IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 17, 2019, accepted July 30, 2019, date of publication August 14, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934731

Learning a Deep Vector Quantization Network for Image Compression

XIAOTONG LU[1], HENG WANG[1], WEISHENG DONG[1], (Member, IEEE), FANGFANG WU[2], ZHONGLONG ZHENG[3], AND GUANGMING SHI[2], (Senior Member, IEEE)

[1]The State Key Laboratory on Integrated Services Network, School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[2]School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[3]Department of Computer Science, Zhejiang Normal University, Jinhua 321004, China

Corresponding author: Weisheng Dong (wsdong@mail.xidian.edu.cn)

This work was supported by the Natural Science Foundation of China under Grant 61622210, Grant 61632019, Grant 61836008, and Grant 61621005.

ABSTRACT Deep convolutional neural network (DCNN) based image codecs, consisting of encoder, quantizer and decoder, have achieved promising image compression results. The major challenge in learning these DCNN models lies in the joint optimization of the encoder, quantizer and decoder, as well as the adaptivity to the input images. In this paper, we proposed a DCNN architecture for image compression, where the encoder, quantizer and decoder are jointly learned. Specifically, a fully convolutional vector quantization network (VQNet) has been proposed to quantize the feature vectors of the image representation, where the representative vectors of the VQNet are jointly optimized with other network parameters through end-to-end training. While most of current DCNN-based methods were only trained once on large-scale datasets, we further perform fine-tuning of the encoder and the codes generated by the VQNet on the input images to further improve the compression performance. Extensive experimental results show that the proposed method achieves state-of-the-art compression results with simple encoder-decoder.

INDEX TERMS Image compression, deep learning, vector quantization, fine tune.

I. INTRODUCTION

Image compression, aiming to represent an image with as little bits as possible, plays a key role in image communication and computer vision applications. Generally, an image compression method first transforms an image into a transform domain, quantizes the coefficients, and then encodes the quantized coefficients. The decoder inverses the process to recover the image. Traditional image compression standards, e.g. PNG, JPEG and JPEG 2000 [20], use hand-crafted transformation and quantization methods, and thus cannot adapt to local image edges and textures, resulting in poor performance in preserving image details for low bit-rates.

Recently, inspired by the great success of the deep convolutional neural network (DCNN) for computer vision tasks [8], [13], [17], there have also been many efforts in developing DCNN-based image compression methods [2], [14], [16], [22]–[24]. The idea of DCNN-based image compression is straightforward. First, an auto-encoder is trained to obtain the latent representation of an image, followed by the quantization of the features, and then the quantized feature coefficients are coded by an entropy encoding method (e.g., Huffman coding or arithmetic coding methods). The decoder inverses the compression process. Due to the powerful nonlinear representation capability and the end-to-end training of the DCNN, these methods have achieved better compression performance than conventional methods in terms of perceptual quality and even PSNR.

Despite the effectiveness of the DCNN for this task, there are two key challenges in optimizing the encoder-quantizer-decoder network. First, as the quantization of image features is non-differentiability, it is difficult to optimize the quantizer and the auto-encoder jointly. Conventional quantizer with fixed quantization bins are often used. Second, the DCNN-based compressors trained on large-scale datasets may still not be able to well adapt to each input image, leading to limited performance. In contrast, the directional wavelet-based compression methods [7] and HEVC method [21],

The associate editor coordinating the review of this manuscript and approving it for publication was Yunjie Yang.

which optimize the parameters of the codec for each input image, have obtained state-of-the-art image compression performance.

In this paper, we proposed a fully convolutional neural network framework for image compression, jointly optimizing the encoder, quantizer and decoder. A novel vector quantization network (VQNet) is proposed to quantize the feature vectors of the latent representation of the input image. The proposed VQNet is simple and intuitive, where a feature vector is first classified into one of a set of quantization centers by a softmax classifier, and then the index of the center is used to represent the feature vector. For de-quantization, the representative vector of the corresponding center is used to recover the original vector. Both the quantization and de-quantization processes can be implemented by a fully connected layer or a convolutional layer. Thus, the quantization centers can be jointly optimized with encoder and decoder. After training the encoder-quantizer-decoder network on a large-scale dataset, we propose to further fine-tune the encoder and the quantization index generated by the VQNet on each input image, while keeping the quantization centers and decoder fixed. By fine-tuning, the proposed compression network can better adapt to the image edges and textures of the test images, leading to substantial improvements of the coding performance. With simple auto-encoder network, the proposed DCNN compressor already achieves compression performance competitive with current state-of-the-art methods.

II. RELATED WORKS

A. TRADITIONAL COMPRESSION METHODS

Traditional image compression methods, e.g., JPEG, JPEG 2000 [20] mainly contain three components, i.e., transformation, quantization and entropy encoding. For example, the DCT or wavelet transforms are used in JPEG and JPEG 2000 respectively, to obtain compact representations of an input image, followed by scalar-quantization of the representation coefficients. Entropy encoding schemes, e.g., Huffman and adaptive arithmetic coders are then used to code the quantized coefficients for JPEG and JPEG 2000, respectively. As the DCT and wavelet transforms are designed for piecewise stationary signals, they fail to adapt to local image edges and textures, leading to poor visual quality for low bit-rates. Directional wavelet transforms [7] have been developed to address the drawback of wavelet transform, significantly improving the compression performance of JPEG 2000. The idea of directional predictions has been further improved in the Better Port Graphics (BPG) method, which is built based on the intra-frame coding scheme of the high efficiency video coding (HEVC) standard [21]. Despite the careful design of the coding methods, there still has rooms for improvements of the compression performance, as the transform, quantization and encoding schemes were manually designed and cannot be jointly optimized. To overcome the drawback of these handcrafted modules, the sparse coding based methods [27], [28] have been proposed. In [27], to adapt to local image edges and textures, the iterative tuned and aligned dictionary (ITAD) learning method has been proposed to encode specific types of images (e.g., the facial images). In [28], a novel rate-distortion (RD) method was proposed to select the sparse representation of a target sparsity level and quantization levels for image compression, showing better compression performance than traditional sparse representation based methods.

