Outlier-Aware Post-Training Quantization for Image Super-Resolution

Hailing Wang* Jianglin Lu Yitian Zhang Yun Fu

Northeastern University, USA

{wang.haili, lu.jiang}@northeastern.edu, markcheung9248@gmail.com, yunfu@ece.neu.edu

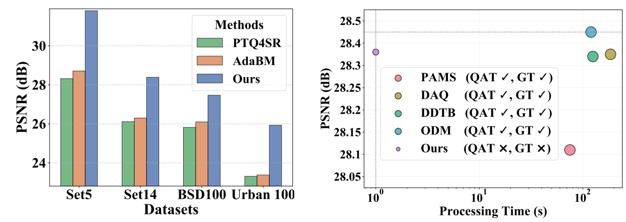
Abstract

Quantization techniques, including quantization-aware training (QAT) and post-training quantization (PTQ), have become essential for inference acceleration of image super-resolution (SR) networks. Compared to QAT, PTQ has garnered significant attention as it eliminates the need for ground truth and model retraining. However, existing PTQ methods for SR often fail to achieve satisfactory performance as they overlook the impact of outliers in activation. Our empirical analysis reveals that these prevalent activation outliers are strongly correlated with image color information, and directly removing them leads to significant performance degradation. Motivated by this, we propose a dual-region quantization strategy that partitions activations into an outlier region and a dense region, applying uniform quantization to each region independently to better balance bit-width allocation. Furthermore, we observe that different network layers exhibit varying sensitivities to quantization, leading to different levels of performance degradation. To address this, we introduce sensitivity-aware finetuning that encourages the model to focus more on highly sensitive layers, further enhancing quantization performance. Extensive experiments demonstrate that our method outperforms existing PTQ approaches across various SR networks and datasets, while achieving performance comparable to QAT methods in most scenarios with at least a 75 speedup.

1. Introduction

The goal of image super-resolution (SR) is to enhance image resolution, often by factors of 4× or more, while preserving content and detail. Although deep learning-driven SR models have attained superior results [6, 27, 48], these advancements come at the cost of increased parameter counts. With the growing demand for deploying SR models on edge devices and handling larger input sizes, there is an increasing need for models that balance both parameter efficiency and computational speed. To mitigate this issue, a range of compression techniques has been studied, including distillation [23, 39, 51], pruning [42, 43, 47], quantization [35, 36], and efficient module design [29, 46]. In this paper, we focus on image SR quantization, which not only reduces memory consumption but also significantly improves inference speed.

*Corresponding author.



(a) Comparison with PTQ methods in PSNR using RDN under W4A4. (b) Comparison with QAT methods in PSNR using EDSR under W4A4.

Figure 1. Comparison of our method with SOTA PTQ and QAT baselines. In (b), GT denotes ground truth, the bubble size indicates the amount of training data required, and performance is averaged across four datasets.

The goal of model quantization is to shrink the network's parameters and activations (feature maps) from high precision to a compact representation while maintaining its original performance. Current quantization approaches are typically grouped into quantization-aware training (QAT) [16, 28, 32] and post-training quantization (PTQ) [7, 25, 31, 44], distinguished by whether they require retraining network weights. QAT requires labeled data pairs and model retraining to adapt to the quantization process, whereas PTQ applies quantization without weight updates, making it a more source-efficient alternative to QAT.

In image SR, quantization has been predominantly explored through QAT [13, 21, 28, 52], with only a few PTQ methods [10, 36] proposed. This is primarily because PTQ, particularly in activation quantization, suffers from greater performance degradation compared to QAT. To mitigate this issue, Tu *et al.* [36] introduce the first PTQ method for SR, employing a density-based double cropping technique to constrain the activation distribution within a manageable range. However, directly clipping activations outside the selected range may lead to significant performance degradation. Subsequently, Hong et al. [10] propose a dynamic quantization technique that adjusts bit allocation based on input variations. This is motivated by the observation that assigning different bit widths to various input images and network layers improves quantization performance. While effective, ensuring hardware compatibility for dynamic quantization remains a challenge.

In this paper, we first analyze the activation distributions of SR models and observe that activation outliers are prevalent across various networks (see Figure 3). To investigate their impact, we compare the visual quality of images with and without outliers. Our empirical findings reveal that outliers are strongly correlated with image color information, and directly removing them leads to noticeable color shifts and significant performance degradation (see Figure 2). This underscores the importance of preserving outliers in quantization to maintain color fidelity and enhance overall quantization performance. However, retaining outliers using existing methods consumes a substantial portion of the bit width allocated for normal activations, severely compromising quantization effectiveness. To address this, we propose a dual-region quantization strategy that partitions activations into an outlier region and a dense region. We then apply uniform quantization to each region independently, ensuring a more balanced bit-width allocation between these two regions. Furthermore, we observe that different network layers exhibit varying sensitivity to quantization, as evidenced by the extent of performance degradation when each layer is quantized individually. Based on this insight, we propose a sensitivity-aware loss that encourages the model to focus more on highly sensitive layers, further enhancing overall quantization performance.

Figure 1 presents a comparison of our approach against state-of-the-art (SOTA) PTQ and QAT methods on representative SR networks. Figure 1a shows that our method consistently outperforms PTQ baselines across various datasets. From Figure 1b, our method achieves performance comparable to QAT methods, despite not requiring retraining or ground truth, while providing significantly higher efficiency. To summarize, our core contributions are:

- Our empirical analysis reveals that activation outliers are strongly correlated with color information, and removing them leads to significant color shifts in generated images.
- We identify an allocation trade-off between outliers and normal activations: clipping outliers causes severe performance degradation, while retaining them consumes the bit width allocated for normal ones. To address this, we quantize outliers and normal activations separately, ensuring a more balanced and effective bit-width allocation.
- We uncover that different layers exhibit varying sensitivities to quantization and propose a sensitivity-aware loss function to focus more on highly sensitive layers.
- Comprehensive evaluations show that the proposed approach exceeds existing SOTA PTQ baselines and achieves performance comparable to QAT methods, while delivering a 75 $\times$ speedup.

