IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 26, 2021, accepted June 11, 2021, date of publication June 18, 2021, date of current version July 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3090404

A Layer-Wise Extreme Network Compression for Super Resolution

JIWON HWANG, A. F. M. SHAHAB UDDIN, AND SUNG-HO BAE, (Member, IEEE)
Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea

Corresponding author: Sung-Ho Bae (shbae@khu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, Information and Communications Technology (ICT) and Future Planning under Grant 2018R1C1B3008159.

ABSTRACT Deep neural networks (DNNs) for single image super-resolution (SISR) tend to have large model size and high computation complexity to achieve promising restoration performance. Unlike image classification, model compression for SISR has rarely been studied. In this paper, we found out that DNNs for image classification and SISR have often different characteristics in terms of layer importance. That is, contrary to the DNNs for image classification, the performance of SISR networks hardly decrease even if a few layers are eliminated during inference. This is due to the fact that they typically consist of a bunch of hierarchical and complex residual connections. Based on that key observation, we propose a layer-wise extreme network compression method for SISR. The proposed method consists of: i) reinforcement learning based joint framework for layer-wise quantization and pruning both of which are effectively incorporated into the search space; ii) a progressive preserve ratio scheduling that reflects importance in each layer more effectively, yielding much higher compression efficiency. Our comprehensive experiments show that the proposed method can effectively be applied to the existing SISR networks, thus extremely reducing the model size up to 97% (i.e., 1 bit per weight on average) with marginal performance degradation compared to the corresponding full-precision models.

INDEX TERMS Single image super resolution, model compression, layer-wise, quantization, pruning, joint learning, reinforcement learning.

I. INTRODUCTION

Single Image Super Resolution (SISR) aims to reconstruct a high-resolution (HR) image from a given low-resolution (LR) image and considered as one of the representative problems in image processing. Recently, deep neural networks (DNNs) based SISR methods have widely been studied, showing that DNNs are promising solutions not only for classification problems, but also for regression problems with very high output dimensions like SISR [7], [18].

SRCNN [7] is considered as the first deep learning based SISR method which consists of three convolution layers. After then, VDSR [18], consisting of twenty convolution layers with one residual connection, was proposed, achieving much higher reconstruction performance compared to the three-layered SRCNN [7]. EDSR [25] established a deeper network having about 70 layers with multiple residual connections. Recently, RCAN [36] showed remarkable reconstruction performance by stacking over 800 layers with hierarchical and very complex residual connections. In summary, DNNs for SISR have rapidly advanced by using numerous convolution layers and building more complex architectures with multiple residual connections. As a result, despite of impressive performance, these SISR models can hardly be used in practical applications due to the large amount of memory requirement and high inference latency. As a result, model compression comes into account.

Regarding the classification tasks, model compression techniques for DNNs are attracting increased attention. Han *et al.* [10] proposed a model compression framework for DNNs, called Deep Compression, which compresses the weights in three steps: pruning, quantization and entropy coding. Since then, quantization [8], [17], [37] and pruning [12], [20], [26], [27] methods have independently and/or jointly been studied to improve compression efficiency of DNNs. In addition to the pruning and quantization, several compression methods have also been studied such as distillation [13], new efficient architecture [6], [9], [16], etc.

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

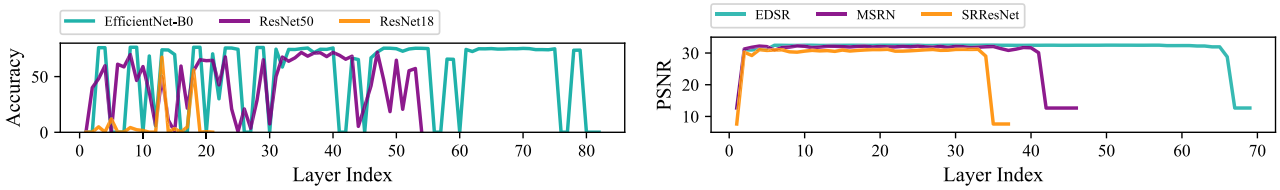


FIGURE 1. The effects of layer pruning on representative image classification networks (left) and SISR networks (right), where we measure the test performance under the condition that one layer (corresponding to the index of x-axis) in a whole network is pruned. Each image shows the ImageNet validation accuracy and SISR reconstruction performance in terms of PSNR on Set5, respectively.

In contrast, regarding the SISR as a regression task, only a few attempts for compressing the models have been taken. The BinarySR [29] is considered as the first SISR compression method that quantizes the weights of the intermediate layers in the whole model to 1 bit with trainable scale factors, thus achieving up to 5× compression ratio. Due to the binarized bit constraint, BinarySR shows moderate decrease in reconstruction performance at the expense of reducing model size and complexity. BinarySR reduces the model size and complexity at the expense of moderate decrease in reconstruction performance.

To alleviate the aforementioned problem, we propose a more efficient model compression method for SISR that can effectively be applied to the whole network where each layer is adaptively compressed based on its importance. More specifically, the proposed method is built based on reinforcement learning (RL) where quantization and pruning are effectively integrated into the search space as non-zero bit allocation and zero bit allocation cases in order to maximize compression efficiency.

The proposed method is derived from our key observation as shown in Figure 1. It shows the effects of layer pruning on representative image classification networks (left) and SISR networks (right), where we measure the test performance under the condition that one layer (corresponding to the index of x-axis) in a whole network is pruned. As shown in Figure 1, contrary to the image classification networks, SISR networks tend to be much more robust to the layer pruning, indicating that many layers in SISR networks hardly affect the performance degradation. This can make the deployment of layer-wise pruning possible, resulting in significantly enhancing the compression efficiency, especially for SISR networks.

Therefore, based on a layer-wise quantization framework used in image classification networks [33], we propose a new compression framework for SISR that jointly performs quantization and pruning in a layer-wise manner. To the best of our knowledge, we are the first to consider the layer-wise importance for SISR networks with RL. Furthermore, to maximize compression efficiency at low preserve ratio, we propose a new progressive preserve ratio scheduling (PPRS) which allows more precise model compression by considering wider dynamic range of bits in each layer. In consequence, the proposed method achieves high preserve ratio of up to 3-10% in exchange for minimum performance degradation. Our contributions are as follows:

- We propose an adaptive compression method for SISR that finds effective length of bits or performs pruning jointly for each layer, based on our key observation that SISR networks have a bunch of redundant layers which minimally affect the performance.
- We propose PPRS that makes the agent in RL find the effective length of bit with a wider range of bits in each layer. It helps to achieve more precise model compression when the target preserve ratio is extremely low.
- To verify the effectiveness, we perform comprehensive experiments with various SISR models. Our experimental results reveal that the proposed method achieves an extreme preserve ratio of up to 3% with marginal performance degradation when the target models are densely connected.

