Overcoming Distribution Mismatch in Quantizing Image Super-Resolution Networks

Cheeun Hong[1] and Kyoung Mu Lee[1,2]

[1] Dept. of ECE & ASRI, {cheeun914, kyoungmu}@snu.ac.kr
[2] IPAI, Seoul National University

Abstract. Although quantization has emerged as a promising approach to reducing computational complexity across various high-level vision tasks, it inevitably leads to accuracy loss in image super-resolution (SR) networks. This is due to the significantly divergent feature distributions across different channels and input images of the SR networks, which complicates the selection of a fixed quantization range. Existing works address this distribution mismatch problem by dynamically adapting quantization ranges to the varying distributions during test time. However, such a dynamic adaptation incurs additional computational costs during inference. In contrast, we propose a new quantization-aware training scheme that effectively **O**vercomes the **D**istribution **M**ismatch problem in SR networks without the need for dynamic adaptation. Intuitively, this mismatch can be mitigated by regularizing the distance between the feature and a fixed quantization range. However, we observe that such regularization can conflict with the reconstruction loss during training, negatively impacting SR accuracy. Therefore, we opt to regularize the mismatch only when the gradients of the regularization are aligned with those of the reconstruction loss. Additionally, we introduce a layer-wise weight clipping correction scheme to determine a more suitable quantization range for layer-wise weights. Experimental results demonstrate that our framework effectively reduces the distribution mismatch and achieves state-of-the-art performance with minimal computational overhead. Codes are available at https://github.com/Cheeun/ODM.

Keywords: Image super-resolution · Network quantization · Quantization-aware training

1 Introduction

Image super-resolution (SR) is a core low-level vision task aimed at reconstructing high-resolution (HR) images from their corresponding low-resolution (LR) counterparts. Recent advances in deep learning [7, 11, 31, 36, 37, 43, 45, 54, 55] have led to astonishing achievements in producing high-fidelity images. However, the remarkable performance is based on heavy network architectures that incur significant computational costs, limiting practical viability, such as mobile deployment.

To mitigate the computational complexity of neural networks, quantization has emerged as a promising avenue. Network quantization has proven effective in reducing computational costs without significant accuracy loss, particularly in high-level vision tasks,

Table 1: Quantization methods on SR. For performance, existing methods rely on channel-wise quantization or input-adaptive module, which incur computational overhead. Our method achieves high accuracy without utilizing channel-wise quantization or input-adaptive modules.

Method	Channel-wise Q	Input-adaptive Modules	PSNR	SSIM
PAMS [33]	✗	✗	29.51	0.835
DDTB [57]	✗	✓	30.97	0.876
DAQ [18]	✓	✓	31.01	0.871
Ours	✗	✗	**31.50**	**0.882**

such as image classification [8, 20, 58]. However, when it comes to quantizing SR networks to lower bit-widths, substantial performance degradation [24] often occurs, posing a persistent and challenging problem to be addressed.

This degradation can be attributed to the significant variance in the feature (activation) distributions of SR networks. The feature distribution of a layer exhibits substantial discrepancies across different channels and images, which makes it difficult to determine a single quantization range for a layer. Early approaches to SR quantization [33] adopt a training scheme to learn the quantization range of each layer. However, despite careful selection, the quantization ranges do not align with the varied values within the channel and image dimensions, which we refer to as *distribution mismatch* in features.

Recent approaches aim to address this issue by incorporating dynamic adaptation modules that accommodate the varying distributions. For example, the quantization range is dynamically adjusted by directly leveraging the distribution mean and variance at test time [18] or by employing input-adaptive dynamic modules [57]. Although adapting the quantization function to each image during inference might handle variable distributions, the dynamic modules introduce considerable computational overhead, potentially compromising the computational benefits of quantization.

In this study, we propose a novel quantization-aware training framework that addresses the distribution mismatch problem with a loss term that regulates the mismatch in distributions. Although directly minimizing the feature mismatch presents the potential for quantization-friendliness, whether it preserves reconstruction accuracy is questionable. We observe that concurrent optimization with the mismatch regularization and the original reconstruction loss can disrupt the image reconstruction process. Therefore, we introduce a cooperative mismatch regularization strategy, where the mismatch is regulated only when it collaborates harmoniously with the reconstruction loss. To determine the cooperative behavior, we assess the cosine similarity of the gradients from each loss, then we weigh the gradients of mismatch regularization based on this similarity. Consequently, we effectively update the SR network to hold both quantization-friendliness and reconstruction accuracy.

Furthermore, we identify the distribution mismatch among the weights of different layers. We discover that employing a fixed policy to determine the layer-wise weight quantization range [17,18,33,47] can be suboptimal and that a further precise range can be obtained by considering both the current distribution of weights and the distinct tendencies of each layer. Therefore, we additionally incorporate layer-specific variations using a correction parameter for each layer. This strategy allows us to accurately find

the quantization range for weights while incurring only a minimal overhead (0.01% additional storage size and no additional bitOPs)–a significantly smaller impact compared to methods that use dynamic modules. Overall, the contributions of our work include the following:

- We introduce the first quantization framework to address the distribution mismatch problem in SR networks without dynamic modules, as compared in Table 1. Our framework updates the SR network to be quantization-friendly and accurate simultaneously.
- Based on the observations of the distribution mismatch in SR networks, we effectively reduce the mismatch by introducing a cooperative mismatch regularization term and a weight clipping correction term.
- Compared to existing approaches to SR quantization, ours achieves state-of-the-art performance with similar or fewer computations.

2 Related Works

2.1 Image Super-Resolution

Convolutional Neural Network (CNN)-based approaches have demonstrated remarkable advancements in the image super-resolution (SR) task [32, 37], but at the cost of substantial computational resources. The intensive computations required by SR networks have spurred interest in developing lightweight SR architectures [10, 22, 23, 28, 54]. Furthermore, various strategies for lightweight SR networks have been explored, including neural architecture search [9,30,34,35,46], knowledge distillation [22,23,53], and pruning [26,42,56]. While these methods predominantly focus on reducing network depth or the number of channels, our work specifically aims to lower the precision of floating-point operations through network quantization.