B. DEEP LEARNING BASED METHODS

Inspired by the strong representation capability of DCNN, DCNN-based compression methods have been proposed. The basic idea of these methods is to first map an input image to a latent representation through an autoencoder, quantize the feature maps, followed by entropy encoding, and reconstruct the input by a decoder. By careful design and joint optimization of the encoder and decoder, promising compression results have been obtained. In [23], Toderici *et al.* used the recurrent neural networks (RNNs) for feature extraction and image reconstruction, and quantized the features into binary codes for entropy coding. The RNN-based method was further improved by introducing SSIM loss and spatially adaptive bits allocation [24]. In [22], convolutional autoencoders were used to extract and decode the features. In [18], Rippel and Bourdev exploited the multi-scale representation of natural images via a pyramidal decomposition and used generative adversarial networks (GANs) to improve the visual quality of the reconstructed images. To guide the bits allocation and thus preserve image details better, Li *et al.* proposed to jointly learn a content-weighted importance map with the encoder and decoder [14]. Considering that the dimensions of high-dimensional data can be effectively reduced by matrix factorization techniques [26], Cai *et al.* introduced a Tuck decomposition to reduce the dimensions of the feature maps before quantization to further reduce the redundancy in the latent representations [3]. Regarding the quantization, the above methods used fixed quantization bins, and used stochastic approximations [2], [4], [12] or smooth derivative approximation of the rounding function [9], [22] to address the gradient vanishing issue. In [1], a soft-to-hard vector quantization method has been proposed to quantize the features, where a continuous relaxation of quantization was adopted for back-propagation pass. Through joint training, the quantization levels can be learned. In [16], a context model based on a 3D-CNN was proposed to learn the conditional probability models to improve the entropy coding. In [29], a novel multiple description compression (MDC) method has been proposed using the DCNN. Specifically, two descriptions were generated through DCNN and compressed by a conventional codec. The images were reconstructed by a reconstruction network from the decompressed descriptions. In [30], a deep auto-encoder network was proposed to generate the multiple descriptions with the aid of entropy estimation and quantization networks. Through end-to-end training, these deep learning based MDC methods have achieved better compression performance than traditional MDC methods.

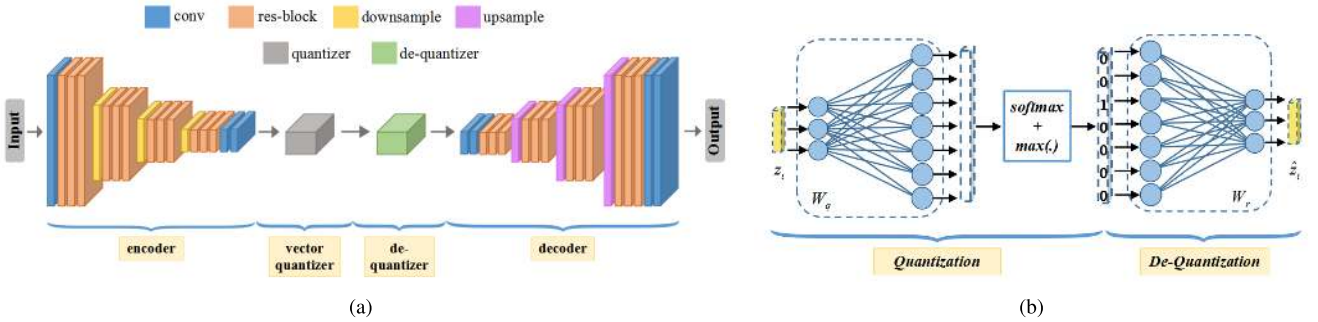


FIGURE 1. (a) The architecture of the proposed image compression network. (b) The architecture of the proposed VQNet.

In this paper, we proposed a vector quantization network (VQNet) to quantize the feature vectors, where the vector quantization is treated as a vector classification problem. Compared with the soft-to-hard VQ method of [1], our method is more simple and straightforward. Experimental results show that the proposed VQNet is more effective than that of [1]. While all the above mentioned methods learn encoder, quantizer and decoder from large-scale dataset, they may not be able to well adapt to each input image, resulting in limited performance. To address this issue, we propose to fine-tune the encoder and the codes (i.e., the indexes) generated by the VQNet on each input image while fixing the decoder, leading to further improvement of compression performance.

III. PROPOSED VQNET FOR IMAGE COMPRESSION

In this section, we first introduce the overall network architecture of the proposed image compression method, and then describe the proposed VQNet, followed by the joint training of all the components.

A. OVERALL NETWORK ARCHITECTURE

For a given training dataset $\mathcal{X} = x_{i=1}^{N}$, we aim to learn an image compression model, consisting of an encoder, a quantizer and a decoder. The encoder $E(\cdot)$ generates the latent representation of an input image x, as $Z = E(x)$. Then, the quantizer $Q(\cdot)$ discretizes the feature representations. Here, we use VQ to represent each feature vector with one of K representative vectors. The index of the representative vector is then losslessly entropy encoded. The decoder $D(\cdot)$ reconstructs the original image from the de-quantized features $\hat{x} = D(\hat{Z})$, where $\hat{Z}$ denotes the recovered feature representation decoded from the bitstream. To achieve good compression performance, we try to optimize $E(\cdot)$, $Q(\cdot)$ and $D(\cdot)$ such that both the distortion $\ell(x, \hat{x})$ and the bit-rates can be minimized, which can be formulated as

$$\min_{E,Q,D} \sum_{x \in \mathcal{X}} \{\ell(x, \hat{x}) + \beta \mathcal{R}(\hat{Z})\}, \tag{1}$$

where $\mathcal{R}(\hat{Z})$ denotes the rate loss, which can be expressed by the entropy of $\hat{Z}$.