II. RELATED WORKS

A. ADVANCED SISR NETWORKS

SRCNN [7] is the first CNN based SR method that used only 3 layers to reconstruct HR image from its LR counterpart. After that, numerous deeper SISR networks have been proposed in search of better reconstruction performance [11], [18], [23], [25], [36]. VDSR [18] showed significant performance improvement by using a deeper network with skip connections. With the introduction of SRResNet [21] and EDSR [25], repeated block architectures have become popular where each block is made up of series of convolutions, activations, skip connections, etc. Thereafter, several outstanding methods have been proposed such as MSRN [23], DDBPN [11], and RCAN [36] that have promising reconstruction ability.

However, a number of repetitions of the blocks in the state-of-the-art networks result in a significantly large model size and cause over-parameterization in the models. Therefore, those models are unsuitable to be used in restricted environments like real-life applications. To overcome this problem, efficient SR models have drawn researchers' attention and smaller models have been introduced such as Memnet [32], CARN [2], CBPN [38].

B. QUANTIZATION

Instead of making smaller models, compression is an alternative solution that enables the deeper models to be applied in restricted environments. Quantization in DNNs is the method that maps full precision weights to a concise set of bits.

In general, quantization methods can be categorized into two classes i.e., non-parametric and parametric quantization methods. Regarding the non-parametric quantization (e.g., log and k-means clustering method [10]), the step size or bit-depth are predefined before training while the weights or statistical values for the weights are fine-tuned during the training step. On the other hand, parametric quantization methods [8], [17] update the weights and hyper-parameters for quantization (e.g., quantization step-size or scaling factor) in an alternative manner. In general, parametric quantization methods have shown higher compression efficiency compared to the non-parametric quantization methods. However, in this paper, we found out that parametric quantization methods cause excessive computational complexity when being used in the RL framework since a huge parameter space should be optimized simultaneously (e.g., finding effective length of bit-depths, step-sizes and weight values in each layer).

Regarding the quantization methods applied in SISR networks, Y. Ma, *et al.* [29] proposed BinarySR which applies only 1 bit to quantize the weights in the convolution layers with learnable scaling factors. However, as the binary states for a weight severely limits the precision, the binarized quantization was applied only to a few intermediate blocks in a whole network, resulting in non-optimal compression efficiency.

Some of the quantization methods compress both weight and activation such as BSRN [34], and PAMS [22]. BSRN [34] achieved good performance in a 1-bit environment through bit accumulation, and PAMS [22] improved the performance by applying distillation to the result of quantization. Contrary to the existing works in [34] and [22] we focus on the weight compression but in a more flexible way. As a result, our proposed method can effectively be applied to general SISR networks. Further, the proposed method also provides a conditioning parameter to adjust model size.

Although [22], [29], [34] paves a way of compressing SISR networks, it does not consider layer-wise importance during quantization, rather, all the layers are quantized with identical bit-depths. However, each layer may have different importance. Therefore, we advocate for applying layer-wise quantization to enhance the compression efficiency.

In [33], a layer-wise compression method, namely hardware-aware quantization (HAQ) was proposed for quantizing weights and activations depending on layer-wise importance using deep deterministic policy gradient (DDPG). Contrary to HAQ that is developed for image classification tasks, our work is extended from HAQ in order to reflect layer-wise importance in SISR model compression. It is worth nothing that directly applying HAQ to SISR networks cannot make full use of SISR networks property, since it does not perform layer-wise pruning. Moreover, we found that HAQ tends to fail in reflecting a variety of layer importance under the condition when a target preserve ratio is extremely low (e.g., 1 bit or 2 bits per weight on average), thus resulting in sub-optimal compression efficiency. We largely extend and specialize HAQ [33] by building a joint compression framework with both quantization and pruning with the proposed PPRS, which effectively remedies the aforementioned problems. It is noted that our proposed method aims to minimize the network size by focusing on reducing the bits for the weight parameters. We verify that LSQ [8], a parametric quantization method, can also effectively be incorporated into our framework as a post fine-tuning process where both weights and activations are quantized with learnable step sizes, achieving much higher compression efficiency. The detailed experimental results are shown in Section V.

C. PRUNING

Pruning is an approach to reduce the resource utilization at test time. It entails a systematic parameter removal from an existing network and produces a smaller model while maintaining the similar performance. In general, pruning methods are categorized into two classes : structured [12], [26] and unstructured [20], [27] pruning methods. The unstructured pruning aims to eliminate un-/less-important individual weights while structured one applies to specific groups, such as a set of neurons, filters, and channels, to fit resource requirements [5]. However, pruning has hardly been applied to SISR networks because existing methods assume that all channel wise features are equally important to the final reconstruction of high resolution image [14]. Also, when it comes to the image classification networks, the structured pruning in a unit of layer has rarely been studied as it can significantly degrade the performance as shown in Figure 1-(a). Contrary to the existing pruning methods, we are the first to apply layer-wise pruning with quantization in a joint optimization manner. This is based on our key observation that descent SISR networks have a bunch of redundant layers that hardly affect the performance drop as shown in Figure 1-(b).

III. METHOD

Figure 2 depicts an overview of the propose framework. Based on the agent DDPG [24], we aim to find the effective length of bits or pruning state for every layer automatically. The actor allocates the bits in each layer based on the policy under the preserve ratio constraint. The preserve ratio is obtained from PPRS that gradually decreases the preserve ratio along the episodes. After finding the effective length of bits/pruning state for each episode, it performs 1 epoch fine-tuning, then calculates the PSNR difference between the original full-precision model and current compressed model, and passes the feedback to the critic as a reward. We define the reward function as

$$\mathcal{R} = \lambda \times (PSNR_{quantized} - PSNR_{original}), \tag{1}$$

where λ is a scaling factor which is set to 0.1 by default.

A. AGENT

As in [33], DDPG [24] is selected as an RL agent, which solves continuous problems with an off-policy actor-critic algorithm. In this study, the RL agent aims to find out the

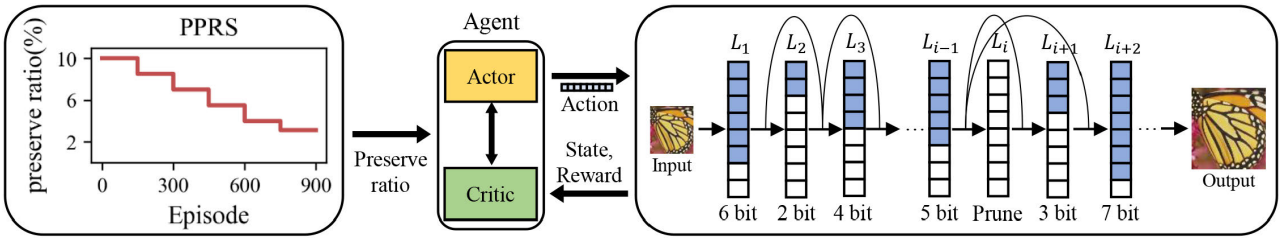


FIGURE 2. An overview of the proposed compression framework. We use DDPG [24] as an agent for RL to find the effective length of bits for every i^{th} layer (L_i) automatically. The actor performs quantization and pruning to allocate effective status in a layer-wise manner under the condition that the average bit-value fits to the predefined preserve ratio. The preserve ratio is obtained from PPRS that gradually decreases preserve ratio. After finding the bits, the PSNR difference value between the full precision original model and the compressed current model is calculated and feed-backed to the critic as a reward.

