Fully Quantized Image Super-Resolution Networks

Hu Wang, Peng Chen, Bohan Zhuang, Chunhua Shen*

The University of Adelaide, Australia   Monash University, Australia   Amazon Australia

ABSTRACT

With the rising popularity of intelligent mobile devices, it is of great practical significance to develop accurate, real-time and energy-efficient image Super-Resolution (SR) methods. A prevailing method for improving the inference efficiency is model quantization, which allows for replacing the expensive floating-point operations with efficient bitwise arithmetic operations. To date, it is still challenging for quantized SR frameworks to deliver a feasible accuracy-efficiency trade-off. Here, we propose a Fully Quantized image Super-Resolution framework (FQSR) to jointly optimize efficiency and accuracy. In particular, we target obtaining end-to-end quantized models for all layers, especially including skip connections, which was rarely addressed in the literature of SR quantization. We further identify obstacles faced by low-bit SR networks and propose a novel method to counteract them accordingly. The difficulties are caused by 1) for SR task, due to the existence of skip connections, high-resolution feature maps would occupy a huge amount of memory spaces; 2) activation and weight distributions being vastly distinctive in different layers; 3) the inaccurate approximation of the quantization. We apply our quantization scheme on multiple mainstream super-resolution architectures, including SRResNet [16], SRGAN [16] and EDSR [17]. Experimental results show that our FQSR with low-bits quantization is able to achieve on par performance compared with the full-precision counterparts on five benchmark datasets and surpass the state-of-the-art quantized SR methods with significantly reduced computational cost and memory consumption. Code is available at: https://git.io/JWxPp

CCS CONCEPTS

• **Computing methodologies** → **Computer Vision**; • **Image processing**; • **Reconstruction**;

KEYWORDS

Image Super-resolution, Network Quantization

*The first two authors contributed equally. Work was done when all authors were with The University of Adelaide.

1 INTRODUCTION

The rapid development of Deep Convolutional Neural Networks (CNNs) has led to significant breakthroughs in image super-resolution, which aims to generate high-resolution images from low-resolution inputs. For real-world applications, the inference of SR is usually executed on edge devices, such as High-Definition televisions, mobile phones or drones, which require real-time, low-power consumption. However, the high computational cost of CNNs prohibits the deployment of SR models to resource-constrained devices.

To improve computation and memory efficiency, various solutions have been proposed in the literature, including network pruning [20, 39], low-rank decomposition [34], network quantization [21, 37] and efficient architecture design [11, 18, 33]. In this work, we aim to train a low-precision SR network, including *all layers and skip connections*. Although current quantization methods have achieved promising performance on the image classification task, training quantized models for pixel-wise dense prediction tasks such as super-resolution, still remains a challenge in terms of the unverifiable efficiency improvement on hardware and the severe accuracy degradation. For example, to our knowledge, existing quantized SR models typically keep the skip connections to be full-precision, resulting in huge memory consumption occupied by high-resolution feature maps and making it impractical to be deployed. In this paper, we introduce the Fully Quantized Image Super-Resolution Networks (FQSR), to yield a promising efficiency-versus-accuracy trade-off.

Typically, a common SR network consists of a feature extraction module, a nonlinear mapping module and an image reconstruction module [30]. Recently, various quantized SR methods [23, 30] leverage binary quantization on the non-linear mapping module of the SR network, while paying less attention to the quantization of the feature extraction and image reconstruction modules. However, we observe that the feature extraction and reconstruction modules also account for significant computational cost during inference (*e.g.*, with respect to ×2 up-scaling models, these two sub-modules occupy 15.6% of total computational FLOPs for SRResNet model and 11.4% for the EDSR model; in ×4 up-scaling, these two sub-modules occupy 45.1% and 38.7% of total computational FLOPs for SRResNet and EDSR, respectively). Therefore, it is essential to quantize all three sub-modules to obtain more compact models. Additionally, in the SR task, the feature dimensions are usually very high. These features will occupy a huge amount of memories, especially when skip connections exist in the network which requires multiple copies of the tensors. Thus, with the quantization of the skip connections, the memory consumption can be saved dramatically (by approximately 8× when compared to the full-precision counterparts). In this paper, we propose to quantize all layers in SR networks to reduce the burden of computation and storage starving SR tasks on resource-limited platforms.

In addition to the fully quantized design, we further introduce specific modifications with respect to the quantization algorithm

for super-resolution. In particular, we empirically observe that the data distributions of the activations and weights of different layers differ *drastically* for the SR task. For the distribution with a small value range, the corresponding quantization interval should be sufficiently compact in order to maintain appropriate quantization resolution. On the other hand, if the quantization interval is too compact for distribution with a large value range, it may cause severe information loss. Therefore, we propose to learn quantizers that can find the optimal quantization intervals that minimize the task loss. Furthermore, we also observe that the categorical distribution of quantized values may not fit the original distribution in some layers during training. Thus, we propose a quantization-aware calibration loss to encourage the minimization of the distribution difference.

Our contributions are summarized as follows.

- We introduce fully quantized neural networks for image super-resolution to thoroughly quantize the model, including all layers within three sub-modules. To the best of our knowledge, *we are the first to perform fully end-to-end quantization for the SR task.*
- We identify several difficulties faced by current low bitwidth SR networks. Specifically, we quantize all skip connections to tackle a huge amount of memory consumption issue caused by high-resolution feature maps. We propose quantizers with learnable intervals to adapt the vastly distinct distributions of weights and activations in different network layers. To further reduce the quantization error, we also introduce a self-supervised calibration loss to predict the categorical distribution towards the original continuous distribution.
- Our extensive experiments with various bit configurations demonstrate that our FQSR is able to achieve comparable performance with the full-precision counterparts, while saving considerable amount of computation and memory usage. Moreover, experiments on mainstream architectures and datasets demonstrate the superior performance of the proposed FQSR over a few competitive state-of-the-art methods.

2 RELATED WORK

Image super-resolution. Super-resolution research has attracted increasing attention in recent years. Since the deep learning based super-resolution is first proposed by Dong *et al.* [5, 6], a variety of convolutional neural models have been studied.