TABLE 1. The architecture of the encoder and decoder.

layer	activate size
input	$128 \times 128 \times 3$
$(3 \times 3 \times 32$, pad=1, stride=1)	$128 \times 128 \times 32$
Res. block $\times 3$, No. bott. filter=8	$128 \times 128 \times 32$
$(3 \times 3 \times 64$, pad=1, stride=2)	$64 \times 64 \times 64$
Res. block $\times 3$, No. bott. filter=16	$64 \times 64 \times 64$
$(3 \times 3 \times 64$, pad=1, stride=2)	$32 \times 32 \times 128$
Res. block $\times 3$, No. bott. filter=32	$32 \times 32 \times 128$
$(3 \times 3 \times 64$, pad=1, stride=2)	$16 \times 16 \times 256$
Res. block $\times 3$, No. bott. filter=64	$16 \times 16 \times 256$
$(3 \times 3 \times 32$, pad=1, stride=1)	$16 \times 16 \times m$
VQ and De-Quantization	$16 \times 16 \times m$
$(1 \times 1 \times 32$, pad=0, stride=1)	$16 \times 16 \times m$
$(3 \times 3 \times 256$, pad=1, stride=1)	$16 \times 16 \times 256$
Res. block $\times 3$, No. bott. filter=64	$16 \times 16 \times 256$
Up-sample layer	$32 \times 32 \times 128$
Res. block $\times 3$, No. bott. filter=32	$32 \times 32 \times 128$
Up-sample layer	$64 \times 64 \times 64$
Res. block $\times 3$, No. bott. filter=16	$64 \times 64 \times 64$
Up-sample layer	$128 \times 128 \times 32$
Res. block $\times 3$, No. bott. filter=8	$128 \times 128 \times 32$
$(9 \times 9 \times 16$, pad=4, stride=1)	$128 \times 128 \times 16$
$(3 \times 3 \times 3$, pad=1, stride=1)+ Tanh	$128 \times 128 \times 3$
output	$128 \times 128 \times 3$

Encoder and decoder: The architecture of the convolutional encoder is shown in Fig. 1 (a). It consists of three types of layers, i.e., convolutional layer with stride 1, convolutional layer with stride 2, and residual block layers. An input image is first convoluted with filters of size 3×3, followed by a residual block, whose structure is followed with [8]. There are four residual blocks with the number of bottleneck filter 64. After each residual block, a 3×3 convolutional layer with stride 2 is used to reduce the spatial size of the feature maps, except the final layer where stride 1 is adopted. Finally, m feature maps of size 16×16 are generated by the encoder. For low bit-rates less than 0.5 bpp, we set $m = 32$ and set $m = 64$ for bit rates higher than 0.5 bpp. For the detailed setting of the encoder, please refer to Table 1.

The network architecture of the decoder is almost symmetric to that of the encoder, except that we replace the downsample layer with upsampling layer between two residual block layers. Here, we use the sub-pixel convolution technique

developed in [19] followed by convolution to increase the spatial size of the feature maps by factor 2. In the last layer of the decoder, we use the $Tanh(\cdot)$ function to project the values of the feature maps into the range of $[-1, 1]$. Please refer to Table 1 for the detailed settings of the decoder.

B. PROPOSED VQNET

In this paper, we use vector quantization (VQ) to quantize the output of the encoder. It has been proven that better performance can be obtained by coding vectors instead of scalars [5]. For a given feature vector $z_j \in \mathbb{R}^d$, VQ tries to represent z_j with one of K quantization centers $C = [c_1, c_2, \cdots, c_K] \in \mathbb{R}^{d\times K}$, denoted as c_{k_j}. Then, at the encoder side, the index k_j that can be represented in $L = \log_2 K$ bits is used to represent z_j, i.e., $k_j = Q(z_j)$. At the decoder side, the vector z_j will be recovered using the associated quantization center c_{k_i}, i.e., $\hat{z}_j = Q^{-1}(k_j) = c_{k_j}$. Let $Z \in \mathbb{R}^{w\times h\times m}$ denote the generated feature representation by the convolutional encoder for input image x. To quantize Z with VQ, we first divide it into many non-overlapping small blocks of size $r \times r \times s$ and then reshape these blocks into vectors, denoted as $z_j \in \mathbb{R}^d$, $d = r^2 s$, $j = 1, 2, \cdots, J$, and $J = \frac{w*h*m}{r^2*s}$.

The standard VQ assigns the index k_j to a given vector z_j by solving

$$k_j = \underset{k}{\operatorname{argmin}} ||z_j - c_k||_2^2, \quad k = 1, 2, \cdots, K. \tag{2}$$

Note that the ℓ_2-norm distance between c_k and z_j can be expressed as $d_{j,k} = ||c_k||^2 + ||z_j||^2 - 2c_k^\top z_j$. If the length of each centers c_k is constant, $d_{j,k}$ can be re-expressed as $d_{j,k} = -2c_k^\top z_j + Const$, where $Const$ denotes the constant independent on k. Thus, the distances between z_j and all the centers c_k can be easily computed as the inner products between z_j and C if the lengths of the centers have been normalized, as $d_j = -2C^\top z_j$, where we have removed the constant term. To this end, we normalized the centers such that each center is unitary. In neural network, the inner product between z_j and C can be conveniently implemented as a fully connected layer by setting the layer parameters as $W_q = C^\top$, and the nearest neighbor can be selected as the one having the largest inner product coefficient. The structure of the proposed VQ network is shown in Fig. 1 (b). In the VQNet, we convert the distances d_j into probabilities using the softmax operator, which we found is beneficial for optimizing the centers C in the back-propagation pass. Thus, it is interesting to see that the VQ problem can be solved by the softmax based classification technique. However, different from previously developed supervised feature classification methods, here, the proposed feature classification technique is unsupervised.

Let $b_j = [0, \cdots, 1, \cdots, 0] = \max(\text{softmax}(d_j))$ denote the obtained one-hot vector after the $\max(\cdot)$ operation, which can be represented by the index k_j. The vector z_j is then represented by the index k_j, followed by the entropy encoding. In the decoder side, when the index k_j is decoded from the bitstream, the vector z_j can then be recovered using the associated center c_{k_j}. Note that the de-quantization can also be computed as $\hat{z}_j = Cb_j$, which can also be implemented as a fully connected layer by setting the layer parameter matrix as $W_r = C$. Please refer to Fig. 1 (b) for the architecture of the VQNet. As the $\max(\cdot)$ operator is non-differentiable, we still face the non-differentiability problem. Similar to other methods [9], [14], [22], we set the derivative of the $\max(\cdot)$ operator to 1. Hence, the quantization centers C can be jointly optimized with other network parameters. In our method, we can even remove the constraint that W_r should be equal to $W_q^\top$ to learn reproduction vectors for reconstruction. However, our experiments show that letting $W_r \neq W_q$ doesn't improve the results. Hence, we let $W_r = W_q^\top$. This also indicates that normalizing the centers c_k doesn't effect the accuracy of the reconstruction of z_j. Our experiments also show that the normalization of the c_k improves the compression performance. Note that the fully connected layers W_q and W_r can also been implemented as convolutional layers with 1×1 filters, which helps to accelerate the training speed. In our implementation, both the layers W_q and W_r are implemented as convolutional layers. After de-quantization, the recovered feature vectors $\hat{z}_j$ are reshaped back into the feature space of size $w \times h \times m$ to form $\hat{Z}$, which is then fed into the decoder to reconstruct the image.