pruning probability or quantization bits for a layer in each step. And sequence of such steps for all of the layers is considered as one episode. We update the policy of agent with a variant form of Bellman's Equation used in [33]. Each transition in an episode is defined as $T_i = (S_i, a_i, \mathcal{R}, S_{i+1})$, where S_i and a_i are the state space and action for the i^{th} layer, respectively. For exploration, the Q-function parameterized by θ^Q is formulated as

$$\hat{Q}_i = \mathcal{R}_i - \mathcal{B} + \gamma \times Q(S_{i+1}, w(S_{i+1}) \mid \theta^Q), \tag{2}$$

where the baseline $\mathcal{B}$ is the average value of previous rewards, to reduce the variance of the gradient estimation. γ in Eq. 2 is the discount factor and set to 1, since we assume that the action made for each layer should contribute equally to the final result. For every i^{th} convolution layer, the state space S_i is defined as

$$S_i = (c_{\text{in}}, c_{\text{out}}, s_{\text{kernel}}, s_{\text{stride}}, s_{\text{feat}}, n_{\text{params}}, a_{i-1}, i), \tag{3}$$

where c_{in} and c_{out} are the numbers of input and output channels, respectively. In Eq. 3, s_{kernel}, s_{stride} and s_{feat} are kernel size, stride and the input feature map size of the i^{th} convolution layer consisting of n_{params} number of parameters, respectively. Each parameter in the state space is normalized into [0, 1]. $w(S_{i+1})$ in Eq. 2 equals to the action for the $i + 1^{th}$ layer.

An action a_i is the value in the range of (0, 1) taken by the agent and is used to calculate the final number of quantization bits for i^{th} layer as

$$b_i = round(b_{min} - 0.5 + a_i \times (b_{max} - b_{min} + 1)), \tag{4}$$

where b_{max} and b_{min} are the maximum and minimum numbers of bits, respectively.

As shown in Figure 1, recent SISR networks tend to be less sensitive to the layer-wise pruning. Therefore, we propose a simple yet efficient method to employ the layer-pruning by combining quantization and pruning states into the search space. It is worth noting that the layer pruning is achieved by assigning zero values to all of it's weights and selecting the state $b_i = 0$. Since the first and last layers play important roles as feature extraction and reconstruction parts [25], [36], we do not apply pruning in these layers by assigning b_{min} to 1 for these two parts. On the other hand, we use 0 as the b_{min} for the rest of the layers (i.e., the non-linear mapping part) to allow the layer-pruning. And b_{max} is set to 8 for every layer. Finally, the actor and critic networks are trained with the MSE loss of Q-function. More detailed setting of DDPG agent can be found in [33].

B. QUANTIZATION

We use a k-means clustering quantization method [10] that creates a code book with the centered weight values for clusters and updates them in every fine-tuning epoch. As the aforementioned problem in Section II-B, parametric quantization methods are hard to be used in the RL framework due to excessive computational complexity. We found that among representative non-parametric quantization methods, the k-means clustering method showed the best compression efficiency in our proposed framework.

C. PROGRESSIVE PRESERVE RATIO SCHEDULING

The goal of our search algorithm is to find the effective length of bit and pruning state under the constraint of the predefined preserve ratio. In this paper, the preserve ratio p is defined as

$$s^w = \sum_{l=1}^{n} s_l = \sum_{l=1}^{n} b_l * n_l, \tag{5}$$

$$p = s_c^w / s_o^w. \tag{6}$$

n_l is number of parameters at the l^{th} layer, b_l is found bit of l^{th} layer. We can get the specific layer's size s_l by multiplying them. Total model size s^w can be calculated by summation of every layer's s_l. So s_c^w is total size of the current compressed models and s_o^w is total size of the original 32 bit model. we can get preserve ratio p by dividing s_c^w with s_o^w. For example, quantizing all the layers into 1 bit and 2 bit equal to 0.03125 and 0.0625 in preserve ratio, respectively.

It is worth noting that the existing layer-wise quantization method in [33] tends to converge to a sub-optimal solution under the low preserve ratio condition. That is because of the short dynamic range for b_i that results in allocating a limited range of bits to the layers (See Figure 8). To remedy this problem, we propose PPRS that starts with a suitably large preserve ratio value (e.g., $p = 10\%$) and gradually decreases

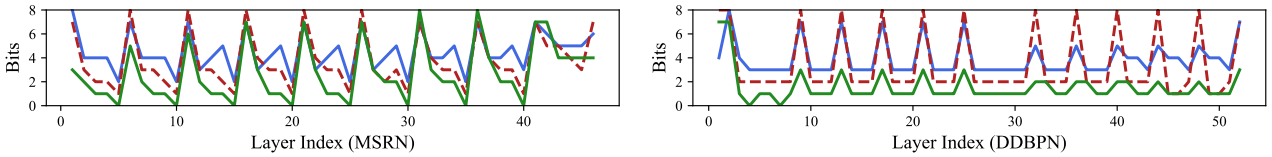


FIGURE 3. Each image represents the final bits found for the MSRN (left) [23] and DDBPN (right) [11] models, respectively. For the wide model, i.e., DDBPN, proxy model was designed to reduce the computational burden. Blue, red, green lines show the results of 10%, 6.25%, 3.125% in target preserve ratio, respectively.

it to satisfy the target preserve ratio p_{tg} during the search stage. The proposed PPRS can be formulated as

$$p^{(t)} = \max(p^{(0)} - \alpha \times \lfloor \frac{t}{\beta} \rfloor, p_{tg}), \tag{7}$$

where $p^{(0)}$ is the initial preserve ratio and $p^{(t)}$ is the preserve ratio at the t^{th} episode. In Eq. 7, α controls the reduction ratio where β determines the discrete step size. We set p_0, α and β as 0.1, 1.5 and 150, respectively. It is noted that when the target preserve ratio is high enough, it is not necessary to adjust the preserve ratio in a progressive manner. Therefore, we apply PPRS under the condition that the target preserve ratio is lower than 0.1.

PPRS was intrinsically introduced by the proposed layer-wise compression scheme and to the best of our knowledge, we firstly introduce PPRS in the context of layer-wise compression for SISR networks, which notably solves the narrow dynamic range problem at extremely low preserve ratio condition (e.g., $p = 3.125\%$). Moreover, PPRS shows better effectiveness in collaboration with the proposed layer pruning, which is verified through the ablation studies in Section V.

