

CABM: Content-Aware Bit Mapping for Single Image Super-Resolution Network with Large Input

Anonymous CVPR submission

Paper ID 2446

Abstract

With the development of high-definition display devices, the practical scenario of Super-Resolution (SR) usually needs to super-resolve large input like 2K to higher resolution (4K/8K). To reduce the computational and memory cost, current methods first split the large input into local patches and then merge the SR patches into the output. These methods adaptively allocate a subnet for each patch. Quantization is a very important technique for network acceleration and has been used to design the subnets. Current methods train an MLP bit selector to determine the proper bit for each layer. However, they uniformly sample subnets for training, making simple subnets overfitted and complicated subnets underfitted. Therefore, the trained bit selector fails to determine the optimal bit. Apart from this, the introduced bit selector brings additional cost to each layer of the SR network. In this paper, we propose a novel method named Content-Aware Bit Mapping (CABM), which can remove the bit selector without any performance loss. CABM also learns a bit selector for each layer during training. After training, we analyze the relation between the edge information of an input patch and the bit of each layer. We observe that the edge information can be an effective metric for the selected bit. Therefore, we design a strategy to build an Edge-to-Bit lookup table that maps the edge score of a patch to the bit of each layer during inference. The bit configuration of SR network can be determined by the lookup tables of all layers. Our strategy can find better bit configuration, resulting in more efficient mixed precision networks. We conduct detailed experiments to demonstrate the generalization ability of our method. The code will be released.

1. Introduction

Single Image Super-Resolution (SISR) is an important computer vision task that reconstructs a High-Resolution (HR) image from a Low-Resolution (LR) image. With the

advent of Deep Neural Networks (DNNs), lots of DNN-based SISR methods have been proposed over the past few years [6, 17, 24, 27, 34]. While in real-world usages, the resolutions of display devices have already reached 4K or even 8K. Apart from normal 2D images, the resolutions of omnidirectional images might reach even 12K or 16K. Therefore, SR techniques with large input are becoming crucial and have gained increasing attention from the community [4, 12, 18, 28].

Since the memory and computational cost will grow quadratically with the input size, existing methods [4, 12, 18, 28] first split the large input into patches and then merge the SR patches to the output. They reduce the computational cost by allocating simple subnets to those flat regions while using heavy subnets for those detailed regions. Therefore, how to design the subnets is very important for these methods. [4, 18] empirically decide the optimal channels after lots of experiments to construct the subnets. [28] proposes to train a regressor to predict the incremental capacity of each layer. Thus they can adaptively construct the subnets by reducing the layers. Compared with pruning the channels or layers, quantization is another promising technique and can achieve more speedup. [12] trains an MLP bit selector to determine the proper bit for each layer given a patch. However, the introduced MLP of each layer brings additional computational and storage cost. Besides, we observe that [12] uniformly samples the subnets for training, making simple subnets (low average bit or flat patches) tend to overfit the inputs while complicated subnets (high average bit or detailed patches) tend to underfit the inputs. Therefore, uniform sampling fails to determine the optimal bit for each layer.

To solve the limitations of [12], we propose a novel method named Content-Aware Bit Mapping (CABM), which directly uses a lookup table to generate the bit of each layer during inference. However, building a lookup table is difficult since there are thousands of patches and corresponding select bits. We observe that edge information can be an effective metric for patch representation. Therefore, we analyze the relation between the edge information of a

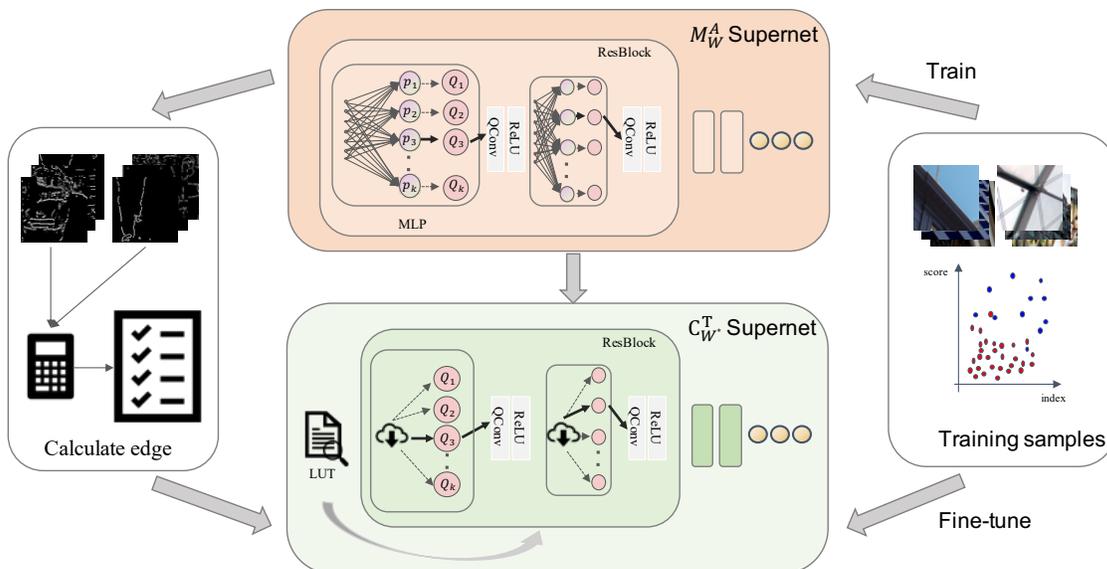


Figure 1. The pipeline of our CABM method. $p_i \in \{p_i\}_{i=1, \dots, K}$ is the probability of choosing i^{th} quantization module and each quantization module uses different bit-width to quantize the input activation. During training, our method learns an MLP bit selector to adaptively choose the bit-width for each convolution. While during inference, we use the proposed CABM to build an Edge-to-Bit lookup table to determine the bit-width with negligible additional cost.