2.2 Network Quantization

By mapping 32-bit floating point values of features and weights in convolutional layers to lower-bit representations, network quantization provides a dramatic reduction in computational resources [5, 8, 14, 29, 58, 59]. Recent works have successfully quantized various networks to low bit-widths with minimal compromise in accuracy [6, 12, 16, 27, 39, 49, 52]. However, these efforts primarily target high-level vision tasks, whereas networks for low-level vision tasks remain vulnerable to low-bit quantization.

2.3 Quantized Super-Resolution Networks

In contrast to high-level vision tasks, SR poses different challenges due to its inherent high sensitivity to quantization [24, 40, 48, 51]. Some works have attempted to recover accuracy by modifying the network architecture [2,25,51] or by assigning different bits for each image [17, 19, 47] or network stage [38]. However, the primary challenge in quantizing SR networks lies in the vastly distinct feature distributions. To address this, Li *et al*. [33] adopted a learnable quantization range for different layers. Subsequently,

recognizing that the distributions vary not only by layer, but also by channel and input, Hong *et al*. [18] introduced a channel-wise dynamic quantization function. Additionally, Zhong *et al*. [57] utilized an input-adaptive dynamic module to tailor quantization ranges for each specific input image. However, such dynamic adaptations of quantization functions during test-time incur non-negligible computational overheads. Instead of relying on test-time adaptive modules, our approach focuses on mitigating the feature mismatch before quantization. Our framework reduces the inherent distribution mismatch in SR networks with minimal overhead, accurately quantizing SR networks without dynamic modules. More recently, Qin *et al*. [44] introduced additional transformation functions in both the forward and backward processes. However, performance degradation is still evident in ultra-low bit (*e.g*., 2-bit) scenarios.

3 Proposed Method

In this section, after a brief introduction to network quantization (Section 3.1), we first analyze the mismatch in the features of SR networks (Section 3.2). Subsequently, we propose a solution to reduce this feature mismatch during training (Section 3.3). Additionally, we examine the mismatch in weights across SR networks (Section 3.4) and introduce a quantization range selection scheme that addresses the weight mismatch (Section 3.5). The overall training process is summarized in Algorithm 1.

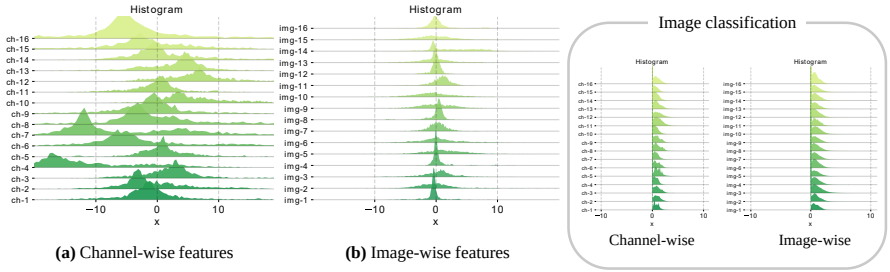
3.1 Preliminaries

To reduce the heavy computations of convolutional and linear layers in neural networks, the input features (activations) and weights of each convolution/linear layer are quantized to low-bit values [5,8,15,29]. Given the input feature of the i-th convolution/linear layer $\boldsymbol{X}_i \in \mathbb{R}^{B\times C\times H\times W}$, where B, C, H, and W represent the dimensions of the input batch, channel, height, and width, respectively, a quantization operator $q(\cdot)$ quantizes the feature $\boldsymbol{X}_i$ with bit-width b:

$$q(\boldsymbol{X}_i; l, u) = \text{Int}(\frac{\text{clip}(\boldsymbol{X}_i, l, u) - l}{s}) \cdot s + l, \tag{1}$$

where $\text{clip}(\cdot, l, u)$ truncates the input into the quantization range of $[l, u]$ and $s = \frac{u-l}{2^b-1}$. After truncation, the truncated feature is scaled to $[0, 2^b - 1]$, then rounded to integer values with $\text{Int}(\cdot)$, and rescaled to range $[l, u]$.

For activation quantization, the quantization range is defined by $[l_a, u_a]$. To obtain better quantization ranges in SR networks, the clipping parameters l_a, u_a for each layer are typically learned through quantization-aware training [33, 57]. Since the rounding function is not differentiable, a straight-through estimator (STE) [3] is used to train the clipping parameters in an end-to-end manner. Following [57], we initialize l_a and u_a as the (100-j)-th and j-th percentile values of the feature, averaged among the training data, where j is set to 99 in our experiments to avoid outliers that corrupt the quantization range. For weights, as their distributions tend to be symmetric and the mean approximates zero, we utilize a symmetric quantization function. Thus, for the weight $\boldsymbol{W}$ of each convolution/linear layer, the quantization range is defined by $[-u_w, u_w]$.



(a) Channel-wise features **(b)** Image-wise features

Figure 1: Distribution mismatch in SR networks. Compared to a classification network (*e.g.*, ResNet-18), an SR network (*e.g.*, EDSR) exhibits significant mismatches within the feature distributions across channel and image dimensions. The large distribution mismatch complicates the selection of an appropriate quantization range. Channels and images from the 2nd layer are randomly selected for visualization. Additional results are available in the supplementary material.