2. Related Works

2.1. Efficient Image Super-Resolution

Efficient SR models fall into several categories, including architectural design, neural architecture search (NAS), knowledge distillation (KD), pruning, and quantization. For efficient architectural design, Ahn *et al.* [2] introduce cascading residual connections and efficient residual blocks to construct a compact SR network. Sun *et al.* [33] propose an efficient feature modulation that combines CNN-like efficiency with transformer adaptability. Regarding NAS-based methods, Huang *et al.* [14] present a differentiable NAS strategy to identify efficient SR networks, integrating both unit-level and network-level search spaces to optimize SR quality. For KD-based approaches, Zhang *et al.* [50] introduce a novel data-free knowledge distillation framework for SR, which is adaptable to various teacher-student configurations. In pruning-based solutions, Wang *et al.* [38] develop a SR network with sparse masks that simultaneously exploit spatial and channel dimensions to jointly identify and remove unnecessary computation at a fine-grained level. In this paper, we focus on quantization-based methods for image SR, as they effectively reduce memory consumption while significantly improving inference speed.

2.2. Quantization for Image Super-Resolution

Quantization methods, including QAT [12, 13, 18, 21, 30, 37, 41, 52] and PTQ [10, 36], have both been explored for image SR. The first QAT-based SR work [21] introduces a trainable truncation parameter to adaptively constrain the quantization range, motivated by the observation that SR models without batch normalization typically exhibit a large dynamic range. Wang *et al.* [37] propose a quantizer with learnable margins, enabling adaptability to variation in weights and activations from one layer to another. To further address dynamic range issues, Hong *et al.* [13] develop a channel distribution-aware quantization scheme. Additionally, some approaches [12, 34] employ dynamic quantization strategies with adaptive bit-width allocation for different inputs and layers. However, ensuring hardware compatibility for dynamic quantization remains an open challenge. In contrast, PTQ for image SR has received significantly less attention. The first PTQ-based SR method [36] introduces a density-based dual clipping and pixel-aware calibration to optimize the quantization parameters. Subsequently, Hong *et al.* [10] introduce a dynamic quantization method with adaptive bit mapping. While effective to some extent, these methods largely overlook the impact of outliers, resulting in suboptimal quantization performance. In this work, we emphasize the importance of outliers in quantization, revealing that outliers are strongly correlated with image color information.

3. Methodology

3.1. Preliminaries

Quantization reduces parameter precision, thereby shrinking memory footprints and accelerating inference. The process of quantizing a floating-point tensor x into a b-bit unsigned integer can be formally described as follows:

$$x_{\text{int}} = \left\lfloor \frac{clamp(x, l, u) - l}{u - l} \times (2^b - 1) \right\rceil, \tag{1}$$

where l and u are the lower and upper bounds of x respectively, $clamp(x, l, u) = \min(\max(x, l), u)$ restricts x within the bounds l and u, and the function $\lfloor \cdot \rceil$ outputs the nearest integer of the input. The quantized floating-point value x_q can be reconstructed from x_{int} within the integer space to approximate the original value x via:

$$x_q = x_{\text{int}} \cdot \frac{u - l}{2^b - 1} + l. \tag{2}$$

When x exhibits a (approximately) symmetric distribution around zero, the bounds u and l can be set to symmetric limits, such as u and $-u$. This adjustment simplifies the quantization process by ensuring symmetry around zero.

Due to the highly asymmetric distribution of activations in SR networks [12, 13, 34, 52] and the relatively low sensitivity of weights to quantization [36], an asymmetric quantizer is typically applied to activations, whereas a symmetric uniform quantizer is utilized for weights. Compared to weight quantization, previous studies [13, 36] have shown that activation quantization is the primary cause of performance degradation in SR models. Therefore, this paper also focuses on activation quantization of SR models.

3.2. Observation & Motivation

Observation 1. In Figure 3, we illustrate the activation distribution of different samples at the same layer (`body.15.conv1`) of the EDSR network. It is evident that all samples contain outliers in their distributions. Most activation values are concentrated within a shallow range (e.g., $[-50, 50]$), which we refer to as the *dense region*. Outliers, on the other hand, are located beyond this region, forming what we call the *outlier region*. Notably, the bounds of the outlier region vary significantly across different samples. For instance, the left bound for sample 1 is -192, whereas for sample 2, it is -273. From this observation, two key questions arise: Do these outliers influence the quality of the restored images, and what specific features do they correspond to in the generated images?

To answer these questions, we clip the outliers (1% of activations) in the feature map and visualize the resulting images in Figure 2. The visual comparison clearly shows that removing outliers leads to noticeable color distortion in both global and local regions of the images, such as faded flower colors. *This observation indicates that activation outliers are closely linked to image color information and should be preserved during the quantization process.* Based on this insight, we propose outlier-aware quantization to minimize quantization errors in Section 3.3.

Observation 2. We further analyze quantization error across different layers by independently quantizing activations in each layer of the EDSR and SRResNet networks. To evaluate quantization performance, we compute the average PSNR between the quantized images and the ground truth (GT) across 100 randomly selected image pairs from the DIV2K dataset [1]. As shown in Figure 4, different network layers exhibit varying degrees of sensitivity to quantization. While some layers experience significant performance degradation, others remain robust to quantization. For instance, in SRResNet, the `head.0` layer suffers a substantial drop in PSNR, plunging from 32.06 dB to 18.26 dB. In contrast, certain layers, such as `body.4.conv1`, maintain high performance with PSNR values of up to 31.20 dB. *Motivated by this observation, we propose focusing more attention on highly sensitive layers in quantization rather than distributing equal attention across all layers.* To achieve this, we introduce a sensitivity-aware loss in Section 3.4.

