Toward Accurate Post-Training Quantization for Image Super Resolution

Zhijun Tu, Jie Hu, Hanting Chen, Yunhe Wang
Huawei Noah's Ark Lab
{zhijun.tu, hujie23, chenhanting, yunhe.wang}@huawei.com

Abstract

Model quantization is a crucial step for deploying super resolution (SR) networks on mobile devices. However, existing works focus on quantization-aware training, which requires complete dataset and expensive computational overhead. In this paper, we study post-training quantization (PTQ) for image super resolution using only a few unlabeled calibration images. As the SR model aims to maintain the texture and color information of input images, the distribution of activations are long-tailed, asymmetric and highly dynamic compared with classification models. To this end, we introduce the density-based dual clipping to cut off the outliers based on analyzing the asymmetric bounds of activations. Moreover, we present a novel pixel aware calibration method with the supervision of the full-precision model to accommodate the highly dynamic range of different samples. Extensive experiments demonstrate that the proposed method significantly outperforms existing PTQ algorithms on various models and datasets. For instance, we get a 2.091 dB increase on Urban100 benchmark when quantizing EDSR×4 to 4-bit with 100 unlabeled images. Our code is available at both PyTorch and MindSpore.

1. Introduction

Image super resolution (SR) is a classical image processing task in computer vision, which reconstructs high-resolution (HR) images from the corresponding low-resolution (LR) images. SR has been widely applied in the real-world scenarios, such as medical imaging [12, 35], surveillance [1, 49], satellite imagery [31, 36] and smartphone display [8, 19]. With the rapid development of deep learning in recent years, SR models with deep neural network (DNN) structure have continued to achieve state-of-the-art performance on various datasets. However, these SR models require significant storage and computational resources, which makes their deployment on mobile devices extremely difficult. To improve the inference efficiency, various techniques have been proposed to compress the models, such as network pruning [16, 50], model quantization [13, 38], compact architecture design [8, 9] and knowledge distillation [29, 41, 45, 46]. Among these approaches, model quantization is much benefit to existing artificial intelligent (AI) accelerators [3, 42], which generally focus on low-precision arithmetic, resulting in lower latency, smaller memory footprint and less energy consumption.

Table 1. Computational overhead of different quantization methods on EDSR model. The FP denotes full-precision training, the Gt denotes the ground-truth, and the Bs denotes batch size.

Method	Type	Data	Gt	Bs	Iters	Run time
EDSR [28]	FP	800	✓	16	15,000	240×
PAMS [25]	QAT	800	✓	16	1,500	24×
FQSR [40]	QAT	800	✓	16	15,000	120×
CADyQ [14]	QAT	800	✓	8	30,000	240×
DAQ [15]	QAT	800	✓	4	300,000	1200×
DDTB [52]	QAT	800	✓	16	3,000	48×
Ours	PTQ	100	✗	2	500	1×

Although the previous SR quantization methods make great effort on improving the performance with given bit-width, their main drawback is that they require quantization aware training (QAT) with complete datasets and expensive computational overhead. As shown in Table 1, the full-precision EDSR model needs to train 15,000 iters with the batch size of 16, takes 8 days on NVIDIA Titan X GPUs [28]. To recover the performance drop of the quantized models, most methods also need to train with the same iterative steps on the complete training dataset, in which one training step in QAT actually takes more GPU memory and longer running time than those of the regular floating-point.

On the contrast, post-training quantization (PTQ) only requires a few unlabeled calibration images without training, which enables fast deployment on various devices within minutes. Nevertheless, different from the image classification, super resolution requires accurate prediction for each pixel of the output images, which is much sensitive to low-bit compression for feature maps. Figure 2 shows the original floating-point activations of different layers and samples, we observe three properties of their distributions that are much unfriendly to quantization: (1) **Long-tailed**:

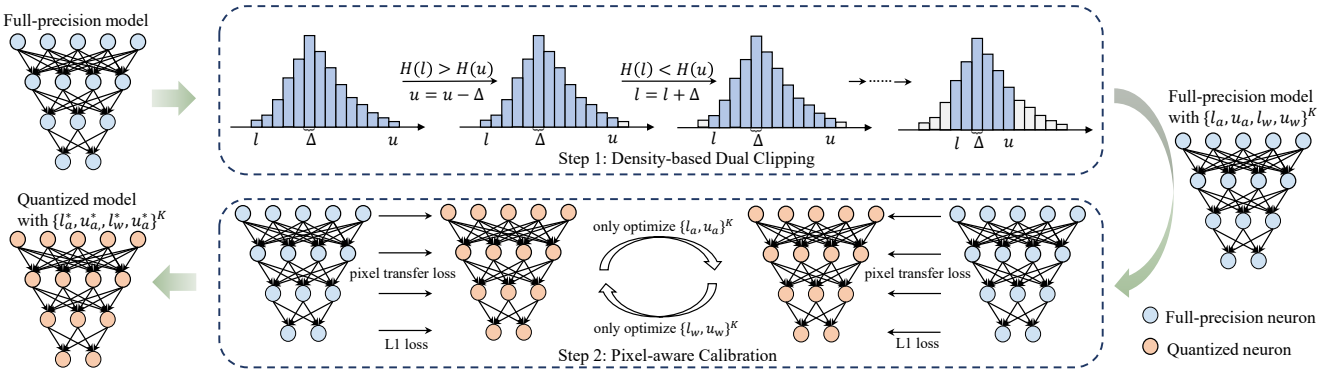


Figure 1. The overview of the proposed post-training quantization framework for image super resolution.

the distribution shows to be dense in the middle yet sparse in the tails, which means most of values lie in a small range, while only a few outliers have larger amplitude; (2) **Asymmetric**: the density on the two tails of the distribution is asymmetric, the skewness differ for different layers; (3) **Highly-dynamic**: the activation range varies, or even by twice, for different input samples. Therefore, the existing PTQ methods which are designed for image classification can not be transferred to the SR task directly.