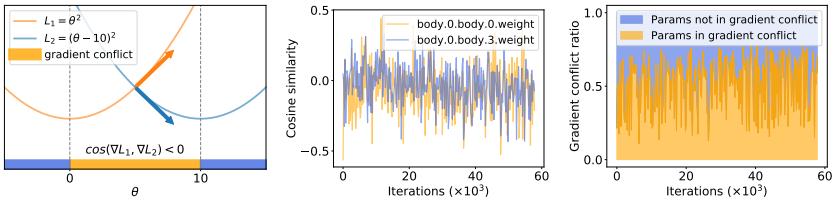
3.2 Distribution Mismatch in SR Networks

The unfriendliness to quantization in SR networks arises from diverse feature (activation) distributions, as reported in previous studies [18, 33, 57], primarily due to the absence of batch normalization layers in SR networks. As illustrated in Figure 1, where there are notable discrepancies between the channel and image distributions, quantization grids are unnecessarily allocated to regions with minimal feature density. Early SR quantization methods tackled this issue by employing learnable quantization range parameters [33] for each feature. However, even though the quantization-aware training process strives to find the optimal range for each feature, it fails to account for the channel-wise and input-wise variance in distributions. This mismatch results in a large quantization error that can impair SR performance. To deal with these distribution mismatches, existing methods adopt different quantization ranges for each channel [18] or input image [18, 57]. Nevertheless, the test-time adaptation modules used to determine these ranges introduce unwanted computational overhead during inference. Thus, our straightforward solution is to pre-adjust the distributions to be quantization-friendly, thereby eliminating the need for additional adaptations at inference. The following sections will introduce a new quantization-aware training scheme designed to resolve the distribution mismatch problem.

3.3 Cooperative Mismatch Regularization

Instead of trying to identify a quantization range capable of accommodating diverse feature distributions, our approach aims to regularize the distribution mismatch beforehand. Obtaining an appropriate quantization range for a feature with high image- and channel-wise variance is difficult, as a certain number of channels or images will invariably be distant from the selected quantization range. In this work, we refer to the total distance of each feature from the selected quantization grid as the mismatch,

$$M(\boldsymbol{X}_i) = ||\boldsymbol{X}_i - q(\boldsymbol{X}_i; l_a, u_a)||_2, \tag{2}$$



(a) Gradient conflict [13] **(b)** Cosine similarity of two gradients **(c)** Gradient conflict ratio

Figure 2: Conflict between mismatch regularization and reconstruction loss. Mismatch regularization updates a number of parameters in the *contradictory* direction to the reconstruction loss, which we refer to as gradient conflict. (b) When the two losses are jointly used, gradient conflict consistently occurs during training, outputting a negative cosine similarity value. (c) We plot the ratio of conflicted gradients during training. Nearly half of the parameters undergo gradient conflict, which indicates that merely combining mismatch regularization with the reconstruction loss can impair SR accuracy. Visualizations are done on EDSR.

where $||\cdot||_2$ calculates the Frobenius norm. Further analyses of the definition of mismatch are provided in the supplementary material. We can reduce the overall mismatch by directly regularizing the mismatch of each feature to be quantized. The mismatch regularization loss is obtained by summing the mismatch over all quantized features:

$$\mathcal{L}_M = \sum_{i}^{\#\text{layers}} M(\boldsymbol{X}_i). \tag{3}$$

The mismatch regularization loss can be used in line with the original reconstruction loss typically used in the general quantization-aware training pipeline for SR networks:

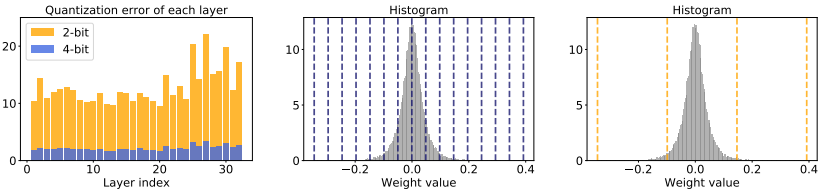
$$\mathcal{L}_R = \mathcal{L}_1(\mathcal{Q}(\boldsymbol{I}_{LR}), \boldsymbol{I}_{HR}), \tag{4}$$

where $\mathcal{L}_1$ loss indicates the l_1 distance between the reconstructed image using the quantized network $\mathcal{Q}$ and the ground-truth HR image $\boldsymbol{I}_{HR}$. Then, the optimization of the parameter θ^t is formulated as:

$$\theta^{t+1} = \theta^t - \beta^t \cdot (\nabla_\theta \mathcal{L}_R(\theta^t) + \nabla_\theta \mathcal{L}_M(\theta^t)), \tag{5}$$

where $\nabla_\theta \mathcal{L}_R(\theta^t)$ denotes the gradient from the original reconstruction loss and $\nabla_\theta \mathcal{L}_M(\theta^t)$ is the gradient from mismatch regularization loss, and β^t refers to the learning rate. Updating the network to minimize the mismatch regularization loss will reduce the overall error from feature quantization.

However, then a question arises: *does reducing the quantization error of each feature lead to improved reconstruction accuracy*? The answer is, according to our observation in Figure 2, not necessarily. During the training process, the mismatch regularization loss can collide with the original reconstruction loss. That is, for some parameters, the direction of the gradient from reconstruction loss and that of the mismatch regularization are opposing, referred to as gradient conflict [13]. As in Figure 2b, the cosine similarity of two gradients oscillates between positive and negative values during training, indicating that the directions of two gradients do not converge and the gradient



(a) Layer-wise QE using max policy **(b)** 4-bit quantization grid **(c)** 2-bit quantization grid

Figure 3: Layer-wise variation in error from weight quantization. (a) Quantization error (QE) varies across different layers when a fixed global policy (*i.e.*, max) is used to determine the quantization range, particularly for low bits. For some layers, using max does not effectively serve as a proper policy for quantization range selection. (b) Outliers often dominate the quantization range, leading to quantization grids being wasted on low-density areas. (c) At low bits, quantization grids fail to cover high-density regions adequately. Therefore, the quantization range should be adjusted for certain layers.

occasionally conflicts. Furthermore, as shown in Figure 2c, the proportion of parameters undergoing gradient conflict is not minor, implying that the regularization loss can severely hinder the reconstruction loss.