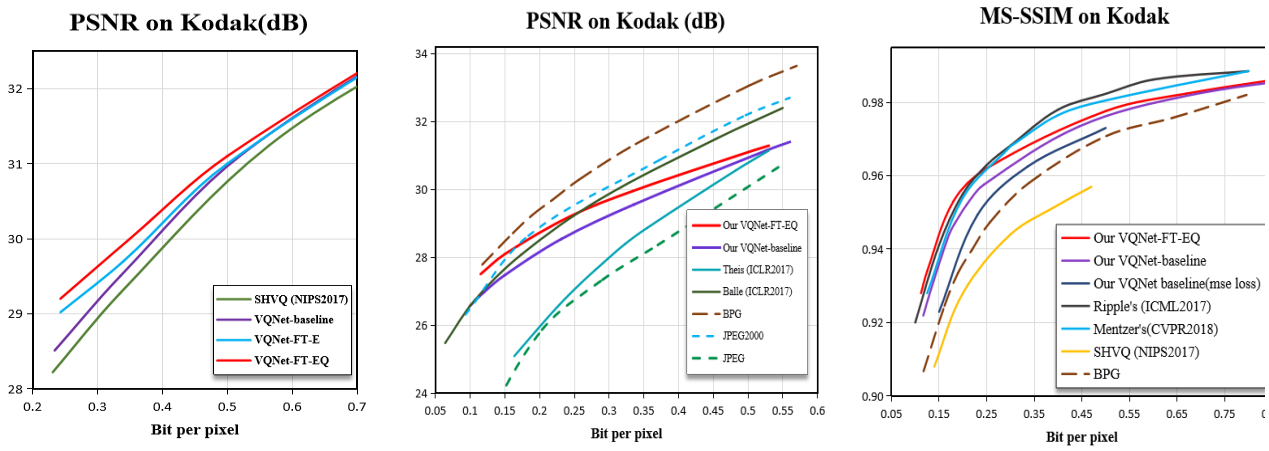
In [1], a soft-to-hard VQ method has also been developed to jointly learn the quantizer and the autoencoder. However, our proposed method is distinct from it in that our VQ method uses the inner product(i.e., angular distance) which can be easily implemented with fully connected layers, assuming that the lengths of centers are normalized. The use of inner product converts the VQ into a standard softmax classifier, which can be conveniently optimized. Our experimental results also verified that our VQNet is more effective than the VQ method proposed in [1]. The PSNR gains are between 0.2 and 0.3 dB on the Kodak dataset. We further proposed to fine-tune the autoencoder and the VQ for each test images IV, leading to further improvements.

C. JOINT OPTIMIZATION

To achieve good compression performance, the encoder, quantizer and the decoder should be jointly optimized via minimizing a rate-distortion objective function, as shown in Eq. (1). The distortion loss is used to ensure the accuracy of the decompressed image. The commonly used distortion loss is the mean square loss, i.e., $\mathcal{L}(\hat{x}, x) = ||\hat{x} - x||_2^2$. In addition, other perceptual loss, e.g., the SSIM metric of [25] can also be used for better perceptual quality.

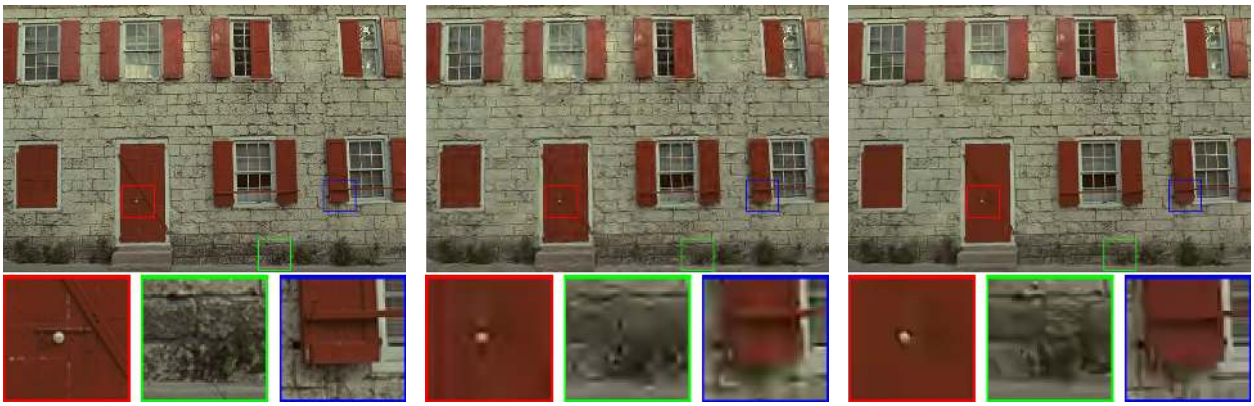
The bit-rate loss for an input image x can be measured by the entropy of the quantized feature vectors, i.e.,

$$H(\hat{Z}) = -\sum_{j=1}^{J}\sum_{k=1}^{K} p_k \log p_k = -J\sum_{k=1}^{K} p_k \log p_k, \tag{3}$$

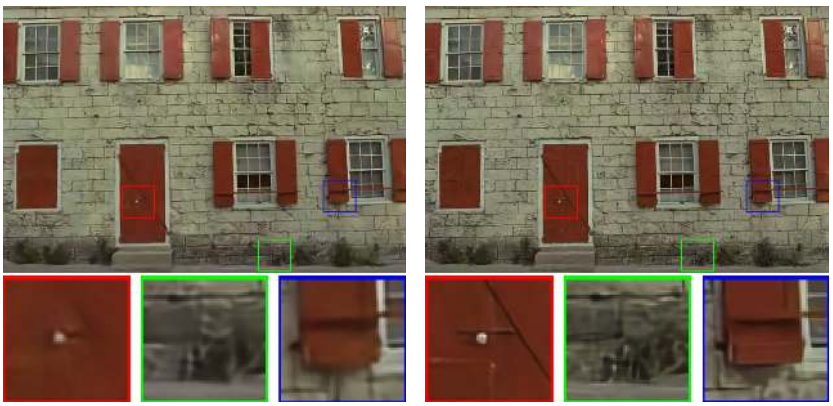


(a) (b) (c)

FIGURE 2. Rate-distortion curves of the test methods on Kodak dataset. (a) shows the performance's improvement by fine-tuning both encoder(Cyan Curve) and quantization(Red Curve). (b) and (c) show the PSNR and MS-SSIM curves under different training loss functions, respectively.