In this paper, we propose a coarse-to-fine method to get the accurate quantized SR model with post-training quantization. We first introduce the density-based dual clipping (DBDC) to cut off most of the outliers for narrowing the distribution to a valid range. Different from previous methods [25, 40], the amplitudes of lower and upper clip are not same and the clipping position is depend on the density of two tails. The clipping scheme is employed iteratively to eliminate the long-tail distribution. The asymmetric quantizer with adjustable lower and upper clip values is adopted to solve the asymmetric distribution in SR models. And then we further propose a novel pixel-aware calibration (PaC) to help the quantized network fit the highly dynamic activations for different samples. The PaC leverages feature maps of the full-precision model to supervise those of the quantized model. To stabilize the finetune process, we only update the quantization parameters instead of the original weights. The whole quantization process of our method can be finished within minutes with a few unlabeled images. The contributions of this paper are summarized as follow:

(1) We present a detailed analysis to demonstrate the challenge of post-training quantization on image super resolution, indicating that the performance degradation of quantized SR model suffers from the long-tailed, asymmetric and highly-dynamic distribution of feature maps.

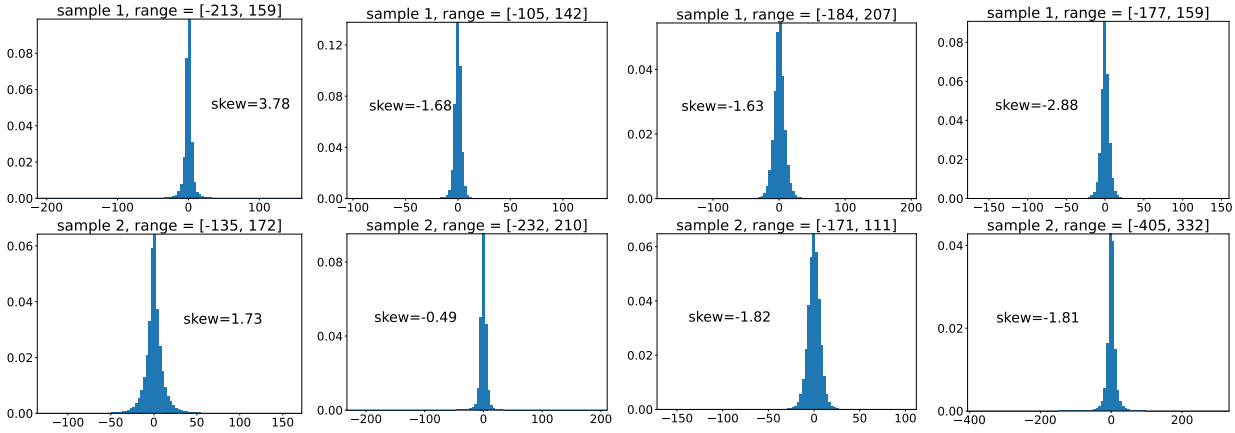
(2) We introduce a coarse-to-fine quantization method to accommodate above problems. With the density-based dual clipping and the pixel-aware calibration, the proposed method is able to conduct accurate quantization with only a few unlabeled calibration images. To the best of our knowledge, we are the first to optimize the post-training quantization for image super resolution task.

(3) Extensive experiments on various benchmark models and datasets demonstrate that our method significantly outperforms the existing PTQ methods, and is able to achieve comparable performance with the QAT in some setting. Further, our method can speed up the convergence and bring up the performance when combined with QAT methods.

2. Related works

Model quantization is a promising technique for compressing deep neural networks, which has received extensive attention and has been applied in various tasks widely. Ma *et al.* [32] firstly explored the weight binarization for image SR task, only compressed the residual blocks with a learnable weight for each binary filter. BAM [44] proposed a bit accumulation mechanism to approximate the full-precision convolution and BTM [21] introduced a novel training mechanism based on the feature distribution. E2FIF [23] constructed binary super resolution networks from the perspective of information flow integrity.

Except for the 1-bit binarization, some recent works [15, 25, 40, 52] also focus on the optimization of low-bit quantization. PAMS [25] designed a layer-wise symmetric quantizer with the learnable clip value only for high-level feature extraction module. To compress the SR network further, FQSR [40] quantized all the layers and residual branch with learnable quantization interval. DAQ [15] introduced a channel-wise distribution-aware quantization scheme and DDTB [52] designed an asymmetric activation quantizer with dynamic dual trainable clip values, pushing the compression to 2-4 bit. However, these above methods all need to train on the complete dataset and take even longer than the training of full-precision models. To this end, we aim to propose a post-training quantization method to compress the SR models without training, and only need a few unlabeled data to calibrate the clipping values.



(a) body.0.conv1 (b) body.11.conv1 (c) body.21.conv1 (d) body.31.conv1

Figure 2. The histograms of feature maps of different layers with different samples. The skew denotes the asymmetry of the distribution

3. Methods

We first give a brief introduction to the general quantization method. Given a floating-point tensor x that needs to be quantized into N-bit, we denote the lower and upper bounds of x as l and u, respectively. As shown in the Equation (1), there are three steps in tensor quantization: (1) truncate the tensor x into range $[l, u]$, denotes as x_c, (2) map the floating-point tensor x_c of the range $[l, u]$ to the integer tensor x_{int} of the range $[0, 2^N - 1]$, (3) reconstruct the floating point tensor x_q from the integer tensor x_{int}.

$$\begin{aligned} x_c &= \text{Clamp}(x, l, u), \\ x_{int} &= \text{Round}(\frac{x_c - l}{u - l} \times (2^N - 1)), \\ x_q &= x_{int} \times \frac{(u - l)}{2^N - 1} + l, \end{aligned} \tag{1}$$

where $\text{Clamp}(x, l, u) = \min(\max(x, l), u)$, $\text{Round}(\cdot)$ outputs the nearest integer around the input. Usually, the post-training quantization aims to get the clipping values ($\{l_w, u_w\}$ and $\{l_a, u_a\}$) of weights and activations for all the layers without the modification to original weights.

3.1. Analysis of quantization on super resolution

In this section, we investigate the challenge of quantization on super resolution tasks by conducting experiments with different bit-width settings. As shown in Table 2, when we only quantize the weights of EDSR model to 6-bit, there is only a little performance drop (-0.059/-0.002) compared with the full-precision model. On the contrast, when we only quantize the activations, the result on Urban100 decreases much severely, which indicates that the quantization on activations greatly degrade the performance of low-bit precision models.

Then we further visualize the activation distribution of intermediate layers to find out the problems that cause the performance degradation of quantized models. As shown in Figure 2, we conclude three properties that are detrimental to quantization:

Table 2. The quantization results of EDSR$\times$4 on Urban100 .