We aim to avoid the conflict between these two losses, in other words, to minimize the mismatch as long as it does not hinder the reconstruction loss. Thus, we dismiss the mismatch regularization term when it is not cooperative with reconstruction loss and make more use of it when it is cooperative. Specifically, we determine whether the two losses are cooperative by examining the cosine similarity of the gradients of each loss and then simply weigh the gradient of mismatch regularization by the gradient similarity. Our cooperative mismatch regularization can be formulated as follows:

$$\theta^{t+1} = \theta^t - \beta^t \cdot (\lambda_R \cdot \nabla_\theta \mathcal{L}_R(\theta^t) + \lambda_M \cdot \underline{sim(\nabla_\theta \mathcal{L}_R(\theta^t), \nabla_\theta \mathcal{L}_M(\theta^t))} \cdot \nabla_\theta \mathcal{L}_M(\theta^t)), \quad (6)$$

where the underlined term, gradient similarity, is defined as $sim(\boldsymbol{v}_a, \boldsymbol{v}_b) = \frac{cos(\boldsymbol{v}_a, \boldsymbol{v}_b)+1}{2}$ and $cos(\cdot, \cdot) \in [-1, 1]$ calculates the cosine similarity between two vectors. λ_R, λ_M are hyper-parameters to balance the two gradients. If the directions of two gradients are similar (*i.e.*, smaller than $90°$ and closer to $0°$), the gradient similarity is a large value, then the parameter is updated in the direction where mismatch regularization is also substantially considered. On the contrary, if the two gradients point in the other direction (*i.e.*, larger than $90°$ and closer to $180°$), the two losses restrain each other, and the gradient similarity is close to 0. In this case, we follow the gradient of reconstruction loss. This allows the network to reduce the quantization error cooperatively with the reconstruction error. Details on gradient similarity are in the supplementary material.

3.4 Distribution Mismatch in Weights

The weight distributions of SR networks have remained relatively unexplored in previous literature. This is because the weights in SR networks typically exhibit bell-shaped distributions, which are considered easier to quantize compared to the long-tailed, input-wise and channel-wise distinct activation distributions. Consequently, many

Algorithm 1 Quantization-aware training process of ODM

Input: Pre-trained 32-bit network $\mathcal{P}$.
Output: Quantized network $\mathcal{Q}$.

for $t = 1, \cdots$, # iters **do**
  for $i = 1, \cdots$, # layers **do**
    if $t = 1$ **then**
      Initialize activation quantization range $[l_a, u_a]$
      Initialize weight quantization range $[-u_w, u_w]$
    Given quantization range, obtain $q(\boldsymbol{X}_i; l_a, u_a)$ using Eq. (1)
    Given $\boldsymbol{X}_i$ and $q(\boldsymbol{X}_i; l_a, u_a)$, obtain mismatch using Eq. (2)
    Adjust weight quantization range $[-u_w, u_w]$ using Eq. (9)
    Given quantization range, obtain $q(\boldsymbol{W}_i; -u_w, u_w)$ using Eq. (1)
    Replace $\boldsymbol{X}_i, \boldsymbol{W}_i$ in $\mathcal{P}$ with $q(\boldsymbol{X}_i), q(\boldsymbol{W}_i)$
  Calculate mismatch regularization loss (Eq. (3)) and reconstruction loss (Eq. (4))
  Update parameters of $\mathcal{Q}$ with two losses cooperatively using Eq. (6)

studies [17, 33, 47] simply adopt max quantization for weights, setting the quantization range with the maximum value of the current weight distribution. However, we notice that this is suboptimal and may contribute to low performance when SR networks are quantized to ultra-low bits (*e.g.*, 2-bit). The issue arises because, in some layers, the outliers are far from the distribution mean, as visualized in Figures 3b and 3c. Thus, when the maximum value is used to determine the quantization range for such distributions, the quantization range is dominated by outliers, with quantization grids not allocated to high-density regions (*e.g.*, near 0). This leads to substantial quantization errors that can accumulate and degrade restoration performance. In the case of 4-bit quantization (Figure 3b), although grids still cover high-density regions to an extent, a number of grids are wasted on low-density areas. The problem intensifies with low-bit (2-bit) quantization (Figure 3c), where quantization errors become significantly larger. This observation underscores the need for careful selection of the quantization range for layer-wise weights, particularly in low-bit quantization scenarios.

3.5 Weight Clipping Correction

Also, we notice that the error from weight quantization varies across different layers, as shown in Figure 3a. For instance, employing the maximum value as the quantization range can be an effective policy for certain layers, yet this policy proves inadequate for others (*e.g.*, the layer shown in Figure 3c). Given the unique tendency of each layer, applying a uniform global policy for selecting the quantization range across all layers is suboptimal. Existing methods [17, 33, 47] utilize a fixed global policy throughout training to set the quantization range clipping parameter u_w as follows:

$$u_w^t = f(\boldsymbol{W}^t), \tag{7}$$

where the global policy $f(\cdot)$ is the same function for all layers (*e.g.*, $max(\cdot)$). A simple solution to accommodate layer-specific variations is to make the clipping parameter u_w

Table 2: Quantitative comparisons on EDSR of scale 4

Model	Bit	Set5		Set14		B100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	32	32.10	0.894	28.58	0.781	27.56	0.736	26.04	0.785
EDSR-PAMS	4	31.59	0.885	28.20	0.773	27.32	0.728	25.32	0.762
EDSR-DAQ	4	31.85	0.887	28.38	0.776	27.42	0.732	25.73	0.772
EDSR-DDTB	4	31.85	0.889	28.39	0.777	27.44	0.732	25.69	0.774
EDSR-ODM (Ours)	4	**32.00**	**0.891**	**28.47**	**0.779**	**27.51**	**0.735**	**25.80**	**0.778**
EDSR-PAMS	3	27.25	0.780	25.24	0.673	25.38	0.644	22.76	0.641
EDSR-DAQ	3	31.66	0.884	28.19	0.771	27.28	0.728	25.40	0.762
EDSR-DDTB	3	31.52	0.883	28.18	0.771	27.30	0.727	25.33	0.761
EDSR-ODM (Ours)	3	**31.85**	**0.888**	**28.38**	**0.776**	**27.43**	**0.732**	**25.59**	**0.771**
EDSR-PAMS	2	29.51	0.835	26.79	0.734	26.45	0.696	23.72	0.688
EDSR-DAQ †	2	31.01	0.871	27.89	0.762	27.09	0.719	24.88	0.740
EDSR-DDTB	2	30.97	0.876	27.87	0.764	27.09	0.719	24.82	0.742
EDSR-ODM (Ours)	2	**31.50**	**0.882**	**28.14**	**0.770**	**27.27**	**0.726**	**25.17**	**0.755**