(a) **Original** (b) **JPEG2000** (0.186bpp, 24.25dB, 0.8732) (c) **BPG** (0.245bpp, 26.17dB, 0.9274)



(d) **Ballé** (0.234bpp, 25.50dB, 0.9220) (e) **Proposed** (0.252bpp, 26.10dB, 0.9308)

FIGURE 3. Visual comparison of the test methods on the *Wall* image (bits per pixel, PSNR, MS-SSIM).

where p_k denotes the probability of assigning index k to a feature vector z_j. Here, we assume that the vectors z_j are i.i.d. In practice, the probability p_k can be estimated by counting the number of samples followed into cluster k over the sample set. Since p_k is discrete, it is impossible to minimize the entropy defined in Eq. (3) w.r.t. the network parameters. To tackle this problem, we use the surrogate of the entropy adopted in [1], as

$$\tilde{H}(Z) = -J \sum_{k=1}^{K} q_k \log p_k, \tag{4}$$

(a) **Original** (b) **JPEG2000** (0.187bpp, 29.67dB, 0.9528) (c) **BPG** (0.262bpp, 32.84dB, 0.9772)

(d) **Ballé** (0.265bpp, 31.88dB, 0.9751) (e) **Proposed** (0.229bpp, 31.30dB, 0.9731)

FIGURE 4. **Visual comparison of the test methods on the *Leaves* image (bits per pixel, PSNR, MS-SSIM).**

where q_k denotes the soft histogram computed over the set of training images $\{x_i\}_{i=1}^N$, defined as

$$q_k = \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}\frac{e^{-\sigma d_{i,j,k}^2}}{\sum_{l=1}^{K} e^{-\sigma d_{i,j,l}^2}}, \tag{5}$$

where $d_{i,j,k} = -2c_k^\top z_{i,j}$ denote the distance between the feature vector $z_{i,j}$ and the center c_k. Using the soft histogram, the entropy can then be minimized w.r.t. the network parameters and the quantization centers. The rate-distortion objective function can thus be formulated as

$$(\Theta, C, \Omega) = \underset{\Theta,C,\Omega}{\operatorname{argmin}}\frac{1}{N}\sum_{i=1}^{N}\ell(D(\tilde{Q}(E(x_i,\Theta)),\Omega)) + \lambda J(\Theta,\Omega) + \beta\tilde{H}(Z_i), \tag{6}$$

where $\tilde{Q}(\cdot) = Q^{-1}(Q(\cdot))$, Θ and Ω denote the parameters of the encoder $E(\cdot)$ and decoder $D(\cdot)$, respectively. And in order to improve the generalization ability of the network and avoid over-fitting of the deep neural network, we added $J(\Theta, \Omega)$ as the ℓ_2 norm regularization of the network parameters and set λ as the hyperparameter. By adjusting the regularization parameter β, we can make a trade-off between the quality of the decoded image and the bit-rates.

IV. FINE-TUNING ON TEST IMAGES

A. FINE-TUNING OF ENCODER

Inspired by the success of directional wavelet compression method [7] and HEVC standards [21], which optimize parameters of the codecs on each input images to adapt to local image/video structures, we propose to fine-tune the image compression network on each input image. Different from the methods of [7], [21] that encode the codecs' parameters as the side information, we only fine-tune the encoder while frozening the quantizer and decoder. Thus, the quantizer and decoder and can still be used to reconstruct the input images, without the need to encode the parameters of the quanitzer and decoder. To fine-tune the encoder, we extract overlapped patches of size 256×256 with sliding step size $s = 16$ along two axes. Patch augmentation with flips and rotations is adopt, generating about 2000 patches for a typical test images of Kodak dataset. The setting of the fine-tuning is same as the off-line training on large-scale dataset, except that we use much smaller minibatch. Here, minibatch size of 8 is adopted, which leads to better results. We found that the fine-tuning converged after 30 epoches, leading to about 0.5 dB improvements on Kodak dataset at the case for low bitrates.

(a) **Original** (b) **JPEG2000** (0.187bpp, 31.32dB, 0.9706) (c) **BPG** (0.277bpp, 34.47dB, 0.9721)

(d) **Ballé** (0.209bpp, 31.81dB, 0.9616) (e) **Proposed** (0.226bpp, 32.08dB, 0.9701)

FIGURE 5. **Visual comparison of the test methods on the *Boat* image (bits per pixel, PSNR, MS-SSIM).**

(a) **Original** (b) **JPEG2000** (0.375bpp, 27.37dB, 0.9335) (c) **BPG** (0.247bpp, 27.00db, 0.9247) (d) **Proposed** (MS-SSIM loss) (0.253bpp, 25.10dB, 0.9591)

FIGURE 6. **Visual comparison of the test methods on the *House* image (bits per pixel, PSNR, MS-SSIM).**

B. RE-ASSIGNING QUANTIZATION CENTERS

Instead of fine-tuning the quantization centers that need to be encoded as side information to ensure correct de-quantization, we propose to fine-tune the assignment of the quantization centers for each feature vectors. The motivation is that the learned VQNet in the latent feature space according to the Euclidean distance cannot ensure the best compression performance for each test image. Let $q^{(0)} = [k_1, k_2, \cdots, k_J]$ denote the set of indexes assigned to the feature vectors $Z = [z_1, z_2, \cdots, z_J]$ for test image x. The basic idea of the VQNet fine-tuning is to optimize the indexes assigned to Z to further minimize the objective function of Eq. (6). Since the optimization w.r.t. the indexes is intractable, here we propose a random search technique for fine-tuning, similar to the genetic algorithm [6].