ESPCN [26] is proposed to optimize the SR model by learning sub-pixel convolutional filters. Ledig *et al.* [16] introduce a Generative Adversarial Networks (GANs) SR model named SRGAN, along with which the generator is described as SRResNet. Lim *et al.* [17] propose a model named EDSR. Residual channel attention is introduced by Zhang *et al.* [35] to overcome gradient vanishing problem in very deep SR networks.

Besides, much effort has been devoted to improve the efficiency of the SR models by designing light-weight structures. For example, works in [7, 26] speed up the SR without the upsampling operations. Hui *et al.* [14] introduce a light-weighted information multi-distillation block into the proposed super-resolution model.
Model quantization. Model quantization aims to represent the weights, activations and even gradients in low-precision, to yield

highly compact DNNs. Notably, convolutions and matrix multiplications can be replaced with bitwise operations, which can be implemented more efficiently than the floating-point counterpart. In general, quantization methods involve binary neural networks (BNNs) and bitwise quantization. In particular, BNNs [13, 21, 25, 38] constrain both weights and activations to only two possible values (*e.g.*, +1 or −1), enabling the multiply-accumulations being replaced by the bitwise operations: xnor and bitcount. However, BNNs usually suffer from severe accuracy degradation. To make a trade-off between accuracy and efficiency, researchers also study bitwise quantization with higher-bit representation. To date, most quantization techniques employ uniform quantizers to fit the data, based on statistics of the data distribution [3, 36], minimizing quantization error during training [4, 32] or minimizing the task loss with stochastic gradient descent [8, 15, 37].

In terms of quantization for super-resolution, Ma *et al.* [23] apply BNNs to compress super-resolution networks. Note that it only proposes to binarize the weights of the residual blocks within the model. Most recently, Xin *et al.* [30] propose a bit-accumulation mechanism for single image super-resolution to boost the quantization performance. Both the weights and activations are quantized, however, their models are only partially quantized, with the feature extraction module, image reconstruction module and skip connections keeping in full precision. In contrast, our FQSR quantize all layers within three sub-modules and skip connections, which delivers improved efficiency and accuracy trade-off.

3 METHOD

3.1 Preliminary

In this work, we propose to quantize weights of all convolutional layers and activations of all the network layers into low-precision values. According to [25, 36], for two binary vectors $\boldsymbol{a} \in \{0, 1\}^N$ and $\boldsymbol{b} \in \{0, 1\}^N$ within binary neural networks (BNNs), the inner product of them can be formulated as:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \text{bitcount}(\boldsymbol{a} \& \boldsymbol{b}), \tag{1}$$

where bitcount counts the number of bits in a bit vector and & represents the bitwise "and" operation.

More generally, for quantization with higher and arbitrary bitwidths, the quantized values can be viewed as the linear combination of binary bases. Let $\boldsymbol{a}$ be a M-bit quantized vector which can be represented as $\boldsymbol{a} = \sum_{m=0}^{m=M-1} \boldsymbol{a}_m \cdot 2^m$, where $\boldsymbol{a}_m \in \{0, 1\}^N$. Similarly, for another P-bit vector $\boldsymbol{b}$, we have $\boldsymbol{b} = \sum_{p=0}^{p=P-1} \boldsymbol{b}_p \cdot 2^p$, where $\boldsymbol{b}_p \in \{0, 1\}^N$. Formally, the inner product calculation between $\boldsymbol{a}$ and $\boldsymbol{b}$ is

$$\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{m=0}^{M-1} \sum_{p=0}^{P-1} 2^{m+p} \text{bitcount} \left(\boldsymbol{a}_m \& \boldsymbol{b}_p \right). \tag{2}$$

For a general full-precision value v (activation or weight) to be quantized, an interval parameter is introduced to control the quantization range. The quantization function can be formulated as:

$$Q(v) = \left\lfloor \text{clip} \left(\frac{v}{I}, -Q_{\min}, Q_{\max} \right) \times (2^M - 1) \right\rceil \times \frac{I}{2^M - 1}, \tag{3}$$

where I represents the quantization interval, 2^M presents the quantization levels for M-bit quantization, $\lfloor v \rceil$ rounds v to the nearest integer, and clip $(v, v_{\text{low}}, v_{\text{up}}) = \min[\max(v, v_{\text{low}}), v_{\text{up}}]$. For unsigned data, $Q_{\min} = 0$ and $Q_{\max} = 1$; for signed data, $Q_{\min} = -1$ and $Q_{\max} = 1$. At the end of the equation, a scale factor $\frac{I}{2^M-1}$ is multiplied to the intermediate results after rounding operation to re-scale the value back to its original magnitude. In our paper, practically, we privatize quantizers for activations and weights in each layer.

During the training process, latent full-precision weights are kept to update the gradients during back-propagation, while being discarded during inference. The gradient is derived by using the straight-through estimator (STE) [1] to approximate the gradient through the non-differentiable rounding function as a pass-through operation, and differentiating all other operations in Eq. (3) normally.

In model quantization, the values within the quantization interval I will be quantized. The quantization process would proceed smoothly if a suitable quantization interval is determined. However, once the quantization interval does not fit in the distribution of values to be quantized, it would incur a large quantization error. For the super-resolution task, we empirically find that the data distributions of the features and weights of different layers are drastically different, as shown in Figure 1. Thus, different quantization intervals should be allocated for different quantizers. Toward this end, we propose Distribution-Aware Interval Adaptation (DAIA) to estimate the intervals automatically by parameterizing I. To alleviate the optimization difficulty of the interval, we devise to find a
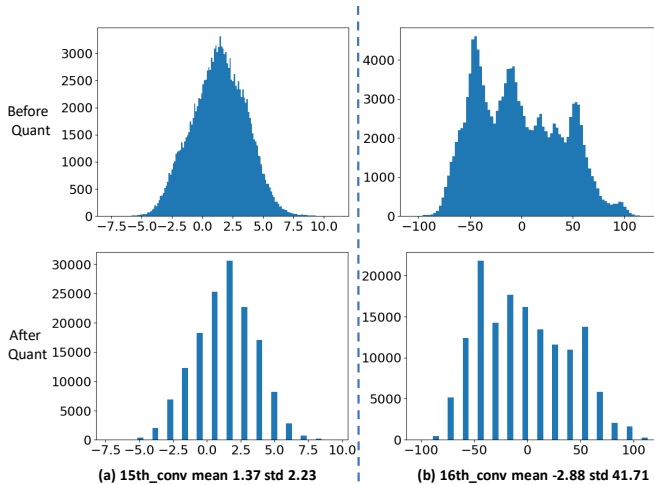