W/A	32/32	6/32	32/6	6/6
PSNR/SSIM	26.646/0.804	26.587/0.802	25.894/0.773	25.890/0.773

(1) Long-tailed. all the distributions show dense in the middle yet sparse in the tail, with most values distributed in small ranges while the total range is always much larger. As shown in Figure 2b, the activation range of body.11.conv1 with sample 2 is $[-232, 210]$, while about 99.16% of values are in $[-30, 25]$, which would lead to large quantization error in dense region for the uniform quantizer.

(2) Asymmetric. The skewness is a measure for the asymmetry of the probability distribution about its mean [7], for an asymmetric distribution, negative skew indicates that the longer tail is on the left side of the distribution, and positive skew indicates that the longer tail is on the right. As we can see that the skewness of the activation distributions show that they are much asymmetric on two tails, such as 3.78 and 1.73 for the body.0.conv1 in Figure 2a, which is much unfriendly to the calibration of zero point using the conventional quantization methods.

(3) Highly-dynamic. Without batch normalization, there is a high range flexibility of activations in super resolution networks. For instance, the activation range of body.31.conv1 with sample 2 is the $2.19\times$ of the sample 1 as shown in Figure 2d. High dynamic range means that the optimal quantization values vary for different samples. Therefore, determining the optimal clipping values that could fit the super resolution dataset is a non-trivial task.

In conclusion, the reasons that cause the performance degradation for quantized SR models are the long-tailed, asymmetric and highly-dynamic activation distribution. To address that, we propose a coarse-to-fine quantization

method for super resolution as Figure 1, the first step is the coarse clipping to get the rough bounds, and the second step is the fine-grained calibration to get the optimal ones. We will introduce the details in the following sections.

3.2. Density-based dual clipping

The distribution of activations in SR models usually shows to be dense in the middle yet sparse in the tail, so that the dense region is far away from the original boundaries, which is very unfriendly to model quantization, especially for low bits. DDBT [52] clips the range with the 1-th and 99-th percentiles of activations to initialize the clipping bounds. However, This method do not consider the asymmetric distribution for image super resolution. Therefore, we propose the density-based dual clipping (DBDC) to cut off outliers of activations, help narrow the distribution to a valid range as shown in Figure 1.

Different from the unilateral clipping [5, 25, 40, 52], our proposed DBDC takes full account of the imbalanced distribution of left and right parts for the SR model. Take the range clipping of one layer as an example, we first divide the original activation x into the N equal interval based on its minimum and maximum value as Equation (2).

$$\begin{aligned} \Delta &= (\max(x) - \min(x))/N, \\ H(p) &= \sum_{i \in x} \mathbb{I}(i{>}p \;\&\; i{<}p + \Delta), \end{aligned} \tag{2}$$

where $H(p)$ denotes the density of the position of p. Based on the assumption that the lower the density of the bound, the less the importance, we aims to keep the regions that contain most density. As shown in Figure 2, the $H(p)$ is usually smaller in the two tails and bigger in the middle. Therefore, in order to get the appropriate lower and upper clipping bounds, we start to search the optimal lower clipping value and upper clipping value (donated as l_a and u_a) from two tails. By comparing the density values between the position of l_a and u_a iteratively, we make the clipping position with smaller density closer to the middle, which can be formulated as:

$$l_a^t, u_a^t = \begin{cases} l_a^{t-1} + \Delta, u_a^{t-1}, & H(l_a^{t-1}) < H(u_a^{t-1}) \\ l_a^{t-1}, u_a^{t-1} - \Delta, & H(l_a^{t-1}) \geq H(u_a^{t-1}) \end{cases}, \tag{3}$$

where the t denotes the iterative step. The termination condition for one batch of calibration samples is that the density in the clipping region accounts for more than the clipping ratio (denoted as $M\%$). After obtaining the appropriate lower and upper clippings for this calibration samples, then we input the next batch of calibration samples. In the meanwhile, the global bounds l_a and u_a are updated by the exponential moving average (EMA) method [10].

$$\begin{aligned} l_a &= \beta \cdot l_a + (1 - \beta) \cdot l_a^T, \\ u_a &= \beta \cdot u_a + (1 - \beta) \cdot l_a^T, \end{aligned} \tag{4}$$

Algorithm 1: Density-based Dual Clipping

Input: Full-precision SR model F of K layers, calibration dataset D, clipping ratio M, the number of bins N.

Output: $\{l_a, u_a\}^K$.

Initialize $\{l_a, u_a\}^K$ with the minimum and maximum values of feature maps.
foreach $d = D_0, D_1, ..., D_n$ **do**
  foreach $k = 1$ *to* K **do**
    $H(x_1), ..., H(x_N) \leftarrow \text{Histogram}(F^k(d))$
    $l \leftarrow \min(F^k(d)), u \leftarrow \max(F^k(d))$
    $\Delta \leftarrow (u - l)/N, \;\; S \leftarrow \sum_{i=x_i}^{x_N} H(i)$
    while $\sum_{i=l}^{u} H(i)/S \geq 1 - M$ **do**
      if $H(l) < H(u)$ **then**
        $l = l + \Delta$
      else
        $u = u - \Delta$
      end
    end
    $l_a^k = \beta \cdot l_a^k + (1 - \beta) \cdot l$
    $u_a^k = \beta \cdot u_a^k + (1 - \beta) \cdot u$
  end
end

where the β is the hyper-parameter of weighting decrease, and the l_a^T and l_a^T denotes the clipping values for current batch of calibration sample. The procedure of the proposed DBDC is summarized in Algorithm 1.

3.3. Pixel-aware calibration

With the coarse tuning of the density-based dual clipping (DBDC), we get initial lower and upper bounds $(l_a, u_a)^K$ for the model of K layers, in which the weights and activations are not quantized. Then we further propose a pixel-aware calibration (PaC) method to finetune these clipping parameters for fitting the highly-dynamic feature maps of different samples at the given bit-width setting.

With the unlabeled calibration images and full-precision pretrained model, we can get outputs and the middle feature maps for different layers, which could provide abundant supervision information for the quantized model. Thus, we can build a dataset for finetuning, the pair of input and label for the i-th sample could be represented as:

$$(\text{input}, \text{label})^i = (x^i, (F_1^i, F_2^i, ..., F_N^i, O^i)), \tag{5}$$

where F_n^i denotes the feature map of the n-th residual block and O^i denotes the output of the full-precision model for the i-th sample, thus we create a finetune dataset with only 100 pairs of input-label.

