSNet [36] and Ours in terms of $\frac{m}{n} = 0.1$, which fully demonstrates the effectiveness of proposed method for rate controlling.

**Offset sub-network.** In the proposed DQBCS, an offset sub-network is introduced to counteract the quantization distortion efficiently. In order to evaluate the performance of offset sub-network, another counterpart version of DQBCS—DQBCS(w/o)-O without the offset sub-network is designed, in which Eq. 3 is used to implement the inverse quantization. Compared DQBCS with DQBCS(w/o)-O in Table. 5, the offset sub-network is obviously effective by 0.4dB in terms of PSNR.

**Reconstruction sub-network.** In our proposed framework, a reconstruction sub-network is proposed to learn a nonlinear mapping from the measurement domain to the image domain. We compare the traditional CS coding frameworks and DQBCS(w/o)-S to evaluate the performance of reconstruction sub-network approximately. From Table. 5, we found that under the random sampling matrix, our reconstruction sub-network has no advantages compared with the optimization based algorithms. In other words, The random sampling matrix is not suitable for our CS coding framework and the potential of sampling matrix for researching is promising.

## 5 CONCLUSIONS

In this paper, we propose an efficient Deep Quantized Block-based Compressed Sensing coding framework (DQBCS) for compressed sensing coding, in which three sub-networks are included, dubbed: sampling sub-network, offset sub-network and reconstruction sub-network for their missions: sampling, quantization, reconstruction. These three sub-networks collaborate with each other and are trained by an end-to-end form using a proposed rate-distortion optimization loss function. Experimental results demonstrate that the proposed CS coding framework achieves state-of-the-art rate-distortion performance and is much faster than other algorithms.

## ACKNOWLEDGMENTS

This work is partially funded by the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant No. 61572155 and 61672188.

# REFERENCES


[1] Amir Adler, David Boublil, Michael Elad, and Michael Zibulevsky. 2016. A deep learning approach to block-based compressed sensing of images. *arXiv preprint arXiv:1606.01519* (2016).

[2] Manya V Afonso, Jos M Bioucas-Dias, and Mrio A T Figueiredo. 2009. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing* 20, 3 (2009), 681–695.

[3] Thomas Blumensath and Mike E. Davies. 2009. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis* 27, 3 (2009), 265–274.

[4] Petros T. Boufounos and Richard G. Baraniuk. 2008. 1-Bit compressive sensing. In *Conference on Information Sciences and Systems(CISS)*. 16–21.

[5] Emmanuel J Candès, Justin Romberg, and Terence Tao. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52, 2 (2006), 489–509.

[6] Cands and J Emmanuel. 2008. The restricted isometry property and its implications for compressed sensing. *Comptes rendus - Mathmatique* 346, 9 (2008), 589–592.

[7] Wenxue Cui, Heyao Xu, Xinwei Gao, Shengping Zhang, Feng Jiang, and Debin Zhao. 2018. An efficient deep convolutional laplacian pyramid architecture for CS reconstruction at low sampling ratios. *arXiv preprint arXiv:1804.04970* (2018).

[8] Wei Dai, Hoa Vinh Pham, and Olgica Milenkovic. 2009. Distortion-rate functions for quantized compressive sensing. In *IEEE Information Theory Workshop on Networking and Information Theory(ITW)*. 171–175.

[9] Khanh Quoc Dinh, Hiuk Jae Shim, and Byeungwoo Jeon. 2014. Measurement coding for compressive imaging using a structural measuremnet matrix. In *IEEE International Conference on Image Processing(ICIP)*. 10–13.

[10] A. M. Dixon, E. G. Allstot, D Gangopadhyay, and D. J. Allstot. 2012. Compressed sensing system considerations for ECG and EMG wireless biosensors. *IEEE Transactions Biomed Circuits System* 6, 2 (2012), 156–166.

[11] Weisheng Dong, Guangming Shi, Xin Li, Yi Ma, and Feng Huang. 2014. Compressive sensing via nonlocal low-rank regularization. *IEEE Transactions on Image Processing* 23, 8 (2014), 3618–3632.

[12] David L Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[13] Joachim H. G. Ender. 2010. On compressive sensing applied to radar. *Signal Processing* 90, 5 (2010), 1402–1414.

[14] Mark Everingham and John Winn. 2011. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit. (2011).

[15] Lu Gan. 2007. Block compressed sensing of natural images. *IEEE International Conference on Digital Signal Processing* (2007), 403–406.

[16] Min Gao. 2016. An Arithmetic Coding Scheme for Blocked-based Compressive Sensing of Images. *arXiv preprint arXiv:1604.06983* (2016).

[17] Xinwei Gao, Jian Zhang, Wenbin Che, Xiaopeng Fan, and Debin Zhao. 2015. Block-based compressive sensing coding of natural images by local structural measurement matrix. *IEEE Data Compression Conference (DCC)* (2015), 133–142.