† We note that the reported results of DAQ [18] are obtained using EDSR of 32 residual blocks. For a fair comparison, we reproduce EDSR-DAQ using EDSR of 16 residual blocks (*i.e.*, EDSR-baseline).

for each layer a learnable parameter [14]:

$$u_w^{t+1} = u_w^t - \beta^t \cdot \nabla_{u_w} \mathcal{L}_R(u_w^t), \tag{8}$$

where β^t denotes the learning rate. This process determines the clipping parameter u_w^t to quantize $\boldsymbol{W}^t$ based on the weight of the previous iteration, $\boldsymbol{W}^{t-1}$. However, since the weight is also updated at iteration step t ($\boldsymbol{W}^{t-1} \rightarrow \boldsymbol{W}^t$), a mismatch occurs between the current weight and the weight quantization range derived from the previous weight. To address this, we first obtain the quantization range by applying the global policy to the current weight, then adjust the range with a learnable parameter that accounts for layer-specific tendencies. Our clipping parameter is formulated as follows:

$$u_w^{t+1} = f(\boldsymbol{W}^{t+1}) \cdot (\gamma_w^t - \beta^t \cdot \nabla_{\gamma_w} \mathcal{L}_R(\gamma_w^t)), \tag{9}$$

where γ_w is the learnable parameter for each layer representing the layer-wise adjustment. Each γ_w is initially set to 1. To prevent outliers from dominating the initial quantization range, we set the global policy $f(\cdot)$ as the j-th percentile function. Our clipping correction scheme introduces only one additional parameter per layer; thus, the overall computational overhead is minimal. For further details, please refer to Section 4.3.

4 Experiments

The efficacy and adaptability of the proposed quantization framework, ODM, are assessed through its application across several SR networks. The experimental settings are described (Section 4.1), and quantitative (Section 4.2), qualitative (Section 4.4), and complexity (Section 4.3) evaluations are conducted on various SR networks. Ablation studies are conducted to examine each component of the framework (Section 4.5).

Table 3: Quantitative comparisons on RDN of scale 4

Model	Bit	Set5		Set14		B100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RDN	32	32.24	0.896	28.67	0.784	27.63	0.738	26.29	0.792
RDN-PAMS	4	30.44	0.862	27.54	0.753	26.87	0.710	24.52	0.726
RDN-DAQ	4	31.91	0.889	28.38	0.775	27.38	0.733	25.81	0.779
RDN-DDTB	4	31.97	0.891	28.49	0.780	27.49	0.735	25.90	0.783
RDN-ODM (Ours)	4	**32.06**	**0.892**	**28.49**	**0.780**	**27.52**	**0.736**	**25.88**	**0.783**
RDN-PAMS	3	29.54	0.838	26.82	0.734	26.47	0.696	23.83	0.692
RDN-DAQ	3	31.57	0.883	28.18	0.771	27.27	0.728	25.47	0.765
RDN-DDTB	3	31.49	0.883	28.17	0.772	27.30	0.728	25.35	0.764
RDN-ODM (Ours)	3	**31.79**	**0.887**	**28.33**	**0.776**	**27.42**	**0.732**	**25.51**	**0.770**
RDN-PAMS	2	29.73	0.843	26.96	0.739	26.57	0.700	23.87	0.696
RDN-DAQ	2	30.71	0.866	27.61	0.755	26.93	0.715	24.71	0.731
RDN-DDTB	2	30.57	0.867	27.56	0.757	26.91	0.714	24.50	0.728
RDN-ODM (Ours)	2	**31.37**	**0.880**	**28.08**	**0.770**	**27.24**	**0.727**	**25.09**	**0.755**

4.1 Implementation Details

Models and Training. The proposed framework is applied directly to existing representative SR networks that produce satisfactory SR results, but involve heavy computations: EDSR (baseline) [37] and RDN [55]. Furthermore, we apply our method to the Transformer-based SR model, SwinIR-S [36]. Following prior works on SR quantization [17, 18, 33, 40, 51, 57], weights and activations of the high-level feature extraction module are quantized, which is the most computationally demanding. Training and validation are conducted using the DIV2K [1] dataset. ODM trains the network for 60K iterations with a batch size of 8. The weights are updated with an initial learning rate of β^0=10^{-4}. For cooperative update, we update clipping parameters with $10 \cdot \beta^0$ initial learning rate. The learning rates are halved every 15K iteration. The hyperparameter for the percentile is set to $j = 99$, and to balance the gradient of the loss terms, we set λ_R=1 and λ_M=10^{-5}. Specially, we set λ_M=10^{-6} for RDN whose overall mismatch is large. Ablation studies on the hyperparameters are in the supplementary material. All our experiments are implemented using PyTorch and run on an RTX 2080Ti GPU.

Evaluation. We evaluate our framework on the standard benchmark (Set5 [4], Set14 [32], BSD100 [41], and Urban100 [21]) by measuring the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM [50]). To assess the computational complexity of our framework, we measure bitOPs and storage size. BitOPs refers to the number of operations weighted by the bit-widths of the two operands. Storage size is calculated as the number of stored parameters weighted by the precision of each parameter value.

4.2 Quantitative Results

To evaluate the effectiveness of our proposed scheme, we compare the results with existing SR quantization works using their official codes: PAMS [33], DAQ [18], and