For each z_j, we perform the random searches within the centers that are similar to c_{k_j}. This is equivalent to select centers based on the similarities to c_{k_j}. The similarities between each pairs of the centers can be obtained by computing the covariance matrix $S = C^\top C$, where each row

$s_k = [s_{k,1}, s_{k,2}, \cdots, s_{k,K}]$ records the correlations between c_k and other centers. We apply the softmax operation to each s_k, and use the resulting values $p_k = [p_{k,1}, p_{k,2}, \cdots, p_{k,K}]$ as the probabilities for selecting the new centers. Then, we perform the random selection of new centers for z_j based on p_{k_j}, which can be implemented by a roulette wheel selection method. By applying the random search method M times for each vector in Z based on $q^{(0)}$, we can obtain a set of different quantization results, denoted as $q_m = [k_{1,m}, k_{2,m}, \cdots, k_{J,m}]$, $m = 1, 2, \cdots, M$. Using the set of different quantization results, different bitstream and thus different de-compressed images can be generated by de-quantization and reconstruction with the decoder. We can then choose the best solutions from $\{q_m\}_{m=1}^{M}$ and $q^{(0)}$, denoted as $q^{(1)}$, which minimizes the objective function of Eq. (6). By random search, we may obtain better quantization results in term of rate-distortion, rather than Euclidean distances. Based on $q^{(1)}$, we can obtain new quantization result $q^{(2)}$ using the above described random re-assignment scheme. Such process can be repeated T times for better performance. The proposed VQ fine-tuning algorithm is summarized in *Algorithm* 1. Generally, using the proposed VQ fine-tuning method, we can further improve the compression performance by up to 0.4 dB.

Algorithm 1 VQ Fine-Tuning

1: Compute $p_k = \text{softmax}(s_k)$, where s_k is the k-th row of $S = C^\top C$;
2: Initialize $q^{(0)} = [k_1, k_2, \cdots, k_J]$
3: **for** $t = 1 \rightarrow T$ **do**
4: **for** $m = 1 \rightarrow M$ **do**
5: Generate q_m by selecting new centers for each vector in Z based on $q^{(t-1)}$ and p_k using a roulette wheel selection method;
6: **end for**
7: Select the best quantization results $q^{(t)}$ from $\{q_m\}_{m=1}^{M}$ and $q^{(t-1)}$ based on the rate-distortion loss of Eq. (6);
8: $t = t + 1$
9: **end for**
10: **return** best solution $q^{(T)}$

V. EXPERIMENTAL RESULTS

In this section, we first describe the experimental setting of the proposed method and then compare the proposed method with other state-of-the-art method.

A. EXPERIMENTAL SETTING

To train the proposed VQNet based image compression method (denoted as VQNet), we construct a large image dataset. Total 156, 857 images were collected from the MS-COCO dataset [15] and the CLIC 2018 training dataset.[1] During the training, we randomly selected the training images from the constructed image dataset, from which patches of size 256 × 256 were randomly cropped to form a mini-batch. Patch argumentation with flips and rotations were adopted. The number of feature maps in the bottleneck of the encoder is set to $m = [16, 32, 64]$ for different bit-rates. We varied the regularization parameter β of Eq. (6) in the range of $[1e^{-6}, 5e^{-4}]$ to generate different bit-rates. Fig.7 shows the curve as a function of parameter β. For all the experiments, the regularization parameter λ is set as $\lambda = 1e - 5$. Other parameters of the proposed VQNet are set as, $r = 1, s = 8$, $d = 8$, and $K = 256$. For simplicity, the codes generated by the VQNet were coded by the Adaptive arithmetic coding method, while other advanced entropy coding methods can also be used. The ADAM optimizer of [11] is adopted to train the proposed network, and the network parameters were initialized by the Xavier method [10]. The learning rate was initialized as $2e^{-4}$ and dropped to $1e^{-5}$ gradually. The mini-batch size is set to 32. The proposed deep image compression network was implemented under the PyTorch platform and trained using two Nvidia TitanXP GPU, taking 2 days to converge.

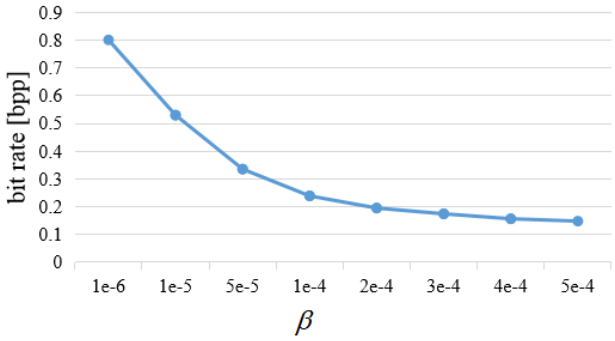


FIGURE 7. The bit rate curve as a function of parameter β.

B. ABLATION STUDY

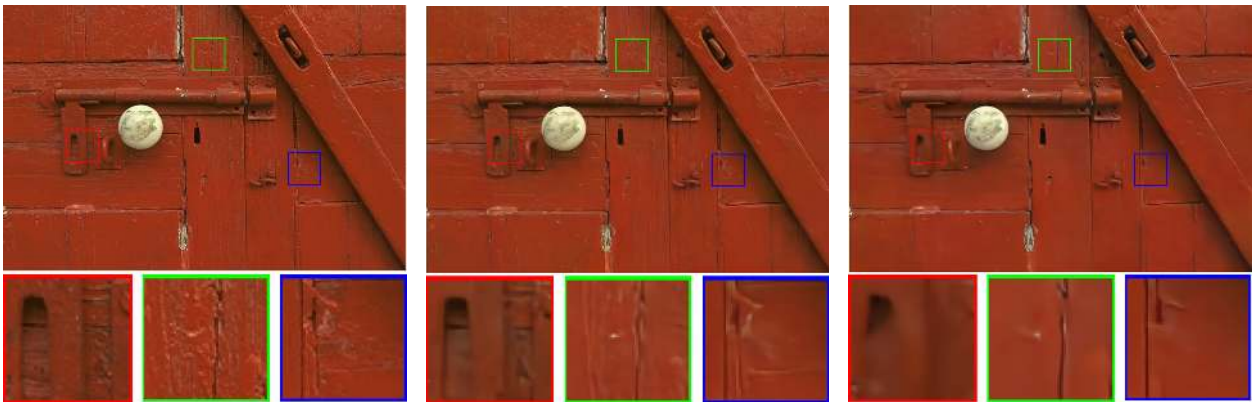
To show the effects of the fine-tuning schemes, we implemented several variants of the proposed methods, i.e., the proposed method without fine-tuning (denoted as VQNet-Baseline), the proposed method with fine-tuning of encoder (denoted as VQNet-FT-E), and the proposed method with fine-tuning of both the encoder and quantization (denoted as VQNet-FT-EQ). The commonly used Kodak dataset[2] is used as test images. And for a typical 512 × 768 input image, the fine-tuning(including patches sampling) takes about 2 minutes on a Titan XP GPU. Fig. 2 (a) shows the PSNR curves as functions of bitrates for these competing methods. From Fig. 2 (a), we can see that the VQNet-FT-E method consistently outperforms its baseline counterpart at low-bitrates. By fine-tuning both encoder and quantization, the proposed VQNet-FT-EQ further improve the compression performance.