3.3. Piecewise Linear Quantizer

As demonstrated in Observation 1, preserving activation outliers is crucial for retaining image information. However, retaining outliers during quantization will consume a substantial portion of the bit width allocated for normal activations, reducing the representation space available for them. Inspired by [7], to achieve a balanced bit-width allocation between outliers and normal activations, we propose a dual-region quantization strategy that partitions activations into two distinct, non-overlapping regions and designs piecewise linear quantization to quantize each region independently. This method preserves the unique characteristics of both normal activations and outliers.

Specifically, given an asymmetric activation range $R = [l_a, u_a]$, where l_a and u_a denote the lower and upper activation bounds, we introduce a learnable breakpoint bp to divide the range R into a symmetric dense region $R_1 = [-bp, bp]$, which contains most normal activations, and an outlier region $R_2 = R_2^- \cup R_2^+ = [l_a, -bp) \cup (bp, u_a]$, which captures the extreme values in the activation distribution. Our piecewise linear quantizer converts a floating-point ten-

Figure 2. After clipping 1% of activation outliers in the full-precision model, the outputs (bottom) exhibit noticeable color distortions compared to the original ones (top), affecting both global regions and detail-rich local areas of the images.

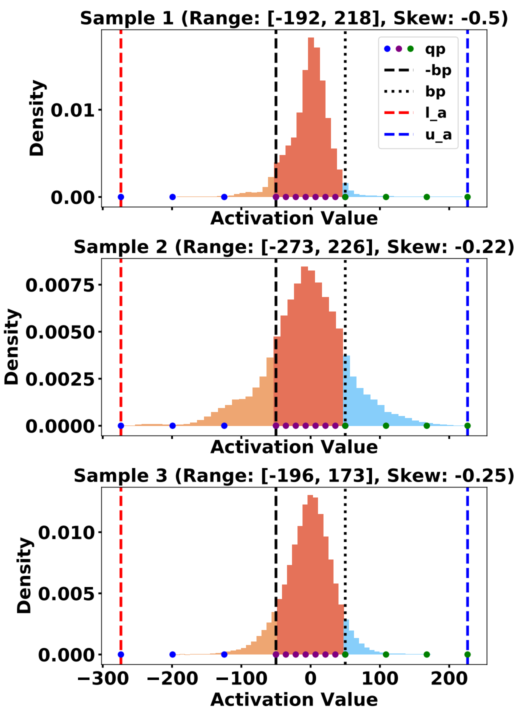


Figure 3. The activation distributions of three different samples at the same layer (`body.15.conv1`) in EDSR exhibit variations in range (`Range`) and skewness (`Skew`). These distributions are divided by a breakpoint bp into a dense region $[-bp, bp]$ and an outlier region $[l\_a, -bp) \cup (bp, u\_a]$, both of which undergo uniform quantization to the corresponding quantization points qp.

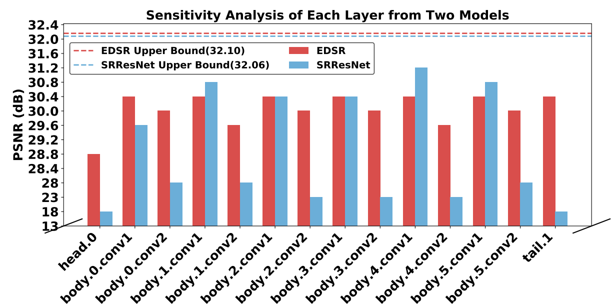


Figure 4. Performance comparison of 4-bit quantization applied individually to each layer of EDSR and SRResNet. Certain layers show a notable drop compared to the upper bound (full-precision performance), indicating higher sensitivity to quantization.

sor x into a b-bit integer representation via:

$$x_{\text{int}} \leftrightarrow \begin{cases} \left\lfloor \frac{clamp(x, -bp, bp)}{2bp} \times \left(2^{b-1} - 1\right) \right\rceil, & x \in R_1 \\ \left\lfloor \frac{clamp(x, l_a, -bp) - l_a}{-bp - l_a} \times \left(2^{b-2} - 1\right) \right\rceil, & x \in R_2^- \\ \left\lfloor \frac{clamp(x, bp, u_a) - bp}{u_a - bp} \times \left(2^{b-2} - 1\right) \right\rceil, & x \in R_2^+ \end{cases} \quad (3)$$

where values in both regions are quantized at the same bit level. We determine appropriate values for l_a, u_a, and bp through a statistical analysis of a calibration set. Specifically, for the first batch, we initialize l_a as the minimum activation value, u_a as the maximum activation value, and bp as the 99th percentile of the activation values. For weight quantization, we set the upper bound of weights u_w as the maximum absolute value in each layer. In subsequent batches, these parameters are updated using an exponential moving average [8].

3.4. Sensitivity-Aware Finetuning

While static statistical analysis on calibration data provides initial estimates for parameters l_a, u_a, u_w, and bp across different layers, the substantial variation in outlier ranges across samples within the same layer may affect the accuracy of these estimates. To address this, we refine these quantization parameters by finetuning the model on the calibration data. Inspired by Observation 2, which highlights that different layers exhibit varying sensitivities to quantization, we design a layer-specific loss function and perform sensitivity-aware finetuning. This strategy directs the model to focus more on highly sensitive layers during quantization rather than distributing equal attention across all layers, thus enhancing the model's adaptability to the dynamic nature of activation distributions across different samples.