Table 4: Quantitative comparisons on SwinIR of scale 4

Model	Bit	Set5		Set14		B100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR	32	32.44	0.898	28.77	0.786	27.69	0.741	26.47	0.798
SwinIR-PAMS	4	31.99	0.890	28.50	0.779	27.49	0.735	25.86	0.780
SwinIR-DAQ	4	31.82	0.887	28.34	0.775	27.37	0.730	25.68	0.772
SwinIR-DDTB	4	32.09	0.891	28.55	0.780	27.54	0.735	26.01	0.783
SwinIR-ODM (Ours)	4	**32.17**	**0.892**	**28.59**	**0.781**	**27.56**	**0.736**	**26.06**	**0.785**
SwinIR-PAMS	3	31.62	0.884	28.23	0.771	27.31	0.728	25.38	0.762
SwinIR-DAQ	3	31.50	0.882	27.99	0.770	27.12	0.727	25.29	0.761
SwinIR-DDTB	3	31.80	0.887	28.34	0.775	27.40	0.731	25.63	0.771
SwinIR-ODM (Ours)	3	**31.94**	**0.889**	**28.39**	**0.777**	**27.45**	**0.733**	**25.72**	**0.775**
SwinIR-PAMS	2	29.48	0.834	26.79	0.733	26.46	0.698	23.72	0.688
SwinIR-DAQ	2	29.10	0.824	26.55	0.725	26.30	0.691	23.51	0.678
SwinIR-DDTB	2	31.01	0.873	27.80	0.762	27.04	0.719	24.79	0.739
SwinIR-ODM (Ours)	2	**31.44**	**0.880**	**28.06**	**0.769**	**27.23**	**0.725**	**25.14**	**0.754**

DDTB [57]. For a fair comparison, we reproduce other methods using the same training iterations, 60K iterations. The supplementary materials provide additional experiments that further demonstrate the applicability of ODM, including results on 300K iterations, scale 2, and fully quantized settings.

EDSR. As shown in Table 2, ODM outperforms other methods in the 4, 3, and 2-bit settings, and notably, the improvement is significant for 2-bit, achieving a gain of more than 0.49 dB for Set5. We notice that 4-bit EDSR-ODM achieves closer accuracy to 32-bit EDSR, with a marginal difference of 0.1 dB for Set5. This indicates that ODM can effectively bridge the gap between the quantized network and the 32-bit network.

RDN. Similarly, Table 3 compares the results on RDN, whose computational complexity is more burdensome than EDSR. The results show that ODM consistently achieves superior performance on 4, 3, and 2-bit quantization. The gain over existing methods is especially large for the 2-bit setting, where it exceeds 0.66 dB for Set5.

SwinIR. Furthermore, we evaluate our framework on the Transformer-based architecture, SwinIR. The linear and convolutional layers of SwinIR are quantized. According to Table 4, ODM is also proven effective in quantizing SwinIR across all bit settings, where the improvement is most notable in the 2-bit setting (0.43 dB).

Comparison with QuantSR. We also compare our method with the concurrent work, QuantSR [44]. As the training code of QuantSR has not been released, we base our comparison on the reported performance. For a fair comparison with QuantSR's reported performance, we also train our model for 300K iterations on SRResNet [32] and SwinIR [36]. In Table 5, the results demonstrate that our method achieves better results than QuantSR; compared to QuantSR, ours shows a gain of 0.51 dB on 2-bit SRResNet and a gain of 0.14 dB on 2-bit SwinIR for Set5.

Table 5: Quantitative comparisons with QuantSR on SRResNet and SwinIR of scale 4. For a fair comparison, our model (ODM*) is trained for 300K iterations following QuantSR.

Model	Bit	Set5		Set14		B100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRResNet	32	32.07	0.893	28.50	0.780	27.52	0.735	25.86	0.779
SRResNet-QuantSR	2	31.30	0.882	28.08	0.769	27.23	0.725	25.13	0.754
SRResNet-ODM*(Ours)	2	**31.81**	**0.888**	**28.32**	**0.774**	**27.38**	**0.730**	**25.54**	**0.767**
SwinIR	32	32.44	0.898	28.77	0.786	27.69	0.741	26.47	0.798
SwinIR-QuantSR	2	31.53	0.885	28.16	0.772	27.28	0.727	25.26	0.761
SwinIR-ODM*(Ours)	2	**31.67**	**0.885**	**28.23**	**0.772**	**27.33**	**0.728**	**25.36**	**0.762**

Table 6: Computational complexity comparison with SR quantization methods

Model	Bit	Storage size	BitOPs	PSNR	SSIM
EDSR	32	1517.6K	527.1T	32.10	0.894
EDSR-PAMS	2	411.7K	215.1T	29.51	0.835
EDSR-DAQ	2	411.7K	215.6T	31.01	0.871
EDSR-DDTB	2	413.6K	215.1T	30.97	0.876
EDSR-ODM (Ours)	2	**411.8K**	**215.1T**	**31.50**	**0.882**

(a) EDSR

Model	Bit	Storage size	BitOPs	PSNR	SSIM
RDN	32	22271.1K	6032.9T	32.24	0.896
RDN-PAMS	2	1715.9K	236.6T	29.73	0.843
RDN-DAQ	2	1715.9K	287.7T	30.71	0.866
RDN-DDTB	2	1761.6K	236.6T	30.57	0.867
RDN-ODM (Ours)	2	**1716.1K**	**236.6T**	**31.37**	**0.880**

(b) RDN

4.3 Complexity Analysis

Along with the accuracy of SR, we also evaluate the computational complexity of our framework in Table 6. We measure the storage size for the model weights and the bitOPs required for generating a 1920×1080 image with $\times 4$ SR network. Overall, our framework, ODM, achieves higher restoration accuracy with similar or fewer computational resources. Specifically, our weight-clipping correction involves an additional storage size of 0.06K / 0.15K for EDSR / RDN. Since the correction process can be predetermined before test time, no extra bitOPs are required. Compared to existing methods, our method achieves $\times 31.7$ smaller storage size overhead than DDTB and 0.5T fewer bitOPs than DAQ on EDSR. For RDN, the gap is larger; our method's overhead is $\times 304.7$ smaller in storage size than DDTB and 51.1T fewer in bitOPs than DAQ. Moreover, we note that DAQ adopts a channel-wise dynamic quantization function and DDTB an asymmetric weight quantization function, which are not favorably supported by hardware. Despite incurring a minor additional storage size of 0.06K / 0.15K over PAMS, the accuracy improvement over PAMS is significant (+1.99 dB / +1.64 dB). The results prove that our method achieves significant accuracy gain with minimal or no computational overhead.