Figure 1: Here Plots (a) and (b) are the histograms of feature map values of 15th and 16th convolutional layers within SR-ResNet, respectively. For the super-resolution task, we empirically find that the data distribution ranges of the feature maps and weights within different layers are drastically different, as shown in (a) and (b) (mean 1.37 and std 2.23 for (a) and mean -2.88 and std 41.71 for (b)). Thus, we propose a trainable quantizer to adaptively decide the quantization interval according to the current distribution for mitigation of this phenomenon.
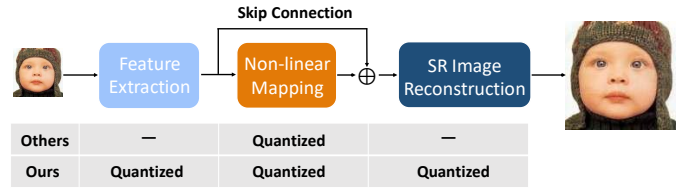

Figure 2: The overview of the proposed fully quantized super-resolution networks. The existing quantization models for super-resolution quantize the Non-linear Mapping part merely; while we quantize all three modules, by which large computation can be saved.

good initial point for I of a quantizer. Specifically, we propose to use the moving average of max values within the tensor V_i (batch-wise activations or convolutional filters within a layer) to be quantized as the initial point:

$$I = \frac{1}{l}\sum_{i=0}^{l-1} max(V_i). \tag{4}$$

This process is performed at the first l iterations of the model training as a warmup. Then the parameterized interval I is optimized in conjunction with other network parameters using back-propagation with stochastic gradient descent. Similar to the training process of [8], the gradient through the quantizer $Q(\cdot)$ to the quantization interval I is approximated by STE as a pass-through function. Such that the intervals can be tuned in conjunction with other parameters of the model to further increase the representation ability of models.

3.2 Fully Quantized Inference

According to [30], the super-resolution process can be divided into three sub-modules: input feature extraction module $\mathcal{E}$, nonlinear mapping module $\mathcal{M}$ and SR image reconstruction module $\mathcal{R}$. Formally, for an input low-resolution image lr, the aforementioned process to generate a super-resolution image sr can be presented as:

$$sr = \mathcal{R}(\mathcal{M}(\mathcal{E}(lr))). \tag{5}$$

Usually, there is a skip connection to link the feature extraction module $\mathcal{E}$ and the reconstruction module $\mathcal{R}$. Despite $\mathcal{E}$ and $\mathcal{R}$ consist of simple structures, they play an important role to achieve good performance in the super-resolution process. Moreover, large computational burdens are not only laid on the nonlinear mapping $\mathcal{M}$, but also on the reconstruction module $\mathcal{R}$, since the convolutional layers before upsampling are with a large number of channels to deal with the super-resolution output. However, current super-resolution quantization models propose to quantize $\mathcal{M}$ only [23, 30]. In addition, the existence of the skip connections within the network inevitably incurs huge memory consumption, which is known to dominate the energy consumption [10]. To obtain an energy-efficient super-resolution framework, we propose to fully quantize all layers of the three modules, especially including all

skip connections. The overview of the proposed fully quantized super-resolution network is shown in Figure 2 and the comparison of quantization differences between existing SR quantization methods and our proposed FQSR networks is presented in Table 1.

Quantization for BN. During the inference phase, if the batch normalization layer is adopted in the quantized model, it can be folded into the preceding convolutional layer to get rid of the extra floating-point operations. The folding of the batch normalization operation is formally presented as:

$$
\begin{aligned}
z &= \gamma \left[\frac{(w \cdot x + b) - \mu}{\sqrt{\sigma^2 + \epsilon}} \right] + \beta, \\
&= \frac{\gamma w}{\sqrt{\sigma^2 + \epsilon}} \cdot x + \frac{\gamma(b - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta, \\
&= w_{fold} \cdot x + b_{fold},
\end{aligned}
\tag{6}
$$

where w, x and b are the weights, inputs and bias term of the preceding convolutional layer, respectively; μ and σ are the mean and standard deviation of the corresponding dimension; z is the output of the batch normalization layer and w_{fold}, b_{fold} are the weights and bias after folding, respectively.

Quantization for skip connections. Residual learning is critical to fetch exceptional representations in computer vision tasks. In the residual structural networks, skip connections are the core components to build direct links between shallow layers and deeper ones. Nevertheless, in quantized models, the skip connections carrying floating-point operands will hinder the model to be applied practically on embedding systems or mobile platforms because the quantization status of each layer is inconsistent. In addition, it will inevitably increase the computation as well. Moreover, for the super-resolution task, the input images and the images after super-resolution are usually in very high resolution (such as 2K or 4K). Therefore, the intermediate features conveyed through skip connections will occupy a huge amount of memory consumption.

In order to address the aforementioned issues, we quantize the skip connections through quantizing the output features of all convolutional layers and the element-wise addition layers. Consequently, the memory consumption will be saved dramatically (can be saved approximately 8× when compared to the full-precision counterparts). Additionally, if the skip connections are quantized,

Table 1: Comparison of the quantized operations of different methods. Within the table, "✓" represents whether quantization is enabled for the column; "All modules" include input feature extraction module $\mathcal{E}$, nonlinear mapping module $\mathcal{M}$ and SR image reconstruction module $\mathcal{R}$; "wt" stands for weight quantization of convolutional layers; "fm" denotes the feature map quantization; "sc" denotes the quantization of skip connections.