With the limitation of the number of calibration images,

the finetune of this method only focus on the clipping parameters, thus keeping the number of parameters and samples similar for avoiding over-fitting. The supervision contains output and feature maps, we adopt the conventional ℓ_1 loss as Equation (6) for the output:

$$L_o = \frac{1}{H_o \cdot W_o \cdot C_o} ||O - O_q||_1, \tag{6}$$

where $||\cdot||$ denotes the $\mathcal{L}_1$ norm, the (H_o, W_o, C_o) denotes the height, width and channel number of the output shape respectively, and O and O_q represent the outputs of full-precision model and quantized model. For the supervision of feature maps, inspired by [47, 51], we propose the pixel transfer loss to calculate the distance of the intermediate outputs of full-precision model and quantized model. We first apply $\mathcal{L}_2$ norm on the feature maps:

$$\hat{F}_i = \frac{F_i}{||F_i||_2}, \quad \hat{F}_{q_i} = \frac{F_{q_i}}{||F_{q_i}||_2}, \tag{7}$$

where the F_i and F_{q_i} represent the output of the $i-$th residual block of full-precision model and quantized model. And then we calculate the mean square error of these two feature map of all the blocks, the pixel transfer loss is as following:

$$L_{pt} = \frac{1}{B} \sum_{i}^{N} \frac{1}{H_i \cdot W_i \cdot C_i} ||\hat{F}_i - \hat{F}_{q_i}||_2, \tag{8}$$

where (H_i, W_i, C_i) denotes the height, width and channel number of the output shape of the i-th residual block respectively, and B denotes the block number. This similar technique is also used in QAT methods [25, 52], called structured knowledge transfer, in which the feature maps are transformed with a spatial mapping firstly. Different from that, our proposed pixel transfer loss does not extract the spatial attention, but directly aligns the error of each pixel of the quantized model and the corresponding full-precision model, which is much benefit to post-training quantization with a few unlabeled images. Then we can get the total loss:

$$L_{PaC} = L_o + \lambda L_{pt}, \tag{9}$$

where λ is the hyper-parameter to balance these two losses, which is set to 5 in our experiments. With the minimization of total loss that takes full account of the reconstructed loss and cumulative error of quantization, the quantized model tends to imitate the full-precision model and attempt to find the clipping parameters that are best adapted to the highly-dynamic distributions.

To stabilize the finetune process, we further propose to iteratively optimize the clipping parameters of weights and activations instead of finetuning them together. As shown in Figure 1, we firstly freeze the clipping parameters of activations, finetune those of weights with the total loss. The gradient calculation of these parameters are similar as [5, 25], and we also approximate $\partial \mathrm{w}_q / \partial \mathrm{w}$ with 1 by using straight-through estimator [6] as previous methods. Then we freeze the clipping parameters of weights, finetune those of activations with the same loss function. The calculation of the gradient is the same as that of weights. This iterative optimization repeats circularly until it reaches the calibration epochs, in which the original weights are not updated.

4. Experiments

4.1. Experimental setup

Datasets and evaluation metrics. Following the existing quantization methods [25, 40, 52] for super resolution, we conduct the experiments on the DIV2K [37] dataset. Different from QAT that needs complete training dataset, we randomly choose 100 images as the calibration set, in which the HR images are not used. And the test datasets in our experiments are Set5 [2] with 5 images, Set14 [48] with 14 images, BSD100 [33] with 100 images and Urban100 [17] with 100 images. For the evaluation metrics, we adopt Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [43] on the Y channel between the reconstructed image and the corresponding HR image.

Implementation details. The SR models are EDSR [28] and SRResNet [24] with upscaling factors of 2 and 4, the pretrained parameters are trained based on the open-source code. All the layers are quantized in our experiments, where the first layer and the last layer are always quantized to 8-bit. For DBDC, we set the epoch number to 1 with batch size of 16. The clipping ratios are set to $4e-5$, $5e-5$ and $1e-5$ for the 4, 6 and 8-bit quantization, and the smoothing parameters β is set to 0.9. The samples during clipping are shuffled, and the number of equal bins are 2048. For EDSR, we do not clip the weight and adopt the minimum and maximum as the lower bound and upper bound, and we clipping the weight with ratio of 1e-4 for SRResNet, which will work better. For PaC, the total epoch is 10 with batch size of 2, the initial learning rates for clipping values of weights and activations are set to 0.001 and 0.05, respectively. The optimizer we adopt is Adam [22] and the learning rate scheduler is CosineAnnealingLR [30]. The baselines that we choose include two kinds of post-training quantization methods, the first one is the commercial quantization toolkits for existing AI accelerate devices, including OpenVINO [11], TensorRT [39] and SNPE [18], the unspported bit-width (4 and 6-bit) are simulated with MQbench [27], the other is the classical post-training methods, contain MSE [4], Percentile [26] and MinMax [20], which are widely used in image classification, the calibration epoch is set to 20 with batch size of 16. All the experiments are conducted with PyTorch [34].

Table 3. PSNR(dB)/SSIM comparisons between existing post-training quantization methods and ours on EDSR of scale 4 and scale 2. Bit denotes the bit-width of weights and activations.