patch and the bit of each layer. Inspired by the fact that a MLP selector learns the nonlinear mapping between a patch and the bit, instead of building the Edge-to-Bit lookup table based on linear mapping, we design a tactful calibration strategy to map the edge score of a patch to the bit of each layer. The bit configuration of SR network can be determined by the lookup tables of all layers. Our CABM can achieve the same performance compared with the MLP selectors while resulting in a lower average bit and negligible additional computational cost. Our contributions can be concluded as follows:

- We propose a novel method that maps edge information to bit configuration of SR networks, significantly reducing the memory and computational cost of bit selectors without performance loss.
- We present a tactful calibration strategy to build the Edge-to-Bit lookup tables, resulting in a lower average bit for SR networks.
- We conduct detailed experiments to demonstrate the generalization ability of our method based on various SR architectures and scaling factors.

2. Related work

DNN-based Image Super-Resolution With the rapid development of DNNs, lots of DNN-based SISR methods have been proposed over the past few years. SRCNN [6] is the pioneering work that applies DNNs to the SISR task. Their network consists of three modules including

feature extraction, non-linear mapping, and image reconstruction. The following works mostly follow the network design of SRCNN and improve the performance of SISR. For instance, VDSR [17] proposes to use a very deep neural network to predict the residual instead of the HR image. SRResNet [19] introduces the residual block proposed by ResNet [11] to SR network and improves the performance. EDSR [24] finds that the BN layer will impair the SR performance and removes it from the structure of SRResNet, further boosting the SR performance. RCAN [34] uses the attention mechanism and constructs deeper networks for SR. Real-ESRGAN [29] extends the powerful ESRGAN [30] to real-world blind SISR. They introduce a high-order degradation modeling process to simulate complex real-world degradations. USRNet [33] proposes an end-to-end trainable unfolding network that leverages both learning-based methods and model-based methods. Therefore, they can handle the SISR problem with different scaling factors, blur kernels, and noise levels under a unified framework. SwinIR [23] is a strong baseline model that introduces Swin Transformer [25] to image restoration. Their non-linear mapping module is composed of several residual Swin Transformer blocks. To reduce the computational cost, there are also many efficient SISR methods. For example, ESPCN [27] invents the pixel-shuffle layer to obtain the HR output given the LR input. LAPAR [22] presents a method based on a linearly-assembled adaptive regression network. Restormer [32] proposes an efficient Transformer model by making several key designs in the building blocks. All of those methods train one the SR model on large-scale image datasets such as DIV2K [1] and test on the given in-

put images. However, they are not designed for large input as the practical scenario of SR usually needs to super-resolve large input like 2K to higher resolution (4K/8K).

Single Image Super-Resolution with Large Input

With the development of display devices, the resolutions of monitors have reached 4K or even 8K. Recently, there are some methods of super-resolving large input to a higher resolution. Since the computational and memory cost grows quadratically with the spatial resolution, recent methods all split the large input into local patches and merge the SR patches to the output. ClassSR [18] is the seminal work that explores the problem of SISR with large input. They use a classification network to choose the restoration difficulty of each patch and allocate the optimal subnet for SR. However, ClassSR has two limitations. Firstly, it requires storing all the subnets. Secondly, the classification network brings additional computational costs. To solve the above limitations, APE [28] proposes to train a supernet for weight sharing. They use a regressor to predict the incremental capacity of each for the patch. ARM [4] also trains a supernet for weight sharing, however their subnets are constructed by reducing the channels instead of layers. CADyQ [12] uses network quantization to design the content-aware subnets. They train an MLP bit selector to determine the proper bit for each layer based on the content of an input patch. However, the MLP bit selector brings additional computational and storage costs. Besides, the selected bit is not optimal due to the uniform sampling.

Network Quantization Network quantization is a very effective technique to accelerate the speed. They map the 32-bit floating point values of feature and weight to lower bit values [3, 5, 9, 16, 20, 37]. Recent works also propose to allocate different bit-widths to different layers [7, 8, 31]. However, these works are mainly focused on high-level visual understanding tasks such as image classification. Different from high-level tasks, super-resolution is more sensitive to the network quantization. PAMS [21] proposes to train the learnable upper bounds for quantization due to the absence of BN layers. DAQ [13] uses different quantization parameters for each feature channel. DDTB [35] presents a novel activation quantizer to accommodate the asymmetry of the activations. CADyQ [12] designs mixed precision subnets for the input patch and uses MLP selectors to determine the bit configuration. However, the introduced MLP brings additional computational and storage cost to SR networks.