To quantify the sensitivity of each layer to quantization, we pass the calibration data $\mathcal{D}_{\text{cal}}$ through a full-precision SR network $\mathcal{K}$ and compute the mean variance of the feature maps. We use this variance as an indicator of quantization sensitivity, where higher variance corresponds to greater sensitivity. The layer-wise quantization sensitivity s_k is defined as:

$$s_k = \frac{\exp\left(\frac{1}{N}\sum_{x\in D_{\text{cal}}}\sigma(x_k)\right)}{\sum_{j=1}^{K}\exp\left(\frac{1}{N}\sum_{x\in D_{\text{cal}}}\sigma(x_j)\right)}, \tag{4}$$

where $\sigma(x_k)$ represents the standard deviation of the feature map x_k at the k-th layer, N is the total number of batches in the calibration dataset, and K is the total number of layers in the SR network. We optimize the quantization parameters using a training loss L_{all}, which consists of a reconstruction loss $\mathcal{L}_{\text{rec}}$ and a sensitivity-aware loss $\mathcal{L}_{\text{sen}}$, formulated as:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{K}(I_{lr}^{i}) - \mathcal{Q}(I_{lr}^{i})\right\|_1, \tag{5}$$

$$\mathcal{L}_{\text{sen}} = \frac{s_k}{K}\sum_{k=1}^{K}\left\|\frac{F_{\mathcal{K}}^{k}}{\|F_{\mathcal{K}}^{k}\|_2} - \frac{F_{\mathcal{Q}}^{k}}{\|F_{\mathcal{Q}}^{k}\|_2}\right\|_2, \tag{6}$$

$$L_{\text{all}} = \mathcal{L}_{\text{sen}} + \lambda\mathcal{L}_{\text{rec}}, \tag{7}$$

where λ is a balancing parameter, I_{lr}^{i} denotes the i-th low-resolution input image, $\mathcal{K}$ and $\mathcal{Q}$ represent the pre-trained full-precision and quantized networks, respectively. $F_{\mathcal{K}}^{k}$ and $F_{\mathcal{Q}}^{k}$ denote the feature outputs at the k-th layer of $\mathcal{K}$ and $\mathcal{Q}$, respectively. Notably, our approach requires only low-resolution images for computing L_{all}, eliminating the need for ground-truth high-resolution images. This further enhances the practicality of our method.

During the fine-tuning phase, we update the quantization parameters in a staged manner for progressive optimization. Specifically, in the first epoch, we update u_w while keeping all other parameters fixed. In the next epoch, only l_a and u_a are updated, keeping the rest unchanged. And in the subsequent epoch, we update bp while keeping other parameters frozen. This cycle is repeated over multiple iterations to gradually refine the quantization parameters. The overall quantization process is summarized in Algorithm 1.

Algorithm 1: Quantization Algorithm

Input: Full-precision SR network $\mathcal{K}$ with K layers, calibration dataset $\mathcal{D}_{\text{cal}} = \{I_{lr}^{i}\}_{i=1}^{N}$, where N is the number of calibration batches
Output: Quantized network $\mathcal{Q}$

Calibration Phase:
for $i = 1, \dots, N$ **do**
  for $k = 1, \dots, K$ **do**
    if $i = 1$ **then**
      $u_w^k \leftarrow \max|W^k|$,
      $l_{a,1}^k, u_{a,1}^k \leftarrow \min(F^k(i)), \max(F^k(i))$,
      $bp_1^k \leftarrow \text{Perc}_{99}(F^k(i))$;
    else
      $l_{a,i}^k \leftarrow \beta \cdot l_{a,i-1}^k + (1-\beta)\cdot\min(F^k(i))$,
      $u_{a,i}^k \leftarrow \beta \cdot u_{a,i-1}^k + (1-\beta)\cdot\max(F^k(i))$,
      $bp_i^k \leftarrow \beta \cdot bp_{i-1}^k + (1-\beta)\cdot\text{Perc}_{99}(F^k(i))$;
    end
  end
  Obtain $\{s_k^i\}_{k=1}^{K}$ using Eq. (4);
end
Fine-tuning Phase: ;
for *epoch = 1, ..., #epochs* **do**
  if *epoch* mod $3 = 1$ **then**
    Update $\{u_w^k\}_{k=1}^{K}$ with Eq. (7);
  else if *epoch* mod $3 = 2$ **then**
    Update $\{l_a^k, u_a^k\}_{k=1}^{K}$ with Eq. (7);
  else
    Update $\{bp^k\}_{k=1}^{K}$ with Eq. (7);
  end
end

4. Experiments

4.1. Experimental Setup

In our experiments, we follow previous methods [10, 36] and build the calibration set by randomly sampling 100 low-resolution images from the DIV2K [1] training dataset, without including ground truth images. Following the setting in [10, 36], the test sets include Set5 [3], Set14 [45], BSD100 [26], and Urban100 [15]. We also consider larger datasets, including Test2K and Test4K [19], which are generated by downsampling the images in the DIV8K dataset [9]. We evaluate our method on representative SR networks, including EDSR [24], RDN [49], and SRResNet [20]. For EDSR network, we chose the model configuration that consists of 16 residual blocks with 64-channel dimensions. For evaluation, we calculate Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [40] between the quantized image and the corresponding high-resolution image on the Y channel.

In the calibration phase, we conduct calibration for one epoch using a batch size of $N = 16$. In the first batch, we employ min-max [17] to establish initial quantization ranges for weights and activations. The breakpoint is set by taking the 99th percentile values of the activations for each layer. In subsequent batches, the exponential moving average (EMA) hyperparameter β is set to a fixed value of 0.9. In the fine-tuning phase, we utilize Adam optimizer [5] to optimize the clipping ranges of weights, activations, and breakpoints over 10 epochs with a batch size of 2. We set the hyperparameter $\lambda = 5$. The initial learning rate is set to 0.001 and decays by a factor of 0.9.