C. COMPARISONS WITH OTHER METHODS

We also compared the proposed VQNet-FT-EQ method with several recently proposed deep learning based methods,

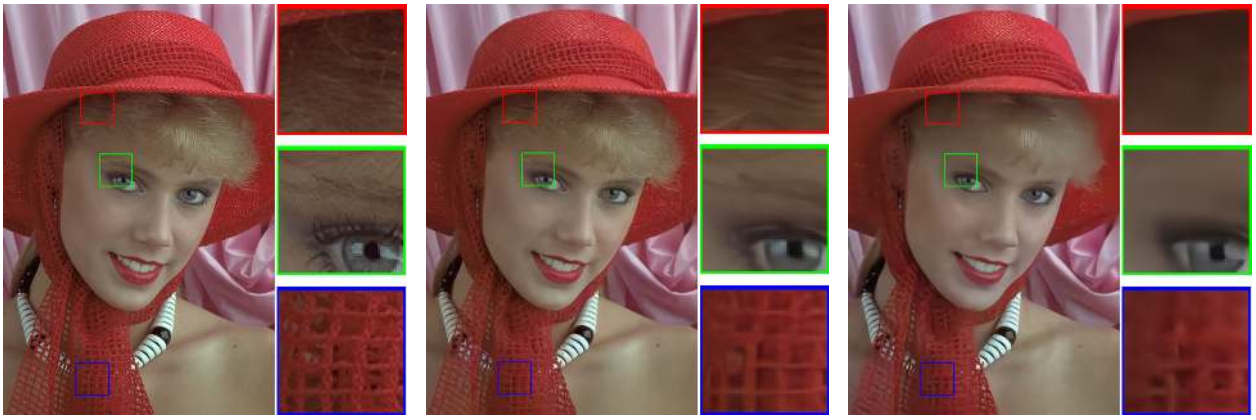
[1] http://www.compression.cc/

[2] http://r0k.us/graphics/kodak/



(a) **Original** (b) **VQNet-baseline** (0.228bpp 31.20dB 0.9398) (c) **SHVQNet-baseline** (0.231bpp 29.61dB 0.9048)

FIGURE 8. Visual comparison with [1] on the *Door* image (bits per pixel, PSNR, MS-SSIM).



(a) **Original** (b) **VQNet-baseline** (0.237bpp 32.53dB 0.9717) (c) **SHVQNet-baseline** (0.241bpp 30.78dB 0.9515)

FIGURE 9. Visual comparison with [1] on the *Girl* image (bits per pixel, PSNR, MS-SSIM).

including the soft-to-hard VQ based method (denoted as SHVQ) [1], the method by Theis *et al.* [22], Ripple's [16], Mentzer's [18] and Ballé *et al.* [2]. The traditional compression methods, i.e., JPEG, JPEG 2000 and the state-of-the-art BPG method were also compared. Both the PSNR and the MS-SSIM metrics were used to verify the performance of the test methods. For fairness, the results of other test methods were downloaded from the authors' website or directly borrowed from their papers. Fig. 2 (b) and (c) show the PSNR and MS-SSIM curves of the test methods as functions of bitrates on the Kodak dataset, respectively. It can be seen that the BPG method achieved the best rate-distortion performance in terms of PSNR. The proposed VQNet-FT-EQ performs much better than the SHVQ [1] method and the method by Theis *et al.* [22], showing the effectiveness of the proposed VQNet. The compared competitive methods [16] and [18], as the latest and best image compression algorithm, gain excellent performance form both complex network and well-designed entropy model, whereas our method uses simple Huffman coding and a concise network model. Regarding the comparisons with latest state-of-the-art methods, our method has advantages in lower bit-rates and is inferior for higher bit-rates(shown in the fig.2(b) and (c)), possibly due to the difficulties in learning centers of fine-granularity at high bit-rates. As shown in Fig. 2 (c), the proposed method is comparable and even better than [16], [18] at lower bit-rates(below 0.3bpp) in terms of MS-SSIM, indicating that our proposed quantization and finetuning are obviously effective. For higher bit rates, the performance of the proposed method is inferior to other competing methods. The reason may be that it is difficult to learn fine-grained clusters for higher bit rates. In our implementation, we quantize each feature vector $z_j \in \mathbb{R}^d$ into K=256 clusters for all bit rates. We found that the number of clusters is sufficient to represent the feature vectors and obtain good compression performance for low bit rates. For higher bit rates, larger number of clusters is needed to reduce the quantization errors of z_j. However, we found that it is rather challenging to learn finer grained clusters. We have tried to learn K=512 clusters for higher bit rates and found that this didn't improve the compression performance.

We also compare proposed method with the SHVQ [1], which also used VQ to quantize the features. In Fig. 2 (a),

TABLE 2. Average running time of a RGB image of size 768 × 512 from the Kodak dataset by the test methods.

	VQNet	VQNet-FT	JPEG	JPEG2000	BPG	Ripple's
encoding time	56.3ms	17.45s	15.3ms	310.4ms	50.1ms	8.6ms
decoding time	50.6ms	50.6ms	12.5ms	82.3ms	39.1ms	9.9ms

our experiments shows that our proposed VQNet outperforms SHVQ 0.2 to 0.3dB in PSNR(Green and purple curves). As we don't find the explict statement of the loss function of [1] for their MS-SSIM results, we show the MS-SSIM results trained under different loss functions in Fig. 2 (c)(blue curve corresponds to MSE loss, purple curve corresponds to MS-SSIM loss). Obviously, our algorithm is superior to [1] in both MSE loss and MS-SSIM loss. In order to highlight our contribution more fairly, we replace the VQ module in our algorithm with the structure of [1], then train the network with the same parameters until convergence. The experimental results show that our proposed algorithm is superior to [1]. The results of visual image contrast are shown in Figs. 8-9.