4.4 Qualitative Results

Figure 4 provides qualitative results and comparisons with the output images of quantized EDSR, RDN, and SwinIR. Our method, ODM, produces visually cleaner output

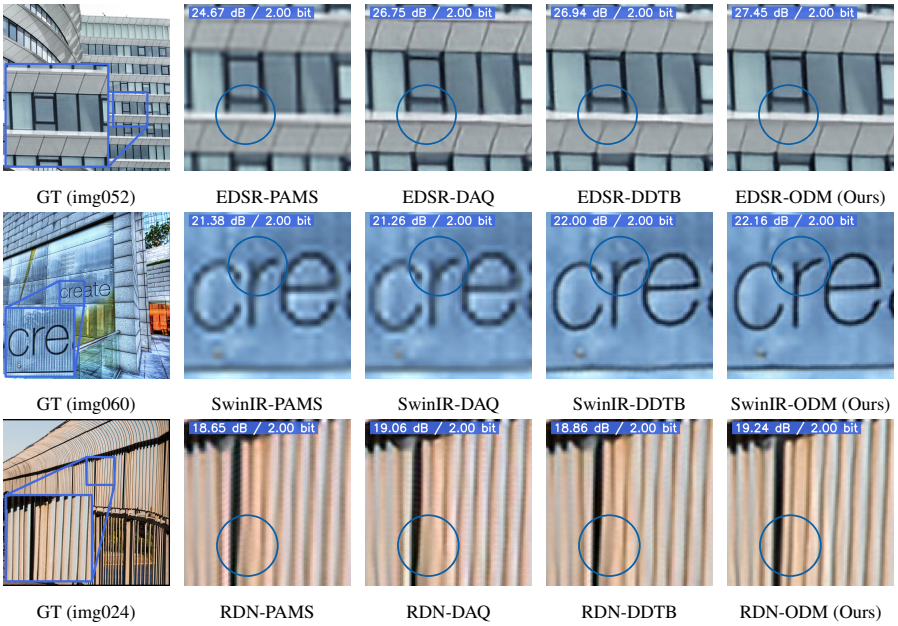


Figure 4: Qualitative results on Urban100 with EDSR, SwinIR, and RDN-based models

images compared to existing methods. In contrast, existing methods, especially PAMS, suffer from blurred lines or artifacts. These qualitative results stress the importance of alleviating the distribution mismatch problem in SR networks.

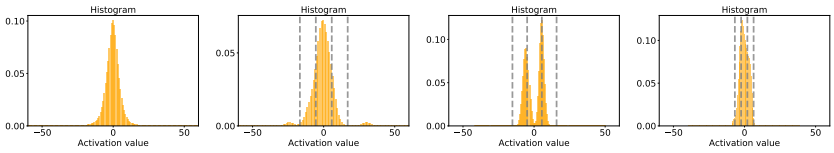
4.5 Ablation Study

In Table 7, we verify the importance of each attribute of our framework: cooperative mismatch regularization and weight clipping correction. According to the results, each attribute individually improves baseline accuracy. Weight clipping correction improves the baseline by +1.18 dB for Set5, indicating that considering both the layer-wise trend and the current weight distribution is important for weight quantization. Also, while directly integrating mismatch regularization with the reconstruction loss (Model (**d**)) rather degrades performance by -0.13 dB, our cooperative scheme (Model (**e**)) improves the SR accuracy by +0.51 dB. This shows that reducing the mismatch in both activations and weights is important for accurately quantizing SR networks.

Furthermore, we visualize the feature distributions in Figure 5 to validate the importance of our cooperative regularization term. After training only with the reconstruction loss, the outliers remain far from the quantization grid. When mismatch regularization loss is simply added to the original reconstruction loss, the activation distribution falls into a narrower range and resembles a multi-modal distribution near the quantization grids. Although this distribution is quantization-friendly, it significantly deviates from the original activation of the 32-bit network and removes originally dense values (near

Table 7: Ablation study on each attribute of our framework. *WCC* refers to weight clipping correction and *MR* refers to mismatch regularization. *Cooperative* denotes whether the mismatch regularization is cooperatively used with the reconstruction loss. Non-*cooperative MR* denotes that the two losses are simply used together.

Model	WCC	Cooperative	MR	Set5 (PSNR / SSIM)	Urban100 (PSNR / SSIM)
(a)	-	-	-	29.94 / 0.848	23.99 / 0.703
(b)	-	✓	✓	30.34 / 0.859	24.27 / 0.715
(c)	✓	-	-	31.12 / 0.876	24.91 / 0.746
(d)	✓	-	✓	30.99 / 0.871	24.79 / 0.735
(e)	✓	✓	✓	**31.50 / 0.882**	**25.17 / 0.755**



32-bit network $\mathcal{L}_R$ $\mathcal{L}_R + \mathcal{L}_M$ Cooperative MR

Figure 5: Distribution of activations before quantization. Using our cooperative mismatch regularization results in distributions more robust to low-bit quantization. 8th conv layer of EDSR-ODM (2-bit) on 'baby' (Set5) are visualized.

0). This can lead to an accuracy drop, as demonstrated in Table 7 (**d**). In contrast, our cooperative mismatch regularization results in quantization-friendly distribution while largely preserving the activations in the dense region of the original 32-bit network.

5 Conclusion

SR networks suffer accuracy loss from quantization due to the inherent mismatch in feature distributions. Instead of employing resource-demanding dynamic modules to handle distinct distributions during test time, we introduce a new quantization-aware training technique that alleviates this mismatch through distribution optimization. We utilize cooperative mismatch regularization to update the SR network to be quantization-friendly and accurate. Additionally, to address the mismatch in layer-wise weights, we propose a weight-clipping correction strategy. These straightforward solutions effectively reduce the distribution mismatch with minimal computational overhead.

Acknowledgment

This work was supported in part by the IITP grants [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), No.2021-0-02068, and No.2023-0-00156], the NRF grant [No.2021M3A9E4080782] funded by the Korean government (MSIT).