For comparison with PTQ methods, we follow [36] and quantize all layers in the models for the Set5, Set14, BSD100, and Urban100 datasets, with the first and last lay-

Method	FT	W / A	Set5			Set14			BSD100			Urban100		
			FAB	PSNR	SSIM	FAB	PSNR	SSIM	FAB	PSNR	SSIM	FAB	PSNR	SSIM
EDSR [24]	–	32 / 32	32.0	32.10	0.894	32.0	28.58	0.781	32.0	27.56	0.736	32.0	26.04	0.785
EDSR-MSE [4]	×	6 / 6	6.0	31.84	0.887	6.0	28.37	0.775	6.0	27.45	0.731	6.0	25.73	0.775
EDSR-MinMax [17]	×	6 / 6	6.0	31.56	0.866	6.0	28.26	0.760	6.0	27.29	0.714	6.0	25.76	0.760
EDSR-Percentile [22]	×	6 / 6	6.0	24.30	0.793	6.0	24.31	0.728	6.0	24.68	0.700	6.0	21.93	0.696
EDSR-PTQ4SR [36]	✓	6 / 6	6.0	31.80	0.884	6.0	28.34	0.768	6.0	27.37	0.722	6.0	25.79	0.769
EDSR-AdaBM [10]	✓	6 / 6	5.7	31.92	0.887	5.6	28.47	0.777	5.4	27.47	0.731	5.7	25.89	0.778
EDSR-Ours	✓	6 / 6	6.0	32.03	0.891	6.0	28.55	0.780	6.0	27.54	0.735	6.0	25.99	0.782
EDSR-MSE [4]	×	4 / 4	4.0	27.74	0.827	4.0	26.03	0.734	4.0	25.95	0.702	4.0	23.63	0.712
EDSR-MinMax [17]	×	4 / 4	4.0	26.83	0.624	4.0	25.04	0.546	4.0	24.57	0.503	4.0	23.12	0.536
EDSR-Percentile [22]	×	4 / 4	4.0	24.03	0.776	4.0	23.95	0.712	4.0	24.42	0.687	4.0	21.62	0.677
EDSR-PTQ4SR [36]	✓	4 / 4	4.0	30.51	0.836	4.0	27.62	0.735	4.0	26.88	0.693	4.0	24.92	0.721
EDSR-AdaBM [10]	✓	4 / 4	3.8	31.02	0.860	3.7	27.87	0.751	3.5	26.91	0.700	3.7	25.11	0.736
EDSR-Ours	✓	4 / 4	4.0	31.54	0.879	4.0	28.26	0.769	4.0	27.36	0.726	4.0	25.61	0.765
RDN [49]	×	32 / 32	32.0	32.24	0.895	32.0	28.67	0.784	32.0	27.63	0.739	32.0	26.29	0.793
RDN-MSE [4]	×	6 / 6	6.0	31.02	0.879	6.0	27.77	0.767	6.0	27.01	0.724	6.0	25.01	0.757
RDN-MinMax [17]	×	6 / 6	6.0	30.59	0.863	6.0	27.54	0.752	6.0	26.65	0.703	6.0	24.79	0.733
RDN-Percentile [22]	×	6 / 6	6.0	18.87	0.778	6.0	18.33	0.667	6.0	19.88	0.651	6.0	16.81	0.632
RDN-PTQ4SR [36]	✓	6 / 6	6.0	30.73	0.877	6.0	27.60	0.765	6.0	26.85	0.720	6.0	25.08	0.756
RDN-AdaBM [10]	✓	6 / 6	5.7	31.56	0.881	5.6	28.14	0.769	5.5	27.20	0.722	5.7	25.31	0.755
RDN-Ours	✓	6 / 6	6.0	32.20	0.894	6.0	28.62	0.782	6.0	27.61	0.738	6.0	26.24	0.790
RDN-MSE [4]	×	4 / 4	4.0	25.55	0.831	4.0	24.33	0.725	4.0	24.49	0.689	4.0	21.75	0.692
RDN-MinMax [17]	×	4 / 4	4.0	25.91	0.632	4.0	24.22	0.549	4.0	24.29	0.530	4.0	22.24	0.523
RDN-Percentile [22]	×	4 / 4	4.0	18.83	0.771	4.0	18.28	0.662	4.0	19.83	0.646	4.0	16.77	0.625
RDN-PTQ4SR [36]	✓	4 / 4	4.0	28.32	0.813	4.0	26.11	0.709	4.0	25.82	0.671	4.0	23.31	0.668
RDN-AdaBM [10]	✓	4 / 4	3.8	28.71	0.808	3.7	26.30	0.707	3.6	26.10	0.672	3.8	23.38	0.663
RDN-Ours	✓	4 / 4	4.0	31.80	0.885	4.0	28.39	0.775	4.0	27.47	0.732	4.0	25.93	0.778

Table 1. Performance comparison of PTQ methods on W4A4 (4-bit weight and 4-bit activation) and W6A6 using EDSR and RDN as SR models, both with a scale factor of 4. We add an additional FAB (Feature Average Bit-width) column specifically for the AdaBM method, as it is an adaptive PTQ method. Red marks the highest quantization performance, while green denotes the runner-up.

Method	W / A	Test2K		Test4K	
		PSNR	SSIM	PSNR	SSIM
EDSR [24]	32 / 32	27.71	0.782	28.80	0.814
EDSR-PTQ4SR [36]	8 / 6	27.54	0.768	28.91	0.814
EDSR-AdaBM [10]	8 / 6	27.65	0.779	28.71	0.809
EDSR-Ours	8 / 6	27.59	0.773	28.95	0.819
EDSR-PTQ4SR [36]	4 / 4	26.94	0.723	28.13	0.767
EDSR-AdaBM [10]	4 / 4	27.40	0.758	28.39	0.784
EDSR-Ours	4 / 4	27.49	0.767	28.83	0.814
SRResNet [20]	32 / 32	27.64	0.781	28.72	0.813
SRResNet-PTQ4SR [36]	8 / 6	27.46	0.767	28.78	0.816
SRResNet-AdaBM [10]	8 / 6	27.55	0.777	28.62	0.809
SRResNet-Ours	8 / 6	27.55	0.771	28.84	0.818
SRResNet-PTQ4SR [36]	4 / 4	27.06	0.749	28.32	0.797
SRResNet-AdaBM [10]	4 / 4	27.31	0.766	28.25	0.782
SRResNet-Ours	4 / 4	27.35	0.768	28.80	0.812

Table 2. Performance comparison with PTQ methods using EDSR and SRResNet with a scale factor of 4 on larger datasets.

ers quantized at 8-bit precision. For the Test2K and Test4K datasets, we follow [10] and exclude the first and last layers from quatization. For comparison with QAT baselines, we follow [12, 21, 34] and also ignore the first and last layers in the quantization process.