D. PROXY MODELS

Finding out the optimal compression parameters for significantly large SISR models [11], [25], [36] is highly time consuming and difficult. However, we found out that when the proposed compression technique is applied to the medium size SISR models [21], [23], it results in a repeated pattern for the quantization bits along the layers in non-linear mapping parts as shown in Figure 3-(a). This observation leads us to deriving a proxy model technique. The proxy model technique consists of two steps: i) building a smaller model using the basic block which comes from the original large model; ii) finding the effective length of bits for the smaller model based on the proposed method and transfers the searched block from the smaller model to the original one by repeating the found bits. This approach significantly reduces the computational burdens. An example of using proxy model is shown in Figure 3-(b). Our proxy model technique can especially be useful for large models [11], [25], [36] that have plenty of layers and/or several millions of parameters. That is, we find the effective length of bits for blocks by building the proxy models of EDSR [25], DDBPN [11] and RCAN [36], using only 2%, 53% and 12% of the original resources, respectively.

IV. EXPERIMENTS

A. MODELS AND DATASETS

To verify the effectiveness of the proposed compression method, we perform experiments on the five SISR methods. We found out that the complexity of residual connections plays a crucial role in compression efficiency of the proposed method. Therefore, we divide SISR networks under test into two folds in terms of the complexity in residual connections: i) simple models (SRResNet [21], EDSR [25]); and ii) densely connected models (MSRN [23], DDBPN [11], RCAN [36]).

The pre-trained models for EDSR, MSRN and RCAN are taken from their official github pages. The other methods i.e., SRResNet, DDBPN are trained on the DIV2K dataset [1] and are used as baseline models. The DIV2K dataset contains 800 low and high resolution image pairs for training and 100 pairs for validation. In our experiments, we use the whole training set in the DIV2K dataset to search the effective length of bits and fine-tune the weights. For testing, we use the standard SISR datasets i.e., Set5 [4], Set14 [35], BSD100 [30], and Urban100 [15].

B. SETUPS

After finding the final effective length of bits by searching for 900 episodes, the compressed SISR models are fine-tuned for 200 epochs with Adam optimizer [19] and cosine learning rate decay [28]. We use the initial learning rate of 10^{-4} except RCAN that used the learning rate of 10^{-5}. It is noted that due to a excessively huge number of layers and complex hierarchical residual connections, RCAN tends to suffer from gradient exploding when using the learning rate of 10^{-4}.

For a better training process, we use standard data augmentation techniques following the baseline SISR methods, includes horizontal and vertical flips and random rotation of 90, 180 and 270 degrees [25]. We mainly tested on the scale 4 by default, except the experiments for comparing with the state-of-the-art compression method for SISR [29]. The batch size is set to 16, and the sizes of patches are 192×192 and 96×96 for scale 4 and scale 2, respectively. To evaluate the reconstruction performance of the SISR models, PSNR and SSIM are measured on the Y channel as in [25].

TABLE 1. The performance comparison of our proposed method with the full-precision baseline models and fixed-bit quantized models. The three target compression rates of 3, 2 and 1 bits are correspond to 10%, 6.25% and 3.125% in preserve ratio, respectively.

Models	Preserve Ratio	Set5 PSNR / SSIM	Set14 PSNR / SSIM	B100 PSNR / SSIM	Urban100 PSNR / SSIM
MSRN [23]	Baseline	32.26 / 0.8960	28.63 / 0.7836	27.61 / 0.7380	26.22 / 0.7911
	Fixed 3bit	31.15 / 0.8734	27.95 / 0.7652	27.21 / 0.7236	25.27 / 0.7563
	Fixed 2bit	27.54 / 0.7778	25.54 / 0.6779	25.60 / 0.6441	22.81 / 0.6315
	Ours 3bit	**32.17 / 0.8947**	**28.62 / 0.7823**	**27.59 / 0.7367**	**26.09 / 0.7869**
	Ours 2bit	**31.97 / 0.8916**	**28.51 / 0.7793**	**27.52 / 0.7340**	**25.87 / 0.7792**
	Ours 1bit	**31.60 / 0.8856**	**28.19 / 0.7714**	**27.30 / 0.7266**	**25.29 / 0.7580**
DDBPN [11]	Baseline	32.25 / 0.8957	28.68 / 0.7840	27.64 / 0.7380	26.31 / 0.7916
	Fixed 3bit	32.05 / 0.8911	28.59 / 0.7803	27.58 / 0.7356	26.12 / 0.7857
	Fixed 2bit	31.31 / 0.8776	28.11 / 0.7678	27.27 / 0.7250	25.36 / 0.7595
	Ours 3bit	**32.21 / 0.8947**	**28.66 / 0.7830**	**27.62 / 0.7371**	**26.24 / 0.7896**
	Ours 2bit	**32.00 / 0.8918**	**28.55 / 0.7800**	**27.55 / 0.7346**	**25.98 / 0.7819**
	Ours 1bit	**31.42 / 0.8806**	**28.15 / 0.7682**	**27.28 / 0.7244**	**25.32 / 0.7567**
RCAN [36]	Baseline	32.63 / 0.9002	28.87 / 0.7889	27.77 / 0.7436	26.82 / 0.8087
	Fixed 3bit	30.33 / 0.8546	27.39 / 0.7410	26.90 / 0.7089	24.69 / 0.7317
	Fixed 2bit	26.90 / 0.7625	25.03 / 0.6565	25.13 / 0.6271	22.14 / 0.6045
	Ours 3bit	**32.41 / 0.8967**	**28.69 / 0.7849**	**27.67 / 0.7400**	**26.47 / 0.7988**
	Ours 2bit	**31.80 / 0.8870**	**28.32 / 0.7745**	**27.39 / 0.7294**	**25.66 / 0.7695**
	Ours 1bit	**30.56 / 0.8625**	**27.52 / 0.7532**	**26.90 / 0.7119**	**24.49 / 0.7227**

Models	Preserve Ratio	Set5 PSNR / SSIM	Set14 PSNR / SSIM	B100 PSNR / SSIM	Urban100 PSNR / SSIM
SRResNet [21]	Baseline	31.40 / 0.8826	28.06 / 0.7687	27.24 / 0.7256	25.09 / 0.7502
	Fixed 3bit	29.02 / 0.7955	26.35 / 0.6882	26.14 / 0.6546	23.60 / 0.6356
	Fixed 2bit	27.76 / 0.7207	25.58 / 0.6349	25.75 / 0.6310	22.98 / 0.5985
	Ours 3bit	**31.14 / 0.8768**	**27.88 / 0.7636**	**27.15 / 0.7217**	**24.91 / 0.7420**
	Ours 2bit	**30.39 / 0.8604**	**27.42 / 0.7517**	**26.87 / 0.7116**	**24.43 / 0.7205**
EDSR [25]	Baseline	32.46 / 0.8968	28.80 / 0.7876	27.71 / 0.7420	26.64 / 0.8033
	Fixed 3bit	28.86 / 0.8192	26.40 / 0.7189	26.22 / 0.6837	23.43 / 0.6733
	Fixed 2bit	27.93 / 0.7786	25.79 / 0.6828	25.84 / 0.6548	23.09 / 0.6385
	Ours 3bit	**32.22 / 0.8948**	**28.68 / 0.7849**	**27.65 / 0.7395**	**26.38 / 0.7962**
	Ours 2bit	**30.89 / 0.8664**	**27.88 / 0.7614**	**27.17 / 0.7206**	**25.25 / 0.7524**

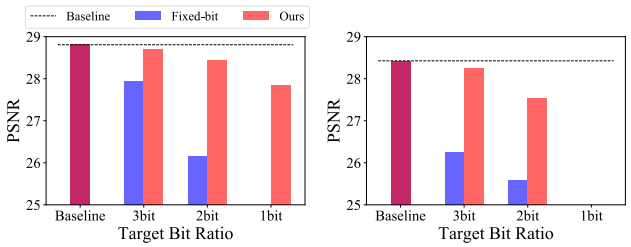


FIGURE 4. Statistical summary of the experimental results in table 1. Left and right images show densely connected models and simple models, respectively.