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPR Workshops (2017)
2. Ayazoglu, M.: Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. In: CVPR Workshops (2021)
3. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
4. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
5. Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: CVPR (2017)
6. Cai, Z., Vasconcelos, N.: Rethinking differentiable search for mixed-precision neural networks. In: CVPR (2020)
7. Chen, X., Wang, X., Zhou, J., Qiao, Y., Dong, C.: Activating more pixels in image super-resolution transformer. In: CVPR (2023)
8. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018)
9. Chu, X., Zhang, B., Ma, H., Xu, R., Li, Q.: Fast, accurate and lightweight super-resolution with neural architecture search. In: ICPR (2021)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (2014)
11. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE TPAMI **38**(2), 295–307 (2015)
12. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq: Hessian aware quantization of neural networks with mixed-precision. In: ICCV (2019)
13. Du, Y., Czarnecki, W.M., Jayakumar, S.M., Farajtabar, M., Pascanu, R., Lakshminarayanan, B.: Adapting auxiliary losses using gradient similarity. arXiv preprint arXiv:1812.02224 (2018)
14. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. In: ICLR (2020)
15. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. In: Low-Power Computer Vision, pp. 291–326. Chapman and Hall/CRC (2022)
16. Habi, H.V., Jennings, R.H., Netzer, A.: Hmq: Hardware friendly mixed precision quantization block for cnns. In: ECCV (2020)
17. Hong, C., Baik, S., Kim, H., Nah, S., Lee, K.M.: Cadyq: Content-aware dynamic quantization for image super-resolution. In: ECCV (2022)
18. Hong, C., Kim, H., Baik, S., Oh, J., Lee, K.M.: Daq: Channel-wise distribution-aware quantization for deep image super-resolution networks. In: WACV (2022)
19. Hong, C., Lee, K.M.: Adabm: On-the-fly adaptive bit mapping for image super-resolution. In: CVPR (2024)
20. Hou, L., Kwok, J.T.: Loss-aware weight quantization of deep networks. In: ICLR (2018)
21. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)
22. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACMMM (2019)
23. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: CVPR (2018)

24. Ignatov, A., Timofte, R., Denna, M., Younes, A.: Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In: CVPR Workshops (2021)
25. Jiang, X., Wang, N., Xin, J., Li, K., Yang, X., Gao, X.: Training binary neural network without batch normalization for image super-resolution. In: AAAI (2021)
26. Jiang, X., Wang, N., Xin, J., Xia, X., Yang, X., Gao, X.: Learning lightweight super-resolution networks with weight pruning. Neural Networks **144**, 21–32 (2021)
27. Jin, Q., Yang, L., Liao, Z.: Adabits: Neural network quantization with adaptive bit-widths. In: CVPR (2020)
28. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: CVPR (2021)
29. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: CVPR (2019)
30. Kim, H., Hong, S., Han, B., Myeong, H., Lee, K.M.: Fine-grained neural architecture search for image super-resolution. Journal of Visual Communication and Image Representation **89**, 103654 (2022)
31. Kim, J., Lee, J., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
32. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
33. Li, H., Yan, C., Lin, S., Zheng, X., Zhang, B., Yang, F., Ji, R.: Pams: Quantized super-resolution via parameterized max scale. In: ECCV (2020)
34. Li, Y., Gu, S., Zhang, K., Gool, L.V., Timofte, R.: Dhp: Differentiable meta pruning via hypernetworks. In: ECCV (2020)
35. Li, Y., Li, W., Danelljan, M., Zhang, K., Gu, S., Van Gool, L., Timofte, R.: The heterogeneity hypothesis: Finding layer-wise differentiated network architectures. In: CVPR (2021)
36. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: ICCV (2021)
37. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops (2017)
38. Liu, J., Wang, Q., Zhang, D., Shen, L.: Super-resolution model quantized in multi-precision. Electronics **10**(17), 2176 (2021)
39. Lou, Q., Guo, F., Liu, L., Kim, M., Jiang, L.: AutoQ: Automated kernel-wise neural network quantization. In: ICLR (2020)
40. Ma, Y., Xiong, H., Hu, Z., Ma, L.: Efficient super resolution using binarized neural network. In: CVPR Workshops (2019)
41. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
42. Oh, J., Kim, H., Nah, S., Hong, C., Choi, J., Lee, K.M.: Attentive fine-grained structured sparsity for image restoration. In: CVPR (2022)
43. Park, J., Son, S., Lee, K.M.: Content-aware local gan for photo-realistic super-resolution. In: ICCV (2023)
44. Qin, H., Zhang, Y., Ding, Y., Liu, X., Danelljan, M., Yu, F.: Quantsr: Accurate low-bit quantization for efficient image super-resolution. In: NeurIPS (2024)
45. Son, S., Kim, J., Lai, W.S., Yang, M.H., Lee, K.M.: Toward real-world super-resolution via adaptive downsampling models. IEEE TPAMI **44**(11), 8657–8670 (2021)
46. Song, D., Xu, C., Jia, X., Chen, Y., Xu, C., Wang, Y.: Efficient residual dense block search for image super-resolution. In: AAAI (2020)
47. Tian, S., Lu, M., Liu, J., Guo, Y., Chen, Y., Zhang, S.: Cabm: Content-aware bit mapping for single image super-resolution network with large input. In: CVPR (2023)

48. Wang, H., Chen, P., Zhuang, B., Shen, C.: Fully quantized image super-resolution networks. In: ACMMM (2021)
49. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: CVPR (2019)
50. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
51. Xin, J., Wang, N., Jiang, X., Li, J., Huang, H., Gao, X.: Binarized neural network for single image super resolution. In: ECCV (2020)
52. Yang, L., Jin, Q.: Fracbits: Mixed precision quantization via fractional bit-widths. In: AAAI (2021)
53. Zhang, Y., Chen, H., Chen, X., Deng, Y., Xu, C., Wang, Y.: Data-free knowledge distillation for image super-resolution. In: CVPR (2021)
54. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV (2018)
55. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR (2018)
56. Zhang, Y., Wang, H., Qin, C., Fu, Y.: Learning efficient image super-resolution networks via structure-regularized pruning. In: ICLR (2022)
57. Zhong, Y., Lin, M., Li, X., Li, K., Shen, Y., Chao, F., Wu, Y., Ji, R.: Dynamic dual trainable bounds for ultra-low precision super-resolution networks. In: ECCV (2022)
58. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)
59. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: CVPR (2018)