4.2. Comparison with Post-Training Quantization

We compare the proposed method with existing PTQ approaches, including MSE [4], Percentile [22], MinMax [17], PTQ4SR [36], and AdaBM [10], across the EDSR [24], RDN [49] and SRResNet [20] networks. The quantitative results for a scale factor of 4 are presented in Table 1, while results for SRResNet are included in the supplementary appendix. As shown, our method consistently delivers the top results on every dataset. Notably, our approach demonstrates a greater advantage over existing approaches on detail-rich datasets and under challenging quantization settings. For instance, when quantizing the RDN model under the W4A4 setting on the Urban100 dataset, our method achieves a substantial PSNR improvement of 2.55 dB over the suboptimal method. Additionally, we observe that across all settings, the MinMax method significantly outperforms the Percentile method, highlighting the importance of preserving outliers in SR tasks. Table 2 presents the comparison results on the larger Test2K and Test4K datasets. As shown, our method consistently outperforms existing PTQ approaches, particularly in challenging settings such as W4A4, further demonstrating its robustness and effectiveness in low-bit quantization scenarios.

4.3. Comparison with Quantization-aware Training

To further demonstrate our method's effectiveness, we benchmark it against existing QAT baselines, including PAMS [21], DAQ [13], DDTB [52] and ODM [11], on

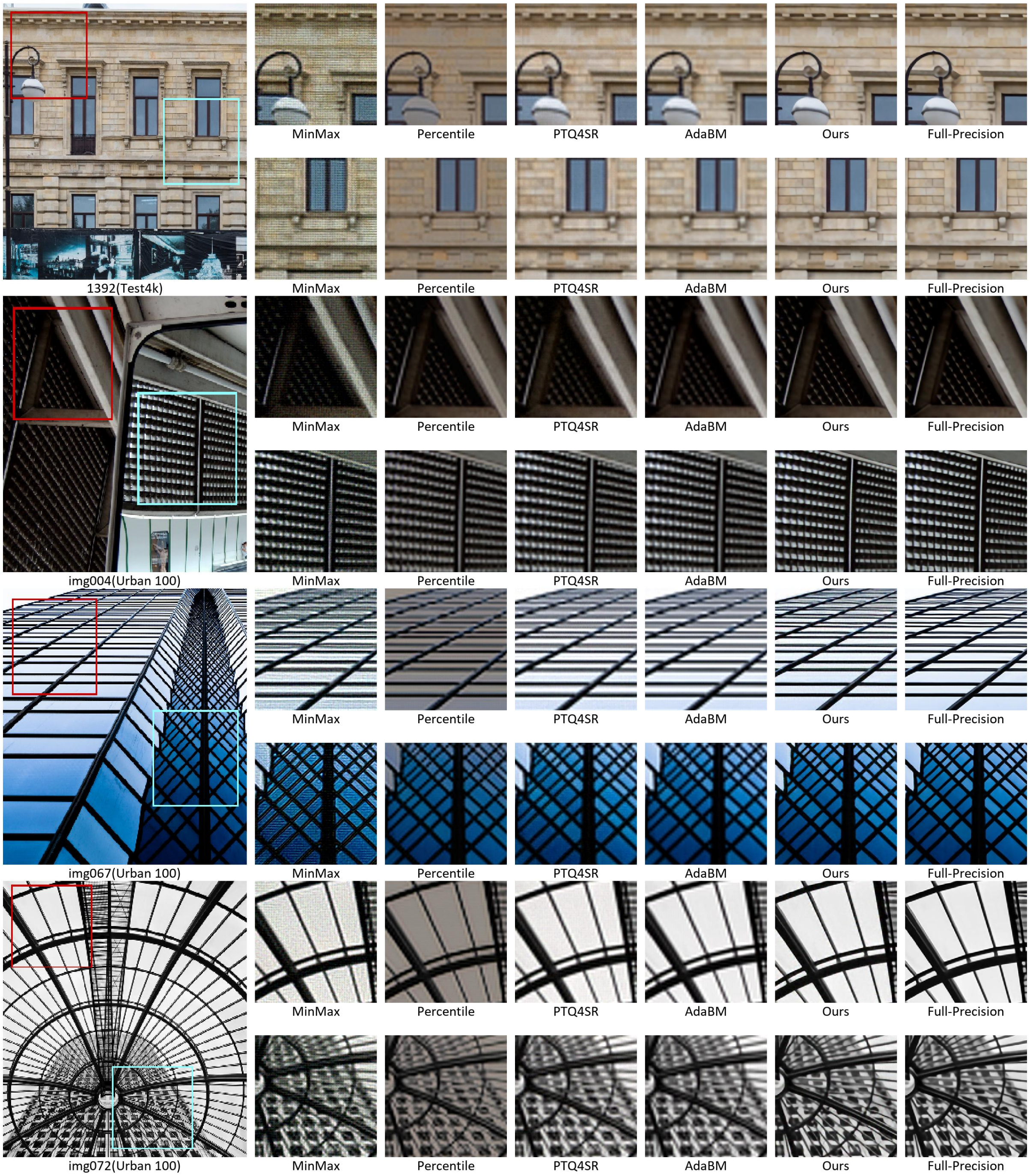


Figure 5. Visual comparison between different PTQ methods using RDN network under W4A4 setting. While baseline approaches suffer from different artifacts, our method effectively preserves the fine details across various scenarios.

the EDSR [24] networks. In practice, QAT methods typically require several hours for quantization due to the need for retraining model parameters. In contrast, our proposed method completes the quantization process in less than 2 minutes, significantly improving efficiency. Table 3 presents the comparison with a scale factor of 4. As shown, compared to QAT methods, our approach achieves at least a 75× speedup while achieving comparable performance without requiring ground truth supervision.