We implemented the proposed method with PyTorch library and use a single RTX 2080 Ti GPU having 12 GB VRAM for training and inference. When designing proxy models, we considered training time to be approximately 1 day for finding bits in a model. For fine-tuning, it takes around 80% less time than training time in each model. But for big models that require much memory occupation (e.g., RCAN, DDBPN and EDSR), we used 3 or 4 GPUs for fine-tuning the final model.

C. COMPARISON WITH FIXED QUANTIZATION

Table 1 shows the performance of the full-precision baseline models, fixed-bit quantized models and the models compressed by the proposed method under the the three target compression rates of 3, 2 and 1 bits which correspond to 10%, 6.25% and 3.125% in preserve ratio, respectively. Note that the fixed-bit quantized model is defined as a quantized model with a k-means clustering hereafter. We do not include the experimental results for the fixed quantized models at the 1 bit rate because their results were significantly low.

As shown in Table 1, the proposed method shows considerably higher compression efficiency compared to the fixed-quantized models. This is because the proposed method reflects the layer importance, thus adaptively quantizing and/or pruning each layer. Also, compared to the simple models (SRResNet and EDSR), the densely connected models (MSRN, DDBPM, and RCAN) yield higher compression

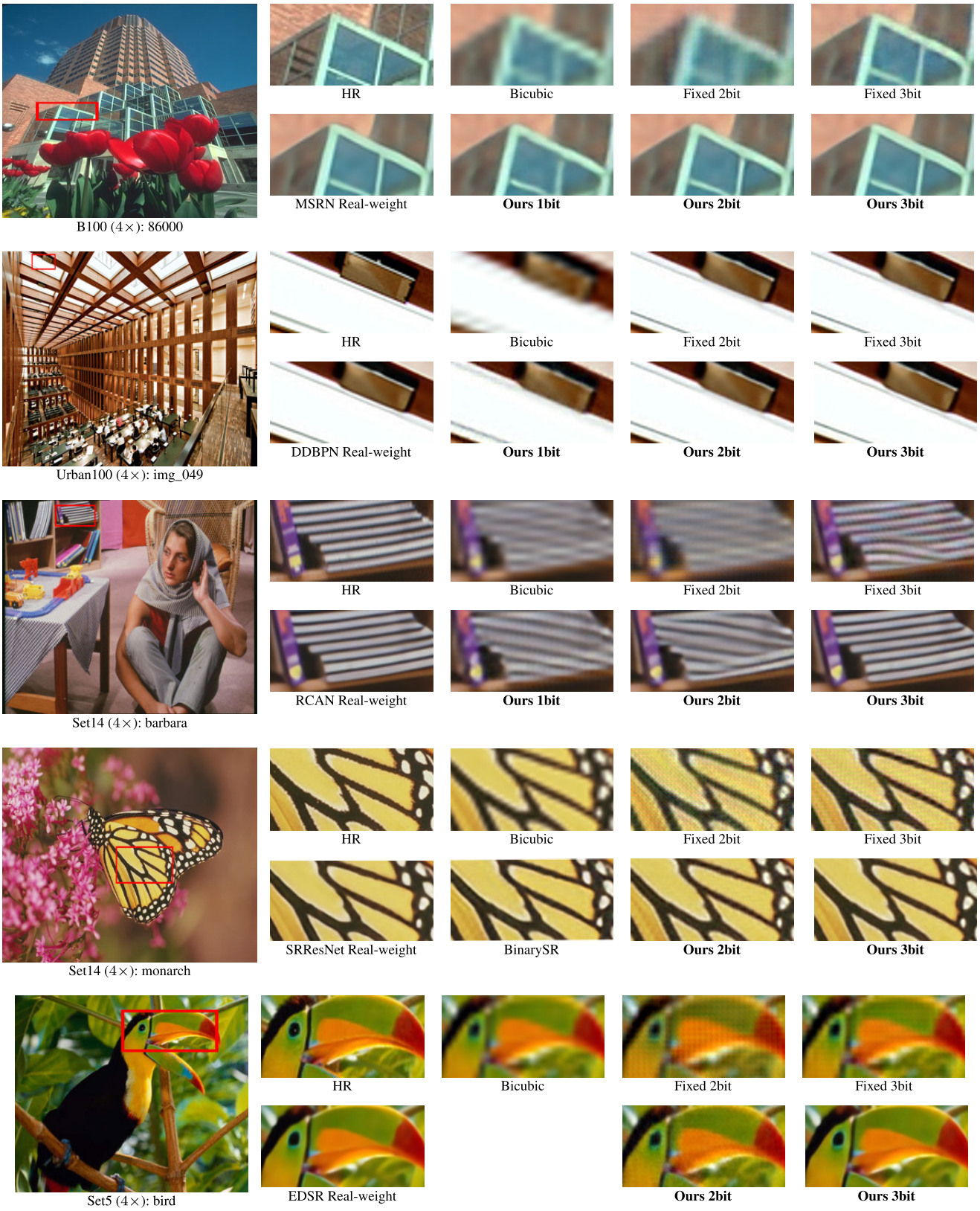


FIGURE 5. Each image compares the proposed method with several SISR methods i.e., bicubic upsampling, real-weight models [11], [21], [23], [25], [36], model quantized with fixed-bit quantization [10]. From the top, the results of MSRN [23], DDBPN [11], RCAN [36], SRResNet [21], and EDSR [25] are presented, respectively. For SRResNet [21], we compare the result with BinarySR [29]. Ours 3, 2 and 1 bit are correspond to 10%, 6.25% and 3.125% in preserve ratio, respectively.

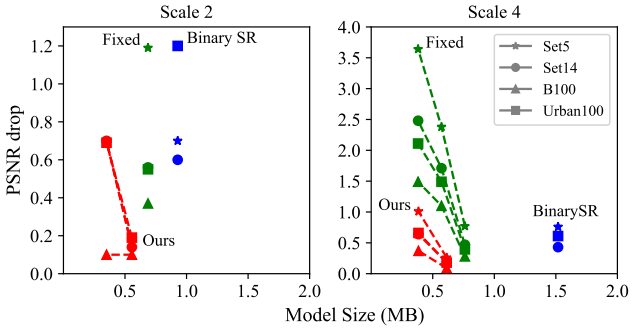


FIGURE 6. The PSNR difference with the full-precision model of SRResNet [21] at scale 2 and 4 under the comparison with the three quantization methods, i.e., ours, BinarySR [29], and fixed-bit [10]. Red, green and blue colors indicate the performance of ours, fixed-bit quantization, and BinanrySR, respectively. The star-, round-, triangle- and rectangle-shapes indicate the models tested on Set5, Set14, B100, Urban100, respectively.