Methods	All Modules	wt	fm	sc
SRResNet_Bin [23]		✓		
SRGAN_Bin [23]		✓		
VDSR_BAM [30]		✓	✓	
SRResNet_BAM [30]		✓	✓	
FQSR (Ours)	✓	✓	✓	✓
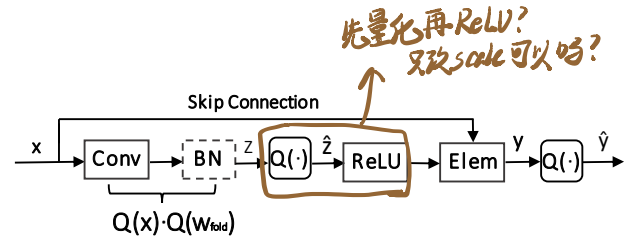
Figure 3: The process of the quantization in a typical residual block. In the process, $Q(\cdot)$ represents the quantization function. The input x is the output of the preceding layer/residual block. The outputs of the convolutional layer z and the element-wise layer y are quantized to ensure the quantization of skip connections.

the models are hardware-friendly since it is fully quantized. Formally, the element-wise addition operation of the skip connection in our quantized network can be formulated as:

$$
y = Q(x) + \text{ReLU}(Q(z)).
\tag{7}
$$

Overall, the process of quantization in a typical residual block in the proposed FQSR network is shown in Figure 3. Within the figure, $\hat{x}$ represents $Q(x)$.

3.3 Self-supervised Quantization-aware Calibration Loss

As shown in Figure 4, for the super-resolution task, we empirically observe that in some layers, especially in the last two layers, the data distributions before and after quantization change drastically. It is may be caused by the last two layers are responsible for upscaling purpose. Thus, the weight distributions are different from other feature transformation layers because of the dissimilar functionalities. However, this phenomenon will affect the model performance significantly due to the large quantization error. In order to minimize the quantization error, an objective function termed as Self-supervised Quantization-aware Calibration Loss (SQCL) is adopted to calibrate the values after quantization to have an approximate distribution as before quantization. In this case, the value before quantization serves as a strong self-supervision signal, which provides additional useful information for network optimization. The SQCL loss is applied to input activations, weights and outputs of each layer.

For a real value v to be quantized, here we are targeting to find optimal parameters to minimize the difference of before and after quantization through back-propagation. Formally, SQCL is formulated as:

$$
L_q = \| Q(v) - v \|_p,
\tag{8}
$$

where $\| \cdot \|_p$ denotes the L_p norm.
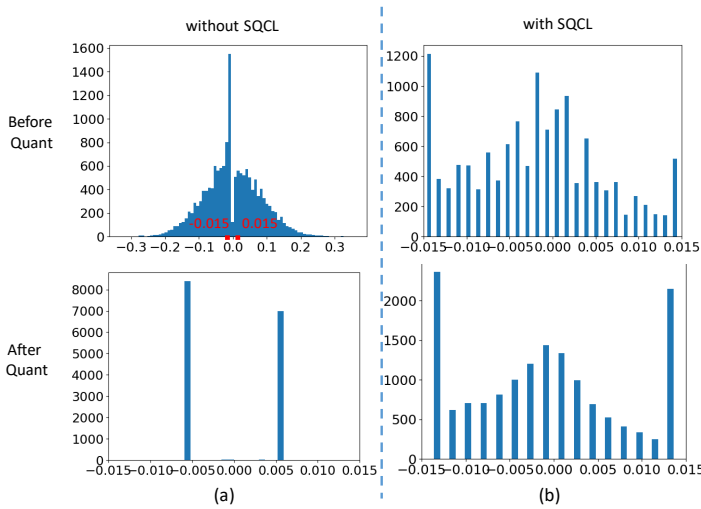
Therefore the final objective function for the proposed super-resolution quantized networks is:

$$
L = L_{sr} + \alpha L_q,
\tag{9}
$$

where L_{sr} represents the super-resolution loss and α is a balancing hyperparameter.

Discussion. The proposed quantization-aware calibration loss (SQCL) is different from the network distillation. In the design of network

Figure 4: We show the weights' distribution of the last convolutional layer within SRResNet and its categorical distribution (4 bits, with 16 bins), before and after adding the SQCL objective. Intuitively, the SQCL is adopted to minimize the jitters before and after quantization to constrain the model quantization in a smoother manner. Except for the sub-figure in the upper-left corner (since the range of the distribution within this sub-figure is hugely different from others), the other sub-figures are drawn under the range [−0.015, 0.015]. As shown in (a), for the model without SQCL objective, the data distributions before quantization and the categorical distribution after quantization are vastly different; With the SQCL objective as illustrated in (b), the categorical distribution is calibrated to fit the data distribution before quantization.

distillation, a teacher network (a larger or full precision model) is leveraged to train a student network (a small or low-bit quantization model). The teacher network is generally well-pretrained and provides ground truth to the student network. In contrast, our proposed quantization-aware calibration loss is an objective for self-supervised calibration of the quantizers, which encourages the quantized tensor to have a similar distribution to the original one. We employ the quantization-aware calibration loss for every quantization function, including weights and features, rather than only the network outputs.

Besides, the proposed quantization-aware calibration loss is also different from the previous quantization method LQ-net [32]. In LQ-net, the authors propose to solve the non-uniform step sizes with a closed-form solution derived from a similar target in Eq. (8). Unlike the LQ-net, we adopt the gradient-based loss function to stabilise the training process of the super-resolution quantization.

4 EXPERIMENTS

4.1 Experimental Setup

Following the existing works [16, 17, 23, 30], we train our fully quantized super-resolution networks on DIV2K [27] dataset and evaluate models on five prevalent benchmark datasets. An extensive ablation study is further conducted to validate the effectiveness of each component within the proposed method.

Datasets and evaluation metrics. We conduct the model training on the DIV2k dataset, which is made up of 800 good quality high/low-resolution image pairs for model training, 100 image pairs for model validation and 100 image pairs for testing. However, the testing HR images for DIV2K is not publicly accessible, so we train models on 800 training images and validate models on 10 validation images. The best validation models are tested on Set5 [2] (5 images), Set14 [31] (14 images), BSD100 [24] (100 images), Urban100 [12] (100 images) and DIV2K (100 validation images). Two scaling settings are considered for model evaluation, containing ×2 and ×4.