Method	Bit	Set5 (×4)	Set14 (×4)	BSD100 (×4)	Urban100 (×4)	Set5 (×2)	Set14 (×2)	BSD100 (×2)	Urban100 (×2)
Baseline	32	32.485/0.899	28.815/0.788	27.721/0.742	26.646/0.804	38.193/0.961	33.948/0.920	32.352/0.902	32.967/0.936
Bicubic	32	28.420/0.810	26.000/0.703	25.960/0.668	23.140/0.658	33.660/0.930	30.24/0.869	29.560/0.843	26.880/0.840
OpenVINO [11]	8	32.148/0.892	28.629/0.782	27.572/0.735	26.454/0.796	32.148/0.892	28.629/0.782	27.572/0.735	26.454/0.796
TensorRT [39]	8	32.329/0.895	28.711/0.784	27.639/0.738	26.548/0.799	37.880/0.958	33.774/0.917	32.217/0.899	32.764/0.933
SNPE [18]	8	32.329/0.896	28.707/0.786	27.646/0.740	26.551/0.800	37.786/0.957	33.751/0.917	32.189/0.898	32.733/0.932
MSE [4]	8	32.191/0.897	28.524/0.785	27.539/0.740	26.341/0.799	37.781/0.960	33.349/0.919	32.114/0.901	32.237/0.934
Percentile [26]	8	32.306/0.897	28.642/0.785	27.630/0.739	26.310/0.796	38.041/0.960	33.686/0.910	32.256/0.901	32.690/0.934
MinMax [20]	8	32.350/0.896	28.730/0.785	27.654/0.740	26.560/0.800	37.983/0.959	33.832/0.918	32.260/0.900	32.719/0.934
Ours	8	**32.460/0.898**	**28.763/0.787**	**27.695/0.741**	**26.567/0.802**	**38.120/0.960**	**33.850/0.920**	**32.313/0.901**	**32.810/0.935**
OpenVINO [11]	6	30.283/0.843	27.426/0.735	26.592/0.687	25.214/0.740	34.337/0.907	31.436/0.860	30.236/0.833	30.172/0.878
TensorRT [39]	6	30.696/0.851	27.719/0.744	26.765/0.694	25.459/0.749	34.735/0.913	31.778/0.867	30.472/0.841	30.582/0.887
SNPE [18]	6	30.493/0.839	27.599/0.735	26.664/0.685	25.386/0.742	34.305/0.903	31.499/0.858	30.249/0.831	30.336/0.877
MSE [4]	6	30.648/0.879	27.593/0.771	26.881/0.725	25.256/0.773	35.746/0.950	32.163/0.909	31.231/0.909	30.302/0.917
Percentile [26]	6	31.496/0.875	28.188/0.768	27.213/0.720	25.890/0.773	36.610/0.944	32.890/0.904	31.599/0.885	31.666/0.917
MinMax [20]	6	31.073/0.863	27.986/0.760	27.011/0.713	25.643/0.713	36.037/0.936	32.544/0.897	31.286/0.878	31.208/0.908
Ours	6	**32.300/0.894**	**28.653/0.784**	**27.627/0.738**	**26.382/0.797**	**37.896/0.958**	**33.675/0.918**	**32.186/0.899**	**32.452/0.932**
OpenVINO [11]	4	20.526/0.542	18.949/0.475	18.636/0.439	18.418/0.467	24.157/0.606	22.642/0.559	22.346/0.543	22.083/0.589
TensorRT [39]	4	21.343/0.512	19.809/0.461	19.495/0.423	19.100/0.450	23.897/0.608	22.325/0.571	22.208/0.553	22.068/0.600
SNPE [18]	4	21.417/0.472	20.035/0.413	19.925/0.392	19.320/0.406	23.284/0.548	22.086/0.522	22.215/0.517	21.873/0.555
MSE [4]	4	24.600/0.737	24.365/0.668	24.343/0.635	22.183/0.649	28.813/0.855	27.898/0.827	27.706/0.813	25.714/0.826
Percentile [26]	4	26.570/0.696	24.834/0.620	24.173/0.576	22.871/0.608	29.803/0.788	27.992/0.758	27.187/0.736	26.514/0.766
MinMax [20]	4	23.132/0.635	21.208/0.569	23.266/0.508	20.220/0.554	28.005/0.744	25.960/0.703	24.684/0.682	24.717/0.725
Ours	4	**31.203/0.867**	**27.977/0.760**	**27.085/0.714**	**25.556/0.764**	**36.327/0.942**	**32.753/0.904**	**31.477/0.884**	**30.900/0.913**

4.2. Results and analysis

The results are shown in Table 3, Table 4 and Table 5. Each model is conducted with multiple configurations, contains two upscaling factors (×2 and ×4) and three bit-width (4, 6 and 8-bit). For reference, we also list the results of bicubic interpolation.

Evaluation on EDSR. Table 3 shows the quantitative results of EDSR, we can see that the existing post-training methods could achieve a not bad results when quantized to 8-bit, but cause great performance drop when further compress to lower bit-width. For the 4-bit, the best baseline of upscaling 4 is the MSE, causes 4.255 dB, 2.828 dB, 3.378 dB and 4.463 dB PSNR drop on these four test sets, even worse than bicubic interpolation. In contrast, with our proposed method, the quantized model can achieve a much better results. When quantized to 8-bit, we could achieve $4\times$ compression ratio with negligible performance drop under two upscaling factors, and when quantized to 4-bit, our method only drop about 1 dB PSNR on upscaling 4, much better than bicubic interpolation and significantly outperform the existing post-training quantization methods. Besides, we find that the model of upscaling 2 shows much sensitive to all the quantization setting than upscaling 4. The conventional quantization method even drop 10 dB on Set5 when quantizd to 4-bit. With our proposed DBDC and PaC, the performance degradation could be reduce within 2 dB, and we could achieve 31.357 dB, only drop 0.884 dB compared with the full-precision model, and much better than bicubic interpolation with 29.650 dB.

Evaluation on SRResNet. As shown in Table 4, for the upscaling of 4, the existing post-training quantization baselines drop within 0.2 dB when quantized to 8-bit, but the performance degradation get much larger when quantized to 6-bit and 4-bit. Contrastively, our proposed method could reduce the performance drop within 0.05 dB when quantized to 8-bit, about 0.2 dB and 1 dB when quantized to 6-bit and 4-bit, all the quantized models with our method could significantly outperform bicubic interpolation and baselines by a large margin. For the upscaling of 2, our method could also perform better on different bit-width settings. When we quantize the model to 4-bit, only get 1.604 dB, 1.348 dB, 0.984 dB and 2.471 dB PSNR drop on these four test sets, outperform the best baseline method (MSE) by 5.248 dB, 3.298 dB, 2.88 dB and 3.52 dB, respectively, and also surpass the bicubic interpolation. The quantization sensitivity of model with upscaling 2 shows the similar as EDSR model, but we still could help reduce the performance drop compared to the corresponding full-precision model.

Comparison with QAT. To further show the effectiveness of our proposed method, we also compare with the existing quantization aware training works [25, 40], in which the parameters are fixed after finetuning instead of the dynamic quantization [14, 52]. Besides, we also implement QAT on our quantized model with only 10 epochs and batch size of 16, in which our PTQ provides the initial parameters. As shown in Table 5, compared with PAMS [25], QAT with

Table 4. PSNR(dB)/SSIM comparisons between existing post-training quantization methods and ours on SRResNet of scale 4 and scale 2. Bit denotes the bit-width of weights and activations.