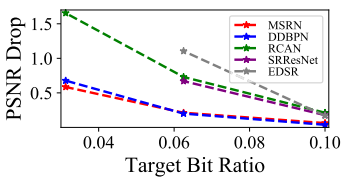
efficiency i.e., the densely connected models have lower performance drop at the same compression ratio. This implies that densely connected layers contain more redundant layers due to plenty of residual connections. It is worth noting that the state-of-the-art SISR networks gain higher performance by adopting more complex and hierarchical residual connections [3], [11], [31], [36]. Therefore, we expect that the proposed method can effectively be applied to the future SISR networks.

For better understanding, we provide the statistical summary of the experimental results in Table 1 at Figure 4. Each bar represents the average PSNR on all the test datasets i.e., Set5 [4], Set14 [35], BSD100 [30], and Urban100 [15] over all test models for the target bit ratio. The pink and blue bars in Figure 4 indicate PSNR values obtained by the proposed method and fixed quantization, respectively. Figure 4-(a) and (b) indicate the average PSNR of densely connected models [11], [23], [36] and simple models [21], [25], respectively. It is noted that some experiments with 1 bit target were not carried out thus the corresponding bars are left blank in the figure. As shown in Figure 4, it clearly shows that the proposed method offers significant performance improvement over fixed bit quantization in all target bit ratios. Figure 5 presents the visual results comparison of fixed-bit and proposed quantization method based on diverse models. As shown in Figure 5, the proposed method preserves high visual quality even under the extremely low bit condition.

D. COMPARISON WITH STATE-OF-THE-ART COMPRESSION METHOD

Additionally, we compare our proposed method with BinarySR [29] at scale 2 and 4, by means of model size and reconstruction performance. For comparison, SRResNet is used as the baseline model as in [29].

Figure 6 shows the performance degradation in PSNR between the proposed method (2, 3 bits), fixed-bit quantization (2, 3, 4 bits) and BinarySR [29]. For scale 2, the fixed



Model	(1)	(2)	(3)
MSRN	6.08M	2	3
DDBPN	10.43M	2	2
RCAN	15.63M	3	2
SRResNet	1.55M	2	1
EDSR	43.68M	2	1

FIGURE 7. Left shows the average PSNR drops along target bit ratios on four benchmark datasets for five SISR models, and right shows each model's number of (1) parameters, (2) residual hierarchies and (3) the number of redundant connections that is counted after keeping the block as a single path on removal. Each value is based on the scale 4 model.

TABLE 2. Effects of layer-wise quantization (LQ), layer-wise pruning (LP) and PPRS. Experiments are performed on Set5 (4×).

LQ	LP	PPRS	$p = 6.25\%$	$p = 3.125\%$
			PSNR / SSIM	PSNR / SSIM
✗	✗	✗	27.54 / 0.7778	24.15 / 0.5923
✓	✗	✗	31.56 / 0.8837	24.15 / 0.5923
✓	✓	✗	**31.97 / 0.8917**	29.87 / 0.8453
✓	✗	✓	31.68 / 0.8883	24.15 / 0.5923
✓	✓	✓	**31.97** / 0.8916	**31.60 / 0.8856**

2 and 3 bit were not compared in the graph because the difference in PSNR was 5dB or more. As shown in Figure 6, the proposed method achieves remarkable compression rate while keeping outstanding reconstruction performance at all target bit rates. Compared to the proposed method, BinarySR [29] allows to use only 1 bit for weight which may significantly hinder flowing the information along the layers. On the other hand, proposed method adaptively allocates the bits in each layer with respect to minimizing the performance degradation. The fourth row of Figure 5 shows the comparison of visual quality with our method and BinarySR [29], resulting in the proposed method achieves better quality with higher compression rate.

V. ANALYSIS AND ABLATION STUDY

A. EFFECT OF COMPLEX RESIDUAL CONNECTIONS ON COMPRESSION

As extension of Fig 4, we analyze the relation between compression performance and the complexity of the residual connections in Fig 7. As shown in the left figure of Fig 7, DDBPN [11] and MSRN [23] show better trade-off in bit-rate and PSNR curves compared to other models. According to the right table in Fig 7, it turns out that the compression efficiency of the proposed method is not correlated with the model size but related with the number of redundant connections in a residual block. That is, the proposed method can yield higher compression efficiency when more redundant residual connects exist in models. This analysis supports our key observation in Fig. 1 and delivers important insight of design principles for SR networks.

B. EFFECTIVENESS OF INDIVIDUAL COMPONENT

The propose method consists of three components, i.e., layer-wise quantization (LQ), layer-wise pruning (LP)

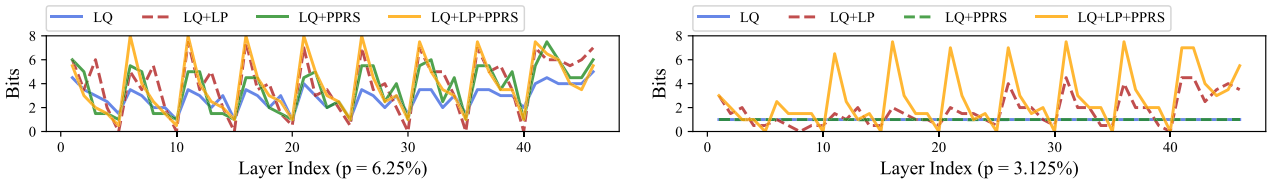


FIGURE 8. The found bits for each layer in MSRN [23] depending on the combination of layer quantization (LQ), layer pruning (LP) and PPRS under the moderate ($p = 6.25\%$) and low ($p = 3.125\%$) preserve ratio condition.

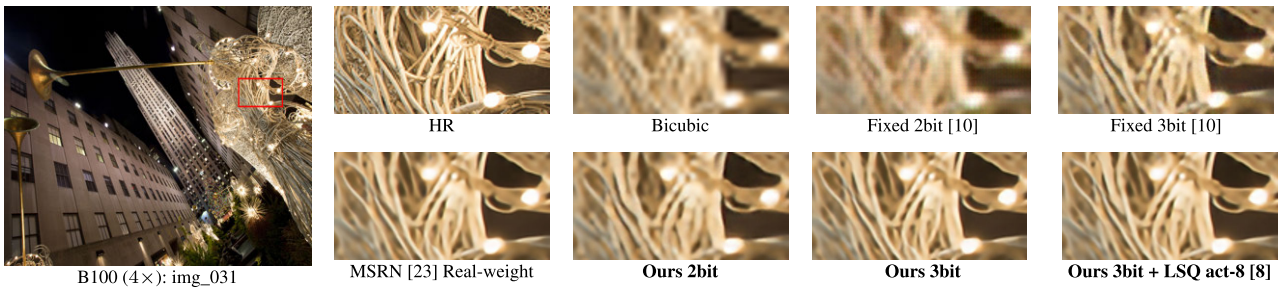


FIGURE 9. Each image compares the result of several SISR results i.e., bicubic upsampling, real-weight models (32 bits per weight) [23], the models quantized with fixed-bit quantization (3bits per weight) [10], ours (3bits per weight) and ours with activation quantization (LSQ) [8] (3 and 8 bits per weight and activation, respectively). The 3, 2 and 1 bits correspond to 10%, 6.25% and 3.125% in preserve ratio, respectively.

and PPRS. To verify the effectiveness of each component, we conduct an ablation study. Table 2 presents the performance for MSRN [23] on the Set5 dataset in combination of the three components and also without applying any of the components i.e., fixed-bit quantization. As shown in Table 2, LQ dramatically improves the compression efficiency compared to the fixed quantization method. Also, LP and PPRS work in a synergistic way, considerably improving the compression efficiency when the target preserve ratio is extremely low (i.e., $p = 3.125\%$).