For the super-resolution model evaluation, we take the most commonly adopted Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [29] as our evaluation metrics. All evaluation is performed by cropping s pixels for ×s upscaling.

Implementation details. During training, random vertical / horizontal flips and 90-degree rotation are performed for data augmentation. The model batch size is set to 16 and Adam optimizer is adopted for model optimization. The initial learning rate is set to 1×10^{-3} for SRResNet and SRGAN and 5×10^{-5} for the EDSR model. The models are trained for 300 epochs with cosine annealing [22] learning rate tuning strategy. The hyperparameter l is set to 20 and the trade-off factor α is set to 0.3. L_1 norm is adopted for SQCL calculation. The models are implemented using PyTorch with one NVIDIA GTX 1080 Ti GPU. The experimental settings are fixed to all of our trained models to keep fair comparisons.

4.2 Overall Performance

We embed the proposed fully quantized super-resolution scheme on three state-of-the-art architectures, containing SRResNet, EDSR and SRGAN, to compare the effectiveness of low-bitwidth models with full-precision models and bicubic interpolation. The results are shown in Tables 2, 3 and 4.

Evaluation on SRResNet. As shown in Table 2, we implement the FQSR on SRResNet with multiple configurations. Also, multiple state-of-the-art quantized SR models are compared with the proposed model. When compared to the bicubic interpolation, the 4/4/8 model (i.e., weights and activations are both quantized to 4 bits; skip connections are quantized to 8 bits) surpasses it by 2.678 with ×2 up-scaling and 2.618 with ×4 up-scaling for PSNR on Set5 dataset. By raising the skip connection precision to 8 bits, the performances are boosted by a large margin, e.g., 1.814 surpass on Set5 and 1.101 on Set14 for PSNR ×2 up-scaling, respectively. It is worth noting that, on both ×2 and ×4 up-scaling settings, the 6/6/8 models achieve comparable results with or outperform the full-precision version of SRResNet. In addition, the 8/8/32 version fully quantized model significantly outperforms the full-precision counterpart by 0.121 and 0.336 on the PSNR metric with Set5 dataset for ×2 and ×4 up-scaling, respectively. The last rows of both ×2 and ×4 up-scaling settings are the lite version 6/6/8 configuration of the proposed FQSR model named as FQSR_Lite, within which the nonlinear mapping $\mathcal{M}$ module only consists of 10 residual blocks rather than 16. Thus, we intend to save more computational cost

while not losing much performance. On both ×2 and ×4 up-scaling settings, the FQSR_Lite surpasses or receives comparable results with the 6/6/8 configuration, but with a less computational cost.

Evaluation on EDSR. In the evaluation on EDSR, as shown in Table 3, the 4/4/8 model is able to outperform bicubic interpolation by a large margin, 3.378 for ×2 and 2.508 for ×4 on Set5. On both ×2 and ×4 up-scaling settings, the 6/6/8 models achieve comparable results with the full-precision version of EDSR. The 8/8/8 and 8/8/32 models outperform the full-precision baseline model on most of the metrics. On ×2, 0.137 PSNR improvement on Set14 and 0.283 PSNR improvement on Urban100 are obtained by the 8/8/32 model compared to the full-precision model.

Evaluation on SRGAN. The evaluation on SRGAN is shown in Table 4. Similar to the evaluation on SRResNet, the performance of 4/4/8 models achieve significantly better performance than bicubic interpolation on most of the metrics and datasets. Surprisingly, on ×2 setting, the 6/6/8 setting outperforms the full-precision model on multiple metrics, *i.e.*, 0.075 PSNR improvement on Set5 and 0.189 PSNR boost on Set14 compared with the full-precision model. Moreover, the 8/8/32 model outperforms the full-precision model on most of the metrics.

4.3 Comparison with Existing SR Quantization Models

The comparison of the proposed FQSR model with Ma *et al.* [23] and Xin *et al.* [30] on SRResNet is shown in Table 2, since they all provide results on SRResNet structure. Worth noting that, in [23], the models are trained 500 epochs for SRResNet and in [30] the learning rate is decreased by half every 200 epoch, while we only train the FQSR model 300 epochs for comparison. With much fewer training epochs, the proposed FQSR models are able to achieve better performance with less computation cost and memory consumption. Following [9, 19, 28], OPs is the sum of low-bit operations and floating-point operations, *i.e.*, for M-bit networks, OPs = BOPs/64 M + FLOPs. Only the multiplication operations are calculated for OPs. In terms of memory consumption, because of the existence of long and short skip connections within the networks, we consider the peak memory consumption of each model at the inference stage. Maximally, feature maps of three convolutional layers are considered for SRResNet_Bin and our proposed FQSR networks (one for long skip connection feature storing, one for short connection and another for the main trunk); the features of only one convolutional layer is considered for SRResNet_w/o $\mathcal{M}$, since it just consists of three convolutional layers without skip connections. However, in terms of the SRResNet_BAM model, because the activation quantization of each layer takes outputs of several preceding layers into consideration (these activations should be stored for re-using), features of 33 convolutional layers within $\mathcal{M}$ are computed. We consider 1020×678 resolution DIV2K dataset images as inputs and ×2 up-scaling as the configuration. The OPs are in the unit G (=1×10^9) OPs and Memory consumption is in the unit M (=1×10^6) Bytes.

In the table, SRResNet_Bin is the binary SR network from paper [23]. Because only the weights of each layer are quantized, floating-point operations are still required in the feature extraction module and image reconstruction module, as well as the skip connection

to link these two modules. Thus, the OPs and memory consumption within in SRResNet_Bin will not be reduced. SRResNet_BAM is the bit accumulation model proposed by [30]. It binarizes both the activation and weights of each convolutional layer, so it reduces the OPs and memory consumption to some extent. However, it does not take the quantization of convolutional layers before and after upsampling into consideration, which introduces huge OPs consumption. This is because in SR models, the convolutional channels should be raised before upsampling and the size of features is increased to their multiples after upsampling operations. What is more, the SRResNet_BAM does not consider quantize the feature extraction module, image reconstruction module and the linking skip connection as well. SRResNet_w/o $\mathcal{M}$ is the model that only consists of one convolutional layer within $\mathcal{E}$ and two convolutional layers within $\mathcal{R}$. The results show without $\mathcal{M}$, the simple full-precision super-resolution model could achieve comparable performance, such that the existence of a full-precision sub-net will shrink the significance of the model quantization dramatically. The results received by the simplified model represents the importance to quantize the Feature Extraction module and the Reconstruction module.