[18] C. Sinan Gntrk, Mark Lammers, Alex Powell, Rayan Saab, and zgr Yilmaz. 2010. Sigma delta quantization for compressed sensing. In *Conference on Information Sciences and Systems(CISS)*. 1–6.

[19] Barry G Haskell, Atul Puri, and Arun N Netravali. 1998. Digital Video: An introduction to MPEG-2. *Journal of Electronic Imaging* 7, 1 (1998), 539–545.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision* (2015), 1026–1034.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (2016), 770–778.

[22] Lihan He and Lawrence Carin. 2009. Exploiting Structure in Wavelet-Based Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing* 57, 9 (2009), 3488–3497.

[23] Yookyung Kim, Mariappan S. Nadar, and Ali Bilgin. 2010. Compressed sensing using a Gaussian Scale Mixtures model in wavelet domain. In *IEEE International Conference on Image Processing(ICIP)*. 3365–3368.

[24] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *Computer Science* (2014).

[25] Felix Krahmer, Rayan Saab, and zgr Yilmaz. 2013. SigmaDelta quantization of sub-Gaussian frame expansions and its application to compressed sensing. *Solar Energy Materials and Solar Cells* 90, 14 (2013), 2087–2098.

[26] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. 2016. ReconNet: Non-iterative reconstruction of images from compressively sensed random measurements. *IEEE Conference on Computer Vision and Pattern Recongnition (CVPR)* (2016), 449–458.

[27] Jason Laska, Petros T. Boufounos, and Richard Baraniuk. 2009. Finite Range Scalar Quantization for Compressive Sensing. *Finite Range Scalar Quantization for Compressive Sensing* (2009).

[28] Jason N. Laska, Jason N. Laska, Petros T. Boufounos, and Richard G. Baraniuk. 2013. Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors. *IEEE Transactions on Information Theory* 59, 4 (2013), 2082–2102.

[29] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. 2013. An efficient augmented Lagrangian method with applications to total variation minimization. *Computational Optimization and Applications* 56, 3 (2013), 507–530.

[30] Shancang Li, Li Da Xu, and Xinheng Wang. 2013. Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things. *IEEE Transactions on Industrial Informatics* 9, 4 (2013), 2177–2186.

[31] Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. 2008. Compressed Sensing MRI. *IEEE Signal Processing Magazine* 25, 2 (2008), 72–82.

[32] D. Marpe, H. Schwarz, and T. Wiegand. 2003. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 620–636.

[33] Christopher A. Metzler, Arian Maleki, and Richard G. Baraniuk. 2016. From Denoising to Compressed Sensing. *IEEE Transactions on Information Theory* 62, 9 (2016), 5117–5144.

[34] Sungkwang Mun and James E. Fowler. 2010. Block compressed sensing of images using directional transforms. In *IEEE International Conference on Image Processing(ICIP)*. 2985–2988.

[35] Sungkwang Mun and James E. Fowler. 2012. DPCM for quantized block-based compressed sensing of images. In *Signal Processing Conference*. 1424–1428.

[36] Wuzhen Shi, Feng Jiang, Shengping Zhang, and Debin Zhao. 2017. Deep networks for compressed image sensing. *IEEE International Conference on Multimedia and Expo (ICME)* (2017), 877–882.

[37] G. J. Sullivan, J. Ohm, Woo Jin Han, and T. Wiegand. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1649–1668.

[38] John Sun and Vivek Goyal. 2010. Quantization for Compressed Sensing Reconstruction.

[39] Thomas Wiegand, Gary J Sullivan, Gisle Bjntegaard, and Ajay Luthra. 2003. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 560–576.

[40] Hantao Yao, Feng Dai, Dongming Zhang, Yike Ma, Shiliang Zhang, Yongdong Zhang, and Qi Tian. 2017. DR2-Net: Deep Residual Reconstruction Network for Image Compressive Sensing. (2017).

[41] Gesen Zhang, Shuhong Jiao, Xiaoli Xu, and Lan Wang. 2010. Compressed sensing and reconstruction with bernoulli matrices. In *IEEE International Conference on Information and Automation*. 455–460.

[42] Jian Zhang, Debin Zhao, and Wen Gao. 2014. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing* 23, 8 (2014), 3336–3351.

[43] Jian Zhang, Debin Zhao, and Feng Jiang. 2013. Spatially directional predictive coding for block-based compressive sensing of natural images. In *IEEE International Conference on Image Processing(ICIP)*. 1021–1025.

[44] Jian Zhang, Debin Zhao, Chen Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. 2012. Compressed sensing recovery via collaborative sparsity. *IEEE Data Compression Conference (DCC)* (2012), 287–296.