Method	Bit	Set5 (×4)	Set14 (×4)	BSD100 (×4)	Urban100 (×4)	Set5 (×2)	Set14 (×2)	BSD100 (×2)	Urban100 (×2)
Baseline	32	32.234/0.896	28.656/0.784	27.630/0.738	26.229/0.791	38.091/0.961	33.752/0.919	32.241/0.900	32.367/0.931
Bicubic	32	28.420/0.810	26.000/0.703	25.960/0.668	23.140/0.658	33.660/0.930	30.240/0.869	29.560/0.843	26.880/0.840
OpenVINO [11]	8	32.003/0.890	28.505/0.778	27.509/0.732	26.039/0.783	37.451/0.955	33.350/0.912	31.978/0.895	31.978/0.924
TensorRT [39]	8	32.013/0.891	28.507/0.779	27.508/0.733	26.069/0.785	37.506/0.956	33.428/0.913	31.984/0.895	32.026/0.925
SNPE [18]	8	32.120/0.893	28.556/0.781	27.562/0.736	26.111/0.788	37.734/0.957	33.529/0.915	32.085/0.896	32.100/0.927
MSE [4]	8	32.006/0.892	28.387/0.779	27.469/0.734	25.910/0.784	37.737/0.958	33.247/0.915	31.972/0.897	31.665/0.926
Percentile [26]	8	32.092/0.893	28.492/0.780	27.525/0.735	26.046/0.786	37.739/0.958	33.414/0.916	32.058/0.897	31.965/0.927
MinMax [20]	8	31.984/0.891	28.495/0.779	27.503/0.733	26.057/0.785	37.539/0.956	33.413/0.913	31.992/0.895	32.020/0.925
Ours	8	**32.207/0.895**	**28.619/0.783**	**27.618/0.738**	**26.191/0.790**	**38.032/0.960**	**33.648/0.919**	**32.212/0.900**	**32.210/0.930**
OpenVINO [11]	6	30.080/0.835	27.348/0.727	26.665/0.683	24.861/0.721	33.539/0.884	31.007/0.849	30.050/0.827	29.505/0.857
TensorRT [39]	6	29.990/0.828	27.277/0.724	26.553/0.681	24.782/0.719	33.634/0.885	30.923/0.846	30.011/0.827	29.270/0.854
SNPE [18]	6	29.650/0.814	27.112/0.714	26.449/0.671	24.690/0.710	33.120/0.874	30.501/0.834	29.654/0.813	28.895/0.842
MSE [4]	6	30.822/0.872	27.642/0.760	27.002/0.718	25.003/0.752	36.010/0.944	32.099/0.898	31.174/0.881	29.935/0.904
Percentile [26]	6	30.970/0.869	27.874/0.760	27.085/0.715	25.340/0.756	35.826/0.936	32.314/0.893	31.192/0.874	30.707/0.902
MinMax [20]	6	30.725/0.859	27.784/0.750	26.987/0.704	25.233/0.744	34.964/0.919	31.895/0.877	30.755/0.856	30.286/0.886
Ours	6	**32.089/0.892**	**28.504/0.779**	**27.561/0.733**	**26.011/0.783**	**37.811/0.959**	**33.295/0.916**	**32.068/0.898**	**31.719/0.926**
OpenVINO [11]	4	24.316/0.573	23.201/0.519	23.276/0.500	21.614/0.528	24.415/0.535	23.570/0.508	23.551/0.502	22.942/0.556
TensorRT [39]	4	23.729/0.461	22.648/0.402	22.808/0.389	21.089/0.399	24.769/0.535	23.753/0.502	23.733/0.491	22.753/0.526
SNPE [18]	4	23.130/0.413	22.317/0.376	22.404/0.358	20.793/0.371	24.111/0.505	23.297/0.477	23.195/0.464	22.452/0.511
MSE [4]	4	27.979/0.784	25.828/0.680	25.704/0.641	23.042/0.639	31.239/0.870	29.106/0.828	28.470/0.801	26.376/0.804
Percentile [26]	4	27.283/0.699	25.411/0.625	25.329/0.603	22.990/0.605	27.369/0.703	26.477/0.689	26.180/0.668	24.866/0.686
MinMax [20]	4	26.639/0.654	25.122/0.599	25.107/0.577	22.746/0.573	25.824/0.603	25.302/0.602	25.191/0.584	23.914/0.606
Ours	4	**31.146/0.878**	**27.889/0.763**	**27.152/0.718**	**25.133/0.753**	**36.487/0.951**	**32.404/0.904**	**31.357/0.885**	**29.896/0.904**

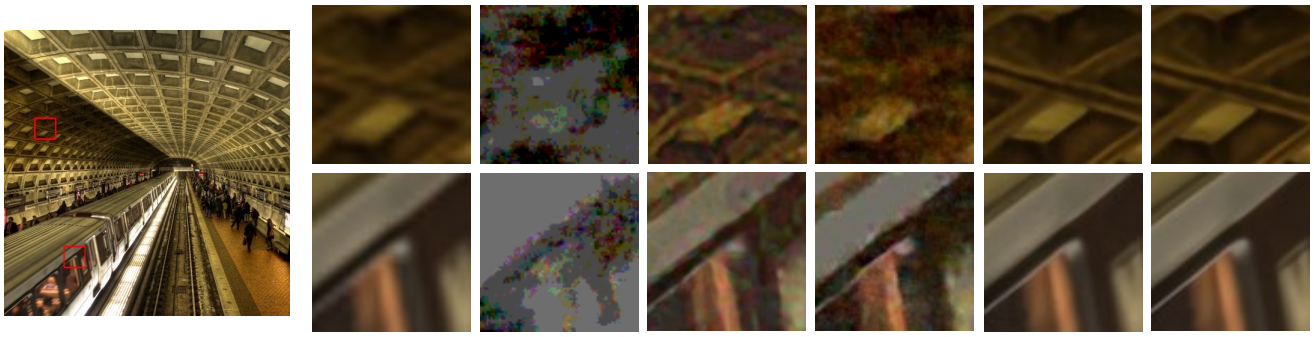


Figure 3. Visual results of different methods on 4-bit EDSR models with upscaling of 4. The metrics below are PSNR(dB) and SSIM.

our method could achieve better results when keeping the same setting of partial quantization(only residual blocks), get 0.19 dB drop for upscaling 4 and 0.201 dB drop for upscaling 2 on Set5, while PAMS gets 0.514 dB and 0.320 dB drop. When quantizing all the layers, QAT with the proposed method also could get much better results compared with FQSR [40], get 0.844 dB drop for upscaling 4 and 0.632 dB drop for upscaling 2 on Set5, while FQSR gets 1.08 dB and 0.847 dB drop. This experiment proves that our proposed method could provide a better initialization to help the faster convergence of QAT process and achieve excellent performance.