For further analysis, in Figure 8, we show the found bits for each layer in MSRN [23],depending on the combination of LQ, LP and PPRS. As shown in Figure 8, compared to the only LQ case, combining LP and PPRS into LQ noticeably widen the dynamic range of the effective length of bits in each layer. This implies that the proposed method can effectively consider layer importance and utilize the characteristics of SISR networks which were observed in Figure 1. It is also noted that, especially under the very low preserve ratio condition (i.e., $p = 3.125\%$), LP is absolutely necessary to find non-1 bits. Furthermore, PPRS plays a more important role to increase the range of found bits. That is, PPRS widens the difference in effective length of bits for each layer.

C. EFFECT OF PARAMETRIC QUANTIZATION

This section verifies the validity and extendability for the proposed method with parametric quantization methods, by which the both weights and activations in a model can effectively be compressed. Table 3 shows the experimental results on MSRN ×4 [23] with proposed and LSQ quantization methods for 2 and 3 bits. For this, LSQ and our methods perform quantization for the weights during training given the target bits (2 and 3 bits). As a result, the proposed method shows higher or identical PSNR performance compared to LSQ.

Furthermore, we perform experiments to see the extension ability of the proposed method with parametric quantization methods to further compress both weights and activations. The 'Ours+LSQ 3 bit' in Table 3 shows the PSNR performance on the model that is trained with our method and LSQ. For this, the detailed procedure is as follows: in the first step, we search for the weight bits using the proposed method and freeze the found bits in each layer; in the second step, we apply LSQ to the model during fine-tuning where we set the target bits for weights as found in the first step and set the target bits for activations to 8.

As shown in Table 3 the proposed method equipped with LSQ takes great advantages of both flexible bit allocation from ours and the dynamic quantization for weights and activations from LSQ, resulting in higher or comparable PSNR values compared to the others even it additionally quantize activations to 8 bits. This indicates that the proposed method can effectively be extended with parameteric quantization methods which were not able to be applied during training due to excessive amount of computation complexity.

Figure 9 shows qualitative comparison between the original proposed method and the proposed method with LSQ. As shown in Figure 9, incorporating LSQ into the proposed method can further compress the models while hardly degrading the visual quality of the reconstructed image.

D. FOUND BITS OF THE RESULTING COMPRESSED MODELS

Our method dynamically prunes a layer in suitable condition, and it is found that most layer pruning occurs when there

TABLE 3. Extension ability of the proposed method by adopting parametric quantization (LSQ [8]) in a post-quantization process. The experiments are performed with MSRN×4 [23] on Set5, Set14, B100 and Urban100. Each value indicates PSNR for each dataset. LSQ and ours with 2 and 3bit indicate results of LSQ [8] and the proposed method with weight bits of 2 and 3 (keeping 32bits for activation). Ours + LSQ 3bit indicates result of applying LSQ [8] with found bits with our proposed methods at 10%(3-bit) preserve ratio with using 8bit for activation. Bold values indicate the best PSNR among same datasets and bits, except for baseline.

	Baseline	LSQ 2bit	Ours 2bit	LSQ 3bit	Ours 3bit	Ours + LSQ 3bit
Set5	32.26	**31.97**	**31.97**	32.04	32.17	**32.21**
Set14	28.63	28.43	**28.51**	28.52	**28.62**	28.58
B100	27.61	27.47	**27.52**	27.50	**27.59**	27.58
Urban100	26.22	25.78	**25.87**	25.90	**26.09**	**26.09**

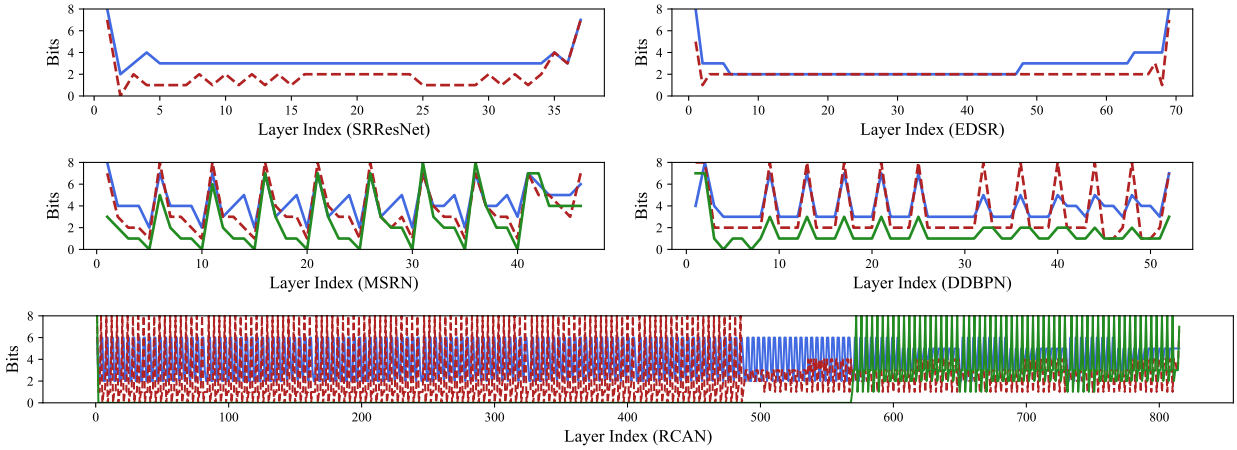


FIGURE 10. Each figure represents the final bits used for SRResNet [21], EDSR [25], MSRN [23], DDBPN [11] and RCAN [36] of scale 4 (from left to right, first line to last line). Blue, red, green lines indicate 10%, 6.25%, 3.125% in preserve ratio, respectively.