Figs. 3-6 show the parts of the decompressed images by the competing methods at bitrate around 0.3bpp. The decompressed images of [2], [22] and were downloaded from the authors' website. One can see that the images decompressed by JPEG 2000 suffer serious ringing artifacts around the edges and textures. While the visual quality of the images produced by Ballé *et al.* and the BPG method are much better than JPEG 2000, the edges and textures are tended to be oversmoothed by these two methods. Obviously, the proposed method reproduced the images with much sharper and clearer edges and textures than other methods.

Table 2 shows the encoding and decoding time of the proposed method and the other test methods, including the JPEG, JPEG 2000, BPG and the Rippel's method [18]. The running time of the other methods were borrowed from [18]. Note that JPEG, JPEG 2000 and BPG methods were implemented using the ffmpeg and libbpg softwares running on a CPU, while Rippel's method [18] was implemented on a GTX 980Ti GPU. Since the source codes of the other deep learning based methods are not available, we cannot obtain the running time of other deep learning based methods. The proposed method was implemented on a NVidia Titan XP GPU. From Table2, we can see that the proposed VQNet method runs comparable fast with the BPG method. The proposed VQNet-FT method runs much slower than the other methods, since it conducts network parameters and quantizer finetuning on the input image. However, the decoding of the proposed VQNet-FT method runs same fast as VQNet method. The proposed method may be accelerated by pruning the redundant convolutional filters of the proposed deep network.

VI. CONCLUSIONS

In this paper, we proposed a novel vector quantization network (VQNet) for image compression. In the proposed VQNet, the feature vectors are classified into one of K centers by a softmax classifier, which was trained in an unsupervised manner. After jointly training VQNet and the encoder and decoder on a large-scale dataset, we propose to further fine-tune the encoder and the codes generated by the VQNet on each input image, while keeping the decoder fixed. The fine-tuning make the proposed network adapt to local image structures, leading to substantial improvements. Experimental results show that the proposed method achieves very competitive image compression results compared to current state-of-the-art methods.

REFERENCES

[1] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1141–1151.

[2] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. ICLR*, 2017.

[3] J. Cai, Z. Cao, and L. Zhang, "Learning a single tucker decomposition network for lossy image compression with multiple bits-per-pixel rates," 2018, *arXiv:1807.03470*. [Online]. Available: https://arxiv.org/abs/1807.03470

[4] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] W. Dong, G. Shi, and J. Xu, "Adaptive nonseparable interpolation for image compression with directional wavelet transform," *IEEE Signal Process. Lett.*, vol. 15, pp. 223–236, Jan. 2008.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

[9] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE ICCV*, Dec. 2015, pp. 1026–1034.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[12] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval," in *Proc. ESANN*, 2011, p. 2.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[14] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proc. CVPR*, Jun. 2018, pp. 3214–3223.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[16] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2018, pp. 4394–4402.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[18] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. ICML*, 2017, pp. 2922–2930.

[19] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1874–1883.

[20] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.
[21] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[22] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *Proc. ICLR*, 2017.
[23] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," 2015, *arXiv:1511.06085*. [Online]. Available: https://arxiv.org/abs/1511.06085
[24] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proc. CVPR*, Jul. 2017, pp. 5435–5443.
[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[26] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, Jun. 2019.
[27] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1061–1073, Sep. 2011.
[28] M. Kalluri, M. Jiang, N. Ling, J. Zheng, and P. Zhang, "Adaptive RD optimal sparse coding with quantization for image compression," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 39–50, Jan. 2019.
[29] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Deep multiple description coding by learning scalar quantization," in *Proc. Data Compress. Conf.*, Mar. 2019, p. 615.
[30] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Multiple description convolutional neural networks for image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2494–2508, Aug. 2019.

XIAOTONG LU received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2016, where he is currently pursuing the Ph.D. degree in intelligent information processing. His research interests include deep learning, compressive sensing, and image restoration.

HENG WANG received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2017, where he is currently pursuing the master's degree in intelligent information processing. His research interests include image restoration and deep learning.

WEISHENG DONG (M'11) received the B.S. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004, and the Ph.D. degree in circuits and system from Xidian University, Xi'an, China, in 2010.

He was a Visiting Student with Microsoft Research Asia, Beijing, China, in 2006. From 2009 to 2010, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. In 2010, he joined the School of Electronic Engineering, Xidian University, as a Lecturer, where he has been a Professor, since 2016. His research interests include inverse problems in image processing, sparse signal representation, and image compression.

Dr. Dong was a recipient of the Best Paper Award at the SPIE Visual Communication and Image Processing (VCIP), in 2010. He is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.

FANGFANG WU received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2008, where she is currently pursuing the Ph.D. degree in intelligent information processing. Her research interests include compressive sensing, sparse representation, and deep learning.

ZHONGLONG ZHENG received the B.S. degree in electronic engineering from the University of Petroleum, China, and the Ph.D. degree in computer science from Shanghai Jiaotong University, China, in 1999 and 2005, respectively. From 2012 to 2013, he was a Visiting Scholar with the Department of Computer Science, University of California, Merced, CA, USA. In 2005, he joined the Department of Computer Science, Zhejiang Normal University, as a Lecturer, where he has been a Professor, since 2011. His research interests include pattern recognition and image processing.

GUANGMING SHI (SM'10) received the B.S. degree in automatic control, the M.S. degree in computer control, and the Ph.D. degree in electronic information technology from Xidian University, in 1985, 1988, and 2002, respectively.

He joined the School of Electronic Engineering, Xidian University, in 1988. From 1994 to 1996, as a Research Assistant, he cooperated with the Department of Electronic Engineering at The University of Hong Kong. Since 2003, he has been a Professor with the School of Electronic Engineering, Xidian University, and the Head of the National Instruction Base of Electrician & Electronic (NIBEE), in 2004. In 2004, he was with the Department of Electronic Engineering, University of Illinois at Urbana-Champaign (UIUC). He is currently the Deputy Director of the School of Electronic Engineering, Xidian University, and the Academic Leader in the subject of Circuits & Systems. He has authored or coauthored over 60 research papers. His research interests include compressed sensing, theory and design of multirate filter banks, image denoising, low-bit-rate image/video coding, and implementation of algorithms for intelligent signal processing (using DSP&FPGA).