If we compare SRResNet_Bin with FQSR quantitatively, the 4/4/8 FQSR models (both ×2 or ×4 scales) are able to outperform SRResNet_Bin models across almost all the metrics. When compared with SRResNet_BAM algorithm, from the table, we can perceive that with approximately 1/5 of the OPs and 1/50 memory consumption only (FQSR 6/6/8 model) on ×2 up-scaling, the FQSR model is able to achieve better results on multiple metrics and datasets (37.541 over 37.210 on PSNR for Set5). If we increase the bit number, the gaps will become bigger. Finally, our proposed lite version 6/6/8 model is able to receive better results compared to SRResNet_BAM with remarkably fewer OPs. Furthermore, we compare FQSR with DoReFa [36] models on the backbone of SRResNet, EDSR and SR-GAN. As shown in table 3, for the 8/8/8 EDSR_DoReFa model on both ×2 and ×4, the 8/8/8 FQSR model outperforms it across most of the metrics. Similar phenomenon is shown in table 4 and table 2 as well. This clearly shows the effectiveness of the proposed FQSR model.

4.4 Ablation Study

Effect of different components. In this section, we examine the effect of each component in our FQSR model. The experimental results are reported in Table 5. We empirically find that the proposed distribution-aware trainable quantization interval and the calibration loss are critical for the model to gain promising performance in the super-resolution process. With the trainable quantizers only, the PSNR performance of the baseline model is raised from 35.536 to 36.372 on Set5, and significant improvements on other metrics and datasets can also be observed. When equipped with strategies, the performance is further boosted to 36.854 on Set5. The ablation study generally shows the effectiveness of the proposed methods.

Effect of α. This section presents the sensitivity of our FQSR model with different α setting in Eq. (9). Figure 5 demonstrates the PSNR and SSIM performance of FQSR on different datasets with the ×2 up-scaling setting. It is clear that FQSR generally performs stably w.r.t. different α settings. From the figure, the trend of curves raising

Table 2: The comparison between existing methods and our FQSR on SRResNet [16]. The OPs are in the unit G Flops and Memory consumption is in the unit M Bytes. Similar to Table 1, "wt" represents weight quantization of convolutional layers; "fm" is the feature map quantization of layers; "sc" denotes the quantization of skip connections; "p1" represents the corresponding models are partially binarized.

Methods	Scale	wt	fm	sc	OPs	Memo	Set5		Set14		B100		Urban100		DIV2K	
							PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRResNet [16]	×2	32	32	32	997.018	531.117	37.760	0.958	33.270	0.914	31.950	0.895	31.280	0.919	-	-
Bicubic	×2	32	32	32	-	-	33.660	0.930	30.240	0.869	29.560	0.843	26.880	0.840	31.010	0.939
SRResNet_Bin [23]	×2	p1	32	32	997.018	531.117	35.660	0.946	31.560	0.897	-	-	28.760	0.882	-	-
SRResNet_BAM [30]	×2	p1	p1	32	159.035	5842.287	37.210	0.956	32.740	0.910	31.600	0.891	30.200	0.906	-	-
SRResNet_DoReFa [36]	×2	8	8	8	62.314	132.779	37.205	0.956	32.967	0.911	31.837	0.894	30.974	0.916	31.050	0.939
SRResNet_w/o $\mathcal{M}$	×2	32	32	32	155.749	177.039	36.863	0.954	32.536	0.907	31.379	0.887	29.525	0.896	33.268	0.934
	×2	4	4	8	15.578	132.779	36.338	0.945	32.403	0.901	31.367	0.882	29.982	0.899	32.357	0.927
	×2	4	4	32	15.578	531.117	36.854	0.953	32.710	0.908	31.583	0.890	30.430	0.909	32.985	0.935
FQSR (Ours)	×2	6	6	8	35.106	132.779	37.541	0.957	33.236	0.913	31.966	0.894	31.398	0.920	33.964	0.941
	×2	8	8	8	62.314	132.779	37.555	0.958	33.202	0.914	31.972	0.896	31.356	0.921	33.452	0.942
	×2	8	8	32	62.314	531.117	37.881	0.959	33.408	0.915	32.093	0.897	31.712	0.924	34.424	0.943
FQSR_Lite (Ours)	×2	6	6	8	14.456	132.779	37.349	0.956	33.070	0.912	31.851	0.895	30.964	0.916	33.753	0.939
SRResNet [16]	×4	32	32	32	383.484	132.777	31.760	0.888	28.250	0.773	27.380	0.727	25.540	0.767	-	-
Bicubic	×4	32	32	32	-	-	28.420	0.810	26.000	0.703	25.960	0.668	23.140	0.658	26.660	0.852
SRResNet_Bin [23]	×4	p1	32	32	383.484	132.777	30.340	0.864	27.160	0.756	-	-	24.480	0.728	-	-
SRResNet_BAM [30]	×4	p1	p1	32	173.997	1460.580	31.240	0.878	27.970	0.765	27.150	0.719	24.950	0.745	-	-
SRResNet_DoReFa [36]	×4	8	8	8	23.968	33.194	31.539	0.885	28.156	0.771	27.299	0.724	25.384	0.763	28.132	0.825
SRResNet_w/o $\mathcal{M}$	×4	32	32	32	173.175	44.260	30.880	0.841	27.808	0.723	27.059	0.694	24.777	0.714	28.081	0.815
	×4	4	4	8	5.992	33.194	31.038	0.874	27.860	0.761	27.090	0.714	24.949	0.744	27.925	0.816
	×4	4	4	32	5.992	132.777	31.303	0.880	28.045	0.767	27.188	0.719	25.165	0.754	28.074	0.821
FQSR (Ours)	×4	6	6	8	13.499	33.194	31.923	0.889	28.404	0.775	27.452	0.727	25.752	0.774	28.571	0.830
	×4	8	8	8	23.968	33.194	32.098	0.888	28.514	0.773	27.526	0.725	25.968	0.770	28.328	0.829
	×4	8	8	32	23.968	132.777	32.096	0.892	28.559	0.780	27.555	0.732	26.034	0.783	28.894	0.836
FQSR_Lite (Ours)	×4	6	6	8	8.339	33.194	31.644	0.886	28.249	0.774	27.348	0.727	25.460	0.765	28.405	0.826

Table 3: The comparison of our FQSR with full-precision networks on EDSR [17] and Bicubic interpolation.