4.4. Qualitative Analysis

To provide a more intuitive assessment of performance, we present the SR results for each method in Figure 5. As illustrated, images produced by MinMax contain numer-

Method	QAT	GT	W / A	Process Time	Set5		Set14		BSD100		Urban100	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	-	✓	32/32	-	32.10	0.894	28.58	0.781	27.56	0.736	26.04	0.785
EDSR-PAMS	✓	✓	4/4	75×	31.59	0.885	28.20	0.773	27.32	0.728	25.32	0.762
EDSR-DAQ	✓	✓	4/4	185×	31.85	0.887	28.38	0.776	27.42	0.732	25.73	0.772
EDSR-DDTB	✓	✓	4/4	125×	31.85	0.889	28.39	0.777	27.44	0.732	25.69	0.774
EDSR-ODM	✓	✓	4/4	120×	32.00	0.891	28.47	0.779	27.51	0.735	25.80	0.778
EDSR-Ours	×	×	4/4	1×	31.79	0.885	28.40	0.778	27.45	0.731	25.75	0.773

Table 3. Comparison with QAT methods using EDSR network. Process time was measured on an NVIDIA GeForce RTX 2080Ti GPU.

PLQ	SAFT	VFT	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
×	×	×	26.83 / 0.624	25.04 / 0.546	24.57 / 0.503	23.12 / 0.536
✓	×	×	30.50 / 0.865	27.71 / 0.755	27.03 / 0.715	25.12 / 0.751
×	–	✓	29.45 / 0.770	26.95 / 0.677	26.27 / 0.632	24.40 / 0.654
×	✓	–	29.87 / 0.809	27.24 / 0.709	26.55 / 0.666	24.57 / 0.689
✓	✓	–	31.54 / 0.879	28.26 / 0.769	27.36 / 0.726	25.61 / 0.765

Table 4. Ablation study on EDSR network under W4A4 setting. PLQ, SAFT, VFT denotes Piecewise Linear Quantizer, Sensitivity-Aware Finetuning, Vanilla Finetuning, respectively.

ous noise artifacts, as preserving outliers in MinMax restricts the expressiveness of normal activations and distorts the image distribution with noise. In contrast, the images generated by Percentile exhibit significant color distortion, highlighting the importance of outliers in maintaining color fidelity. While PTQ4SR and AdaBM mitigate noise and color shift issues to some extent, they still introduce blurring in detail-rich areas, particularly in dense regions of images img004 and img072, leading to a noticeable decline in visual fidelity. By examining intricate textures and flat regions side by side, we find that our method effectively preserves fine details while avoiding noise and color distortion, demonstrating superior SR quality.

4.5. Ablation Study

To assess the efficacy of our proposed piecewise linear quantizer (PLQ) and sensitivity-aware finetuning (SAFT) strategies, we conduct an ablation study using MinMax as the baseline model and assess the impact of these two components. The results are presented in Table 4. As shown, combining both PLQ and SAFT (5th row) achieves the best performance. Compared to the baseline (1st row), applying PLQ alone (2nd row) leads to a significant PSNR improvement, with gains of 3.67 dB, 2.67 dB, 2.46 dB, and 2.00 dB on Set5, Set14, BSD100, and Urban100, respectively. Additionally, we observe that vanilla finetuning (VFT, 3rd row) performs worse than SAFT (4th row), which highlights the effectiveness of our proposed sensitivity-aware loss in improving quantization performance.

4.6. Resource Analysis

To validate the efficiency of our method, we compare its processing time, latency (a single forward-pass inference time), and storage with exiting PTQ baselines. As shown in Table 5, our method reduces processing time compared to PTQ4SR, while remaining comparable to AdaBM. In terms of latency, our approach performs similarly to PTQ4SR and is faster than AdaBM. Additionally, all methods maintain the same storage size. These results demonstrate that our performance improvements are achieved without increasing resource demands.

	Process Time	Latency	Storage size
EDSR-PTQ4SR	126 sec	133 ms	229.517K
EDSR-AdaBM	72 sec	143 ms	229.517K
EDSR-Ours	73 sec	135 ms	229.517K

Table 5. Efficiency comparison with PTQ methods using a scale factor of 4. Processing time and latency (fake quantization) are measured on an NVIDIA 2080Ti GPU.

5. Conclusion

This paper introduces an outlier-aware post-training quantization method for image super-resolution tasks. According to our empirical analysis on activation distribution, we observe that outliers in activations are both ubiquitous and impactful. We then conduct comparison experiments to investigate the impact of outliers and uncover that the outliers are strongly correlated with image color information. Specifically, simply removing outliers in activations will cause noticeable color distortion and considerable performance degradation. However, retaining them will reduce bit occupancy reserved for normal activations. To strike a balance between preserving outliers and maintaining quantization effectiveness on normal activations, we divide the activation distribution into two non-overlapping regions and apply uniform quantization to each region independently. Additionally, motivated by our finding that different network layers exhibit varying sensitivities to quantization, we design a sensitivity-aware loss function to make the model focus more on highly sensitive layers. We then conduct extensive experiments to demonstrate the effectiveness of our method, comparing it against both PTQ and QAT approaches across various datasets and model architectures.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017. 3, 5

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pages 252–268, 2018. 2

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

[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*, 2018. 6

[5] P Kingma Diederik. Adam: A method for stochastic optimization. *arXiv*, 2014. 5

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015. 1

[7] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. Post-training piecewise linear quantization for deep neural networks. In *ECCV*, pages 69–86, 2020. 1, 3