Visualization. Figure 3 shows the visual results of different quantization methods. For reference, we also show the bicubic interpolation image. Compared with existing PTQ methods and bicubic scaling, the reconstructed HR image of our proposed method could achieve outstanding performances, provide more texture and details.

4.3. Ablation studies

To demonstrate the effectiveness of density-based dual clipping (DBDC) and pixel-aware calibration (PaC), we conduct the experiments with a vanilla post-training quantization based MinMax method, and then add these two techniques gradually. The results are shown in Table 6, we can see that when using the conventional post-training quan-

Table 5. PSNR(dB) and SSIM comparisons between existing quantization-aware training methods and the proposed method on EDSR of scale 4 and scale 2 with 4-bit quantization. Bit denotes the bit-width of weights and activations, FQ denotes full-quantization and QAT denotes quantization aware training. QAT with the initialization of our method only run 10 epochs in our experiments.

Method	Scale	Bit	FQ	QAT	Set5	Set14	BSD100	Urban100
PAMS [25]	×4	32			32.095/0.894	28.576/0.781	27.562/0.736	26.035/0.785
		4	×	✓	31.591/0.885	28.199/0.773	27.322/0.728	25.321/0.762
	×2	32			37.985/0.960	33.568/0.918	32.155/0.899	31.977/0.927
		4	×	✓	37.665/0.959	33.196/0.915	31.936/0.897	31.100/0.919
FQSR [40]	×4	32			32.007/0.892	28.486/0.778	27.528/0.731	25.934/0.781
		4	✓	✓	30.928/0.870	27.816/0.761	27.073/0.715	24.927/0.744
	×2	32			37.885/0.958	33.425/0.915	32.106/0.897	31.777/0.924
		4	✓	✓	37.038/0.951	32.835/0.908	31.668/0.889	30.646/0.911
Ours	×4	32			32.485/0.899	28.815/0.788	27.721/0.742	26.646/0.804
		4	×	×	32.105/0.891	28.563/0.781	27.553/0.714	26.051/0.787
		4	×	✓	32.295/0.895	28.576/0.784	27.558/0.738	26.232/0.794
		4	✓	×	31.203/0.867	27.977/0.760	27.085/0.714	25.556/0.764
		4	✓	✓	31.641/0.881	28.217/0.772	27.332/0.727	25.748/0.777
	×2	32			38.193/0.961	33.948/0.920	32.352/0.902	32.967/0.936
		4	×	×	37.837/0.958	33.662/0.917	32.146/0.898	32.335/0.930
		4	×	✓	37.992/0.960	33.838/0.919	32.205/0.900	32.545/0.933
		4	✓	×	36.327/0.942	32.753/0.904	31.477/0.884	30.900/0.913
		4	✓	✓	37.561/0.955	33.442/0.915	31.992/0.896	31.725/0.924

Table 6. The ablation studies of the proposed method on EDSR of scale 4, the results represent the PSNR(dB) and SSIM.

DBDC	PaC	Set5	Set14	BSD100	Urban100
		26.570/0.696	24.834/0.620	24.173/0.576	22.871/0.608
✓		30.406/0.838	27.510/0.735	26.633/0.687	25.312/0.736
	✓	28.000/0.775	26.002/0.681	25.406/0.630	24.116/0.669
✓	✓	31.203/0.867	27.977/0.760	27.085/0.714	25.556/0.764

tization method, it causes much performance degradation compared with corresponding full-precision model, 5.664 dB, 3.822 dB, 3.457 dB and 3.358 dB drop on these four test sets. When only adding the density-based dual clipping to the baseline, it could help improve 3.836 dB, 2,676 dB, 2.46 dB and 2.441 dB, respectively, which shows that DBDC could cut out the outliers and narrow the distribution to a valid range, help reduce the quantization error. When only adding the pixel-aware calibration to baseline, it could also help improve the performance, but still worse than the baseline only with DBDC for the reason that the calibration dataset is too small to provide adequate information for the quantized models to get the accurate clipping values with PaC. Combining the proposed two techniques (DBDC first and then PaC) boosts the PSNR by 4.663 dB, 3.143 dB, 2.912 dB and 2.658 dB compared with the full-precision model, which proves that firstly coarsely clipping outliers and then finetune the clipping values to the optimal parameters could truly improve the performance of post-training quantization for image super resolution.

5. Conclusion

This paper studies the post-training quantization for image super resolution with a few unlabeled calibration images, which could help the fast deployment on mobile devices. Our analysis indicates that the long-tailed, asymmetric distributions and highly dynamic ranges of activations greatly degrade the performance of quantized models. To alleviate that, we propose a coarse-to-fine post-training quantization framework for super resolution. With the density-based dual clipping (DBDC), we could get the initial lower and upper bounds of asymmetric activations for SR models and cut off outliers, and then fine-grained optimize them with a novel pixel aware calibration (PaC) method with the supervision of the full-precision model, accommodate the highly dynamic range of different samples. Extensive experiments demonstrate that the proposed method could significantly outperform existing PTQ algorithms on various models and datasets, and we also show that our method could provide a better initialization for quantization-aware training methods.

Besides, other pixel2pixel tasks are similar to the SR task for that they all remove the BN layers to ensure range flexibility and reduce artifacts. We will further conduct convincing experiments and explore the application of this method to other pixel2pixel tasks in the future work.