Methods	Scale	wt	fm	sc	Set5		Set14		B100		Urban100		DIV2K	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR [17]	×2	32	32	32	37.885	0.958	33.425	0.915	32.106	0.897	31.777	0.924	34.471	0.944
Bicubic	×2	32	32	32	33.660	0.930	30.240	0.869	29.56	0.843	26.880	0.84	31.010	0.939
EDSR_DoReFa [36]	×2	8	8	8	37.849	0.958	33.418	0.915	32.096	0.897	31.746	0.924	34.374	0.943
	×2	4	4	8	37.038	0.951	32.835	0.908	31.668	0.889	30.646	0.911	33.282	0.933
	×2	4	4	32	37.087	0.951	32.868	0.908	31.690	0.889	30.698	0.912	33.329	0.934
FQSR (Ours)	×2	6	6	8	37.817	0.958	33.411	0.915	32.092	0.897	31.784	0.925	34.306	0.943
	×2	8	8	8	37.955	0.959	33.524	0.916	32.152	0.898	32.008	0.927	34.533	0.944
	×2	8	8	32	37.993	0.959	33.562	0.916	32.171	0.898	32.060	0.927	34.586	0.945
EDSR [17]	×4	32	32	32	32.007	0.892	28.486	0.778	27.528	0.731	25.934	0.781	28.880	0.835
Bicubic	×4	32	32	32	28.420	0.810	26.000	0.703	25.960	0.668	23.140	0.658	26.660	0.852
EDSR_DoReFa [36]	×4	8	8	8	32.042	0.891	28.474	0.778	27.518	0.730	25.870	0.778	28.814	0.834
	×4	4	4	8	30.928	0.870	27.816	0.761	27.073	0.715	24.927	0.744	27.963	0.814
	×4	4	4	32	30.983	0.872	27.856	0.762	27.090	0.716	24.947	0.746	27.976	0.815
FQSR (Ours)	×4	6	6	8	31.948	0.889	28.415	0.776	27.466	0.728	25.751	0.775	28.667	0.831
	×4	8	8	8	32.102	0.892	28.524	0.779	27.539	0.731	25.944	0.781	28.853	0.835
	×4	8	8	32	32.157	0.893	28.545	0.780	27.550	0.732	25.989	0.783	28.898	0.836

to peaks then falling can be observed for both PSNR and SSIM performance. In general, $\alpha = 0.3$ is recommended for the FQSR model to achieve the best performance.

5 CONCLUSION

In this paper, we have proposed a fully quantized super-resolution framework, including all layers within three SR sub-modules, as a practical solution to achieve a good trade-off between accuracy and efficiency. We have also identified difficulties faced by current

Table 4: The comparison of Fully Quantized Super-resolution networks with full-precision networks on SRGAN [16] and Bicubic interpolation.

Methods	Scale	wt	fm	sc	Set5		Set14		B100		Urban100		DIV2K	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRGAN [16]	×2	32	32	32	37.446	0.958	33.033	0.914	31.971	0.896	31.300	0.920	33.885	0.942
Bicubic	×2	32	32	32	33.660	0.930	30.240	0.869	29.560	0.843	26.880	0.840	31.010	0.939
SRGAN_DoReFa [36]	×2	8	8	8	37.344	0.956	33.045	0.912	31.876	0.894	31.133	0.918	31.431	0.939
FQSR (Ours)	×2	4	4	8	36.693	0.950	32.644	0.906	31.565	0.888	30.373	0.908	32.921	0.933
	×2	4	4	32	36.731	0.952	32.640	0.906	31.550	0.889	30.327	0.907	32.859	0.934
	×2	6	6	8	37.521	0.957	33.222	0.913	31.955	0.894	31.343	0.919	33.975	0.941
	×2	8	8	8	37.669	0.957	33.293	0.914	32.009	0.895	31.488	0.921	34.162	0.942
	×2	8	8	32	37.665	0.958	33.254	0.914	31.980	0.895	31.378	0.919	34.060	0.941
SRGAN [16]	×4	32	32	32	31.934	0.890	28.451	0.776	27.470	0.728	25.824	0.775	28.712	0.832
Bicubic	×4	32	32	32	28.420	0.810	26.000	0.703	25.960	0.668	23.140	0.658	26.660	0.852
SRGAN_DoReFa [36]	×4	8	8	8	31.351	0.883	28.074	0.770	27.254	0.723	25.294	0.760	28.184	0.824
FQSR (Ours)	×4	4	4	8	30.963	0.872	27.854	0.759	27.078	0.713	24.932	0.742	27.833	0.814
	×4	4	4	32	31.253	0.879	27.997	0.766	27.164	0.718	25.105	0.752	27.967	0.820
	×4	6	6	8	31.874	0.889	28.398	0.775	27.443	0.726	25.732	0.772	28.625	0.830
	×4	8	8	8	31.960	0.890	28.483	0.777	27.514	0.730	25.898	0.778	28.760	0.833
	×4	8	8	32	32.030	0.891	28.482	0.778	27.499	0.729	25.864	0.777	28.793	0.833

Table 5: Ablation study on each component. The experiments are conducted on the 4/4/32 and ×2 up-scaling setting.