[8] Tony Finch. Incremental calculation of weighted mean and variance. *University of Cambridge*, 2009. 4

[9] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In *ICCVW*, pages 3512–3516, 2019. 5

[10] Cheeun Hong and Kyoung Mu Lee. Adabm: On-the-fly adaptive bit mapping for image super-resolution. In *CVPR*, pages 2641–2650, 2024. 1, 2, 5, 6

[11] Cheeun Hong and Kyoung Mu Lee. Overcoming distribution mismatch in quantizing image super-resolution networks. In *ECCV*, pages 380–396. Springer, 2024. 6

[12] Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq: Content-aware dynamic quantization for image super-resolution. In *ECCV*, pages 367–383, 2022. 2, 3, 6

[13] 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 *WACV*, pages 2675–2684, 2022. 1, 2, 3, 6

[14] Han Huang, Li Shen, Chaoyang He, Weisheng Dong, and Wei Liu. Differentiable neural architecture search for extremely lightweight image super-resolution. *IEEE TCSVT*, pages 2672–2682, 2022. 2

[15] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015. 5

[16] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. Efficient quantization-aware training with adaptive coreset selection. *arXiv preprint arXiv:2306.07215*, 2023. 1

[17] 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 *CVPR*, pages 2704–2713, 2018. 5, 6

[18] 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 *AAAI*, pages 1700–1707, 2021. 2

[19] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *CVPR*, pages 12016–12025, 2021. 5

[20] 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 *CVPR*, pages 4681–4690, 2017. 5, 6

[21] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams: Quantized super-resolution via parameterized max scale. In *ECCV*, pages 564–580, 2020. 1, 2, 6

[22] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, pages 2810–2819, 2019. 6

[23] Simiao Li, Yun Zhang, Wei Li, Hanting Chen, Wenjia Wang, Bingyi Jing, Shaohui Lin, and Jie Hu. Knowledge distillation with multi-granularity mixture of priors for image super-resolution. *arXiv preprint arXiv:2404.02573*, 2024. 1

[24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017. 5, 6, 7

[25] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *NeurIPS*, 2021. 1

[26] 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 *ICCV*, pages 416–423, 2001. 5

[27] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Humphrey Shi. Pyramid attention network for image restoration. *IJCV*, 2023. 1

[28] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. pages 16318–16330, 2022. 1

[29] Dongwon Park, Kwanyoung Kim, and Se Young Chun. Efficient module based single image super resolution for multiple problems. In *CVPR*, 2018. 1

[30] Haotong Qin, Yulun Zhang, Yifu Ding, Xianglong Liu, Martin Danelljan, Fisher Yu, et al. Quantsr: accurate low-bit quantization for efficient image super-resolution. *NeurIPS*, 2024. 2

[31] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, 2023. 1

[32] Mingzhu Shen, Feng Liang, Ruihao Gong, Yuhang Li, Chuming Li, Chen Lin, Fengwei Yu, Junjie Yan, and Wanli Ouyang. Once quantization-aware training: High performance extremely low-bit architecture search. In *ICCV*, 2021. 1

[33] Long Sun, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Spatially-adaptive feature modulation for efficient image super-resolution. In *ICCV*, pages 13190–13199, 2023. 2

[34] Senmao Tian, Ming Lu, Jiaming Liu, Yandong Guo, Yurong Chen, and Shunli Zhang. Cabm: Content-aware bit mapping for single image super-resolution network with large input. In *CVPR*, pages 1756–1765, 2023. 2, 3, 6

[35] Zhijun Tu, Xinghao Chen, Pengju Ren, and Yunhe Wang. Adabin: Improving binary neural networks with adaptive binary sets. In *ECCV*, 2022. 1

[36] Zhijun Tu, Jie Hu, Hanting Chen, and Yunhe Wang. Toward accurate post-training quantization for image super resolution. In *CVPR*, pages 5856–5865, 2023. 1, 2, 3, 5, 6

[37] Hu Wang, Peng Chen, Bohan Zhuang, and Chunhua Shen. Fully quantized image super-resolution networks. In *ACM MM*, pages 639–647, 2021. 2

[38] Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *CVPR*, pages 4917–4926, 2021. 2

[39] Yan Wang. Edge-enhanced feature distillation network for efficient super-resolution. In *CVPR*, 2022. 1

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, pages 600–612, 2004. 5

[41] Renjie Wei, Shuwen Zhang, Zechun Liu, Meng Li, Yuchen Fan, Runsheng Wang, and Ru Huang. Ebsr: enhanced binary neural network for image super-resolution. *arXiv preprint arXiv:2303.12270*, 2023. 2

[42] Bin Xia, Jingwen He, Yulun Zhang, Yitong Wang, Yapeng Tian, Wenming Yang, and Luc Van Gool. Structured sparsity learning for efficient video super-resolution. In *CVPR*, 2023. 1

[43] Lei Yu, Xinpeng Li, Youwei Li, Ting Jiang, Qi Wu, Haoqiang Fan, and Shuaicheng Liu. Dipnet: Efficiency distillation and iterative pruning for image super-resolution. In *CVPR*, 2023. 1

[44] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV*, 2022. 1

[45] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730, 2012. 5

[46] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *ECCV*, 2022. 1

[47] Yulun Zhang, Kai Zhang, Luc Van Gool, Martin Danelljan, and Fisher Yu. Lightweight image super-resolution via flexible meta pruning. 1

[48] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1

[49] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018. 5, 6

[50] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *CVPR*, pages 7852–7861, 2021. 2

[51] Yun Zhang, Wei Li, Simiao Li, Hanting Chen, Zhijun Tu, Wenjia Wang, Bingyi Jing, Shaohui Lin, and Jie Hu. Data upcycling knowledge distillation for image super-resolution. *arXiv preprint arXiv:2309.14162*, 2023. 1

[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. In *ECCV*, pages 1–18, 2022. 1, 2, 3, 6