Acknowledgments We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Andreas Aakerberg, Kamal Nasrollahi, and Thomas B Moeslund. Real-world super-resolution of face-images from surveillance cameras. *IET Image Processing*, 16(2):442–452, 2022. 1

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 5

[3] Wei-Hao Chen, Chunmeng Dou, Kai-Xiang Li, Wei-Yu Lin, Pin-Yi Li, Jian-Hao Huang, Jing-Hong Wang, Wei-Chen Wei, Cheng-Xin Xue, Yen-Cheng Chiu, et al. Cmos-integrated memristive non-volatile computing-in-memory for ai edge processors. *Nature Electronics*, 2(9):420–428, 2019. 1

[4] Jungwook Choi, Pierce I-Jen Chuang, Zhuo Wang, Swagath Venkataramani, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Bridging the accuracy gap for 2-bit quantized neural networks (qnn). *arXiv preprint arXiv:1807.06964*, 2018. 5, 6, 7

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 4, 5

[6] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 5

[7] Susan Dean and Barbara Illowsky. Descriptive statistics: skewness and the mean, median, and mode. *Connexions website*, 2018. 3

[8] Tingxing Tim Dong, Hao Yan, Mayank Parasar, and Raun Krisch. Rendersr: A lightweight super-resolution model for mobile gaming upscaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3087–3095, 2022. 1

[9] Zongcai Du, Jie Liu, Jie Tang, and Gangshan Wu. Anchor-based plain net for mobile image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2494–2502, 2021. 1

[10] Tony Finch. Incremental calculation of weighted mean and variance. *University of Cambridge*, 4(11-5):41–42, 2009. 4

[11] Yury Gorbachev, Mikhail Fedorov, Iliya Slavutin, Artyom Tugarev, Marat Fatekhov, and Yaroslav Tarkan. Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 5, 6, 7

[12] Hayit Greenspan. Super-resolution in medical imaging. *The computer journal*, 52(1):43–63, 2009. 1

[13] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 1

[14] Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq: Content-aware dynamic quantization for image super-resolution. *arXiv preprint arXiv:2207.10345*, 2022. 1, 6

[15] Cheeun Hong, Heewon Kim, Sungyong Baik, Junghun Oh, and Kyoung Mu Lee. Daq: Channel-wise distribution-aware quantization for deep image super-resolution networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2675–2684, 2022. 1, 2

[16] Zejiang Hou and Sun-Yuan Kung. Efficient image super resolution via channel discriminative deep neural network pruning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3647–3651. IEEE, 2020. 1

[17] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 5

[18] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 5, 6, 7

[19] Andrey Ignatov, Radu Timofte, Maurizio Denna, and Abdel Younes. Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2525–2534, 2021. 1

[20] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. 5, 6, 7

[21] Xinrui Jiang, Nannan Wang, Jingwei Xin, Keyu Li, Xi Yang, and Xinbo Gao. Training binary neural network without batch normalization for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1700–1707, 2021. 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Zhiqiang Lang, Lei Zhang, and Wei Wei. E2fif: Push the limit of binarized deep imagery super-resolution using end-to-end full-precision information flow. *arXiv preprint arXiv:2207.06893*, 2022. 2

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 5

[25] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams: Quantized super-resolution via parameterized max scale. In *European Conference on Computer Vision*, pages 564–580. Springer, 2020. 1, 2, 4, 5, 6, 8

[26] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2810–2819, 2019. 5, 6, 7

[27] Yuhang Li, Mingzhu Shen, Jian Ma, Yan Ren, Mingxin Zhao, Qi Zhang, Ruihao Gong, Fengwei Yu, and Junjie Yan. Mqbench: Towards reproducible and deployable model quantization benchmark. *arXiv preprint arXiv:2111.03759*, 2021. 5

[28] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 5

[29] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *European Conference on Computer Vision*, pages 41–55. Springer, 2020. 1

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[31] Tao Lu, Jiaming Wang, Yanduo Zhang, Zhongyuan Wang, and Junjun Jiang. Satellite image super-resolution via multi-scale residual deep neural network. *Remote Sensing*, 11(13):1588, 2019. 1

[32] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[33] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 5

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[35] M Dirk Robinson, Stephanie J Chiu, Cynthia A Toth, Joseph A Izatt, Joseph Y Lo, and Sina Farsiu. New applications of super-resolution in medical imaging. In *Super-resolution imaging*, pages 383–412. CRC Press, 2017. 1

[36] Jacob Shermeyer and Adam Van Etten. The effects of super-resolution on object detection performance in satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[37] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 5

[38] Zhijun Tu, Xinghao Chen, Pengju Ren, and Yunhe Wang. Adabin: Improving binary neural networks with adaptive binary sets. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 379–395. Springer, 2022. 1

[39] Han Vanholder. Efficient inference with tensorrt. In *GPU Technology Conference*, volume 1, page 2, 2016. 5, 6, 7

[40] Hu Wang, Peng Chen, Bohan Zhuang, and Chunhua Shen. Fully quantized image super-resolution networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 639–647, 2021. 1, 2, 4, 5, 6, 7, 8

[41] Yanbo Wang, Shaohui Lin, Yanyun Qu, Haiyan Wu, Zhizhong Zhang, Yuan Xie, and Angela Yao. Towards compact single image super-resolution via contrastive self-distillation. *arXiv preprint arXiv:2105.11683*, 2021. 1

[42] Yu Emma Wang, Gu-Yeon Wei, and David Brooks. Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701*, 2019. 1

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[44] Jingwei Xin, Nannan Wang, Xinrui Jiang, Jie Li, Heng Huang, and Xinbo Gao. Binarized neural network for single image super resolution. In *European conference on computer vision*, pages 91–107. Springer, 2020. 2

[45] Songhua Liu Jingwen Ye Xinchao Wang Xingyi Yang, Daquan Zhou. Deep model reassembly. *NeurIPS*, 2022. 1

[46] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 73–91. Springer, 2022. 1

[47] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 5

[48] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 5

[49] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010. 1

[50] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Aligned structured sparsity learning for efficient image super-resolution. *Advances in Neural Information Processing Systems*, 34:2695–2706, 2021. 1

[51] Youpeng Zhao, Huadong Tang, Yingying Jiang, Qiang Wu, et al. Lightweight vision transformer with cross feature attention. *arXiv preprint arXiv:2207.07268*, 2022. 5

[52] Yunshan Zhong, Mingbao Lin, Xunchao Li, Ke Li, Yunhang Shen, Fei Chao, Yongjian Wu, and Rongrong Ji. Dynamic dual trainable bounds for ultra-low precision super-resolution networks. *arXiv preprint arXiv:2203.03844*, 2022. 1, 2, 4, 5, 6