Models	DAIA	SQCL	Set5		Set14		B100		Urban100		DIV2K	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1			35.536	0.944	31.775	0.898	30.972	0.883	28.893	0.888	31.792	0.926
2	✓		36.372	0.945	32.395	0.901	31.397	0.884	30.164	0.902	32.455	0.927
3	✓	✓	36.854	0.953	32.71	0.908	31.583	0.89	30.43	0.909	32.985	0.935
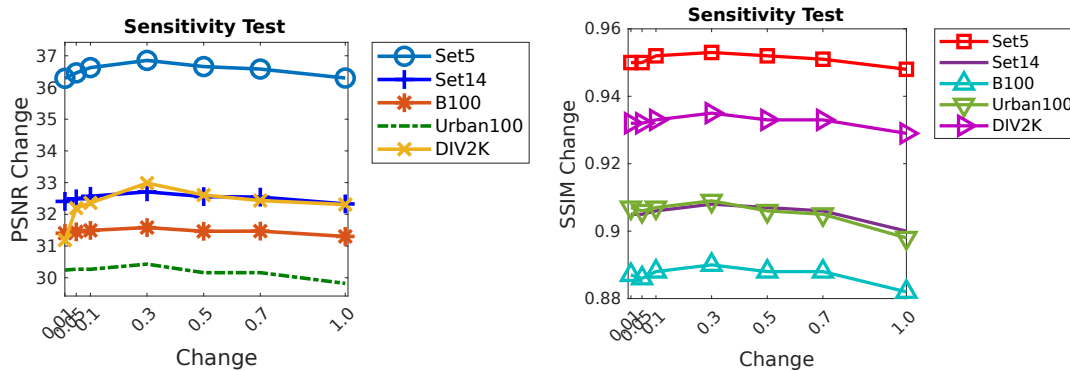
Figure 5: PSNR and SSIM performance of FQSR using different loss trade-off factors on different datasets with ×2 up-scaling.

low-bitwidth SR networks. That is 1) huge memory consumption caused by high-resolution feature maps 2) activation and weight distributions being vastly distinctive in different layers; 3) the inaccurate approximation of the quantization. In order to solve them, we quantize skip connections and two practical components have been proposed, a distribution-aware interval adaptation strategy to automatically decide the quantization intervals during training and a self-supervised quantization-aware calibration loss to explicitly minimize the quantization error. We have evaluated our method on multiple state-of-the-art deep super-resolution models on five benchmark datasets. The extensive experimental results have shown that our proposed FSQR is able to achieve state-of-the-art results while saving considerable computational cost and memory usage compared to the full-precision counterparts and competing methods. The ablation study further shows the proposed DAIA and SQCL are able to boost the model performance in a complementary manner.

REFERENCES

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. 2012. Low-complexity single-image super-resolution based on nonnega-tive neighbor embedding. (2012).

[3] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. 2017. Deep Learning With Low Precision by Half-Wave Gaussian Quantization. In *IEEE Conf. Comput. Vis. Pattern Recog.* 5918–5926.

[4] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085* (2018).

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *Eur. Conf. Comput. Vis.* Springer, 184–199.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 2 (2015), 295–307.

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. 2016. Accelerating the super-resolution convolutional neural network. In *Eur. Conf. Comput. Vis.* Springer, 391–407.

[8] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. 2019. Learned step size quantization. *arXiv preprint arXiv:1902.08153* (2019).

[9] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. 2020. Single path one-shot neural architecture search with uniform sampling. In *Eur. Conf. Comput. Vis.* 544–560.

[10] Song Han, Huizi Mao, and William J Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Int. Conf. Learn. Represent.*

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[12] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single image super-resolution from transformed self-exemplars. In *IEEE Conf. Comput. Vis. Pattern Recog.* 5197–5206.

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. In *Adv. Neural Inform. Process. Syst.* 4107–4115.

[14] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. 2019. Lightweight image super-resolution with information multi-distillation network. In *Pro. ACM Int. Conf. Multimedia.* 2024–2032.

[15] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. 2019. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4350–4359.

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunning-ham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4681–4690.

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 136–144.

[18] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. Darts: Differentiable architecture search. In *Int. Conf. Learn. Represent.*

[19] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. 2020. ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions. *arXiv preprint arXiv:2003.03488* (2020).

[20] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. In *Adv. Neural Inform. Process. Syst.*

[21] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. 2018. Bi-Real Net: Enhancing the Performance of 1-bit CNNs with Improved Representational Capability and Advanced Training Algorithm. In *Eur. Conf. Comput. Vis.*

[22] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[23] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. 2019. Efficient super resolution using binarized neural network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops.* 0–0.

[24] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmen-tation algorithms and measuring ecological statistics. In *Int. Conf. Comput. Vis.*, Vol. 2. IEEE, 416–423.

[25] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Eur. Conf. Comput. Vis.* 525–542.

[26] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.* 1874–1883.

[27] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops.* 114–125.

[28] Ying Wang, Yadong Lu, and Tijmen Blankevoort. 2020. Differentiable Joint Pruning and Quantization for Hardware Efficiency. In *Eur. Conf. Comput. Vis.* 259–277.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 4 (2004), 600–612.

[30] Jingwei Xin, Nannan Wang, Xinrui Jiang, Jie Li, Heng Huang, and Xinbo Gao. 2020. Binarized Neural Network for Single Image Super Resolution. In *Eur. Conf. Comput. Vis.*

[31] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On single image scale-up using sparse-representations. In *Proc. Int. Conf. Curves and Surfaces.* Springer, 711–730.

[32] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. 2018. LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks. In *Eur. Conf. Comput. Vis.*

[33] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conf. Comput. Vis. Pattern Recog.* 6848–6856.

[34] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. 2016. Accelerating very deep convolutional networks for classification and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 10 (2016), 1943–1955.

[35] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Eur. Conf. Comput. Vis.* 286–301.

[36] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* (2016).

[37] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. 2018. Towards Effective Low-bitwidth Convolutional Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[38] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. 2019. Strutured Binary Neural Network for Accurate Image Classification and Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[39] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. 2018. Discrimination-aware channel prun-ing for deep neural networks. In *Adv. Neural Inform. Process. Syst.* 875–886.