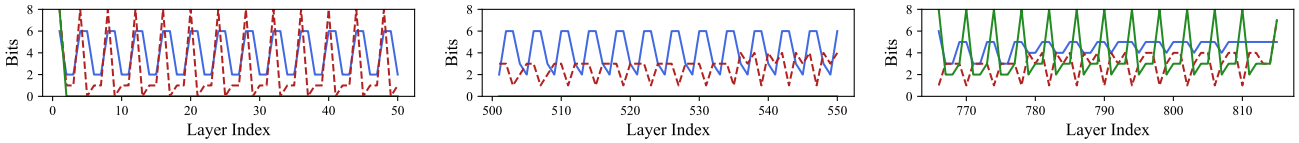


FIGURE 11. Zooming in the three different patterns for bit allication in RCAN [36] result at figure 10. (a), (b) and (c) indicate layer indices of 0-50, 500-550, and 765-815, respectively.

are short skip connections between layers within a block. As a result, even if a certain layers are pruned, other layers still convey information due to the dense connections. Consequently, block redundancy rarely occurs in which the entire specific block becomes redundant.

Figure 10 visualizes the found bits of the resulting compressed models. Unlike simple models (i.e., SRResNet [21], EDSR [25]) that have similar importance between layers, densely connected models (i.e., MSRN [23], DDBPN [11], RCAN [36]) have significant difference between layers. This makes possible to compress these models more efficiently with our proposed method. Since RCAN [36] has more than 800 layers, three sections with different bit patterns are dissected and separately shown in Figure 11. As shown in Figure 11, most forward and intermediate layers are pruned when the preserve ratio is extremely low (e.g., $p = 3.125\%$) as the green lines in the first and second figures often pass 'zero' bits. This implies that the proposed method can also effectively be applied to the networks with a huge number of layers.

E. LIMITATION AND FUTURE WORK

In this paper, we use a simple quantization method, i.e., a vector quantization with a k-means clustering method in the context of layer-wise adaptive quantization under the reinforcement learning condition. Although the simple k-means quantization achieves high performance, it can be further improved by adopting more sophisticated quantization methods (e.g., LSQ [8]). As such state-of-the-art quantization methods are based on learnable quantization parameters during training, they cause a significantly huge computation complexity and training when being used in the reinforcement learning scenario. Therefore, it can be further studied how to incorporate the state-of-the-art quantization methods into the proposed layer-wise compression framework in a light-weight manner.

VI. CONCLUSION

Based on the newly discovered behavior in layer-wise pruning for SISR networks, we introduce an effective and powerful layer-wise compression technique for SISR networks that can adaptively find out the optimal compressed models to meet the predefined compression criterion. The proposed method applies best-suited quantization bits or pruning to each layer based on its importance to the final reconstruction quality. Additionally, in order to increase performance by enabling convergence at extremely low preserve ratio, we propose PPRS that gradually reduces the preserve ratio as the episode passes. The comprehensive experimental results verify the effectiveness of the proposed compression method i.e., it can achieve notable compression rate with promising reconstruction performance.

In summary, the proposed method reveals that considering the characteristics of target network is essential during compression to maximally improving the compression efficiency as well as resulting model performance.

ACKNOWLEDGMENT

The authors would like to express their deepest appreciation to Subin Yang for running the experiments, drawing the experimental figures, and reviewing the paper.

REFERENCES

[1] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.

[2] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 252–268.

[3] S. Anwar and N. Barnes, "Densely residual Laplacian super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 2, 2020, doi: 10.1109/TPAMI.2020.3021088.

[4] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L.-A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–10.

[5] D. Blalock, J. Javier Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" 2020, *arXiv:2003.03033*. [Online]. Available: http://arxiv.org/abs/2003.03033

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[8] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," 2019, *arXiv:1902.08153*. [Online]. Available: http://arxiv.org/abs/1902.08153

[9] Q. Guo, Z. Yu, Y. Wu, D. Liang, H. Qin, and J. Yan, "Dynamic recursive neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5147–5156.

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: http://arxiv.org/abs/1510.00149

[11] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1664–1673.

[12] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4340–4349.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[14] Z. Hou and S.-Y. Kung, "Efficient image super resolution via channel discriminative deep neural network pruning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3647–3651.

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

[16] Y. Jeon and J. Kim, "Constructing fast network through deconstruction of convolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5951–5961.

[17] S. Jung, C. Son, S. Lee, J. Son, J.-J. Han, Y. Kwak, S. J. Hwang, and C. Choi, "Learning to quantize deep networks by optimizing quantization intervals with task loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4350–4359.

[18] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, Jun. 2016, pp. 1646–1654.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[20] A. Kusupati, V. Ramanujan, R. Somani, M. Wortsman, P. Jain, S. Kakade, and A. Farhadi, "Soft threshold weight reparameterization for learnable sparsity," 2020, *arXiv:2002.03231*. [Online]. Available: http://arxiv.org/abs/2002.03231

[21] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[22] H. Li, C. Yan, S. Lin, X. Zheng, Y. Li, B. Zhang, F. Yang, and R. Ji, "PAMS: Quantized super-resolution via parameterized max scale," 2020, *arXiv:2011.04212*. [Online]. Available: http://arxiv.org/abs/2011.04212

[23] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 517–532.

[24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[26] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, "HRank: Filter pruning using high-rank feature map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1529–1538.

[27] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, and M. Jaggi, "Dynamic model pruning with feedback," 2020, *arXiv:2006.07253*. [Online]. Available: http://arxiv.org/abs/2006.07253

[28] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*. [Online]. Available: http://arxiv.org/abs/1608.03983

[29] Y. Ma, H. Xiong, Z. Hu, and L. Ma, "Efficient super resolution using binarized neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.

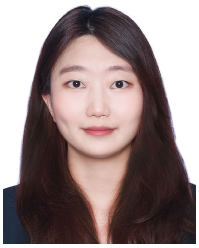
[31] Y. Qiu, R. Wang, D. Tao, and J. Cheng, "Embedded block residual network: A recursive restoration model for single-image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4180–4189.

[32] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.

[33] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "HAQ: hardware-aware automated quantization with mixed precision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8612–8620.

[34] J. Xin, N. Wang, X. Jiang, J. Li, H. Huang, and X. Gao, "Binarized neural network for single image super resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 91–107.

[35] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surfaces*. Berlin, Germany: Springer, Jun. 2010, pp. 711–730.
[36] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.
[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2016, *arXiv:1606.06160*. [Online]. Available: http://arxiv.org/abs/1606.06160
[38] F. Zhu and Q. Zhao, "Efficient single image super-resolution via hybrid residual feature learning with compact back-projection network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019.



JIWON HWANG received the bachelor's degree from the Department of Computer Science and Engineering, Kyung Hee University, South Korea, in 2019, where she is currently pursuing the M.S. degree. Her research interests include single image super resolution and deep learning model compression.

A. F. M. SHAHAB UDDIN received the B.S. and M.S. degrees from the Department of Information and Communication Engineering, Islamic University, Bangladesh, in 2015 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, Suwon, South Korea. His research interests include general problems in machine learning, image quality assessment, perceptual image processing, and inverse problems in image processing.

SUNG-HO BAE (Member, IEEE) received the B.S. degree from Kyung Hee University, South Korea, in 2011, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Postdoctoral Associate with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Since 2017, he has been an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University. He has been involved in model compression/interpretation for deep neural networks and inverse problems in image processing and computer vision.