3. Method

3.1. Preliminary

In this section, we first introduce the background of network quantization since our method uses quantization to construct the subnets. For a given input activation x , the

quantized output x_q can be formulated as:

$$x_q = \lfloor \frac{\text{clamp}(x)}{s(n)} \rfloor s(n), \quad (1)$$

where $\text{clamp}(x) = \max(\min(x, \alpha), -\alpha)$ is the clamping function that uses a trainable upper bound to limit the range of input and $s(n) = \frac{\alpha}{2^n - 1 - 1}$ is the mapping function that symmetrically scales the input to low bit output. The quantization of weights is similar to activations. Different from activations, we use fixed bit-width to quantize weights following existing methods [12, 21]. For a quantized model, the complexity can be measured by the number of operations weighted by the bit-widths (BitOPs).

Our CABM method builds a supernet $\mathcal{C}_{W^*}^T$ for SISR tasks, where W^* denotes the quantized weight and T denotes the activation bit configurations obtained by the Edge-to-Bit lookup tables. All the subnets share the same weight of supernet and each subnet can be represented as $\mathcal{C}_{W^*}^{t_{s(p)}}$, where $t_{s(p)}$ represents a certain bit configuration given patch p and $s(\cdot)$ determines which subinterval the edge score of p belongs to.

3.2. Motivation

Recent works [12, 26] have introduced additional modules such as MLP to adaptively determine the network quantization. Based on the results of MLPs, we find that MLP selectors usually choose bit configurations with high BitOPs for those patches with high edge scores. However, we notice that MLP selectors sometimes choose low BitOPs for those patches with high edge scores as shown in Fig. 2. Based on this observation, we realize that the bit configuration determined by MLP selectors might not be optimal. This is because recent methods uniformly sample the subnets for training, which makes simple subnets tend to overfit the input while complicated subnets tend to underfit the input.

To further demonstrate the problem of uniform sampling, we conduct an experiment that samples the subnets according to the BitOPs. We define three levels of difficulties for subnets: easy, medium, and hard. The probability of sampling each type is calculated as follows.

$$l^m = \left(\frac{N_m \cdot \sum_{k=1}^{N_m} \text{BitOPs}^2(\mathcal{C}_{W^*}^{t_{s(k)}})}{\sum_{m=1}^3 (N_m \cdot \sum_{k=1}^{N_m} \text{BitOPs}^2(\mathcal{C}_{W^*}^{t_{s(k)}}))} \right), \quad (2)$$

where l^m denotes the probability of choosing m^{th} level, $\text{BitOPs}(\cdot)$ calculates the BitOPs of a subnet, N_m indicates the number of samples belongs to each level. As shown in Tab. 1, sampling based on BitOPs achieves better performance compared with uniform sampling. Therefore, we believe the bit configuration determined by MLP selectors is not optimal. In this paper, we propose a novel method named Content-Aware Bit Mapping (CABM) to choose the optimal bit configuration for SISR networks.

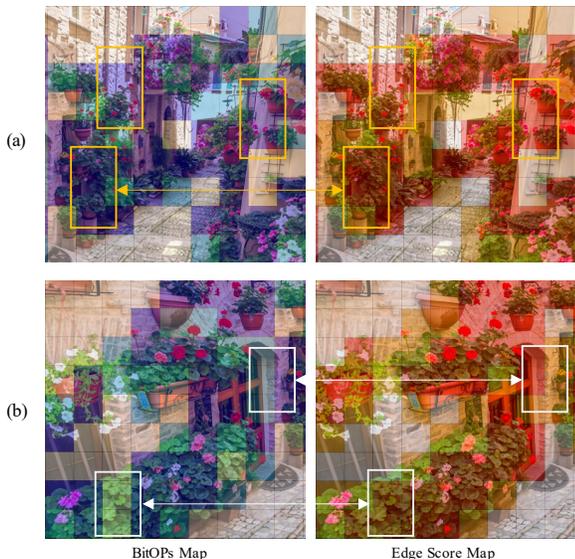


Figure 2. The motivation of our method. The darker color in the image means higher BitOPs / Edge Score. As can be seen in (a), patches with high edge scores often correspond to the bit configurations with high BitOPs. However, (b) shows that they are not always positively correlated, indicating the MLP selectors might fail to find optimal bit configurations.

Table 1. The comparison of uniform sampling and BitOPs sampling. Feature Average Bit (FAB), PSNR, SSIM are reported for EDSR on Urban100. U and B denote uniform sampling and BitOPs sampling respectively.

Model	FAB	PSNR	SSIM
EDSR-U	6.20	25.94	0.782
EDSR-B	6.20	26.01	0.783

3.3. CABM Supernet Training

Train a Supernet with MLP Selectors We first need to train a supernet that can adaptively generate all the bit configurations for model inference. Following existing methods [12, 26], we introduce a supernet \mathcal{M}_W^A that uses MLP to automatically decide bit configurations for various inputs. We choose the standard deviation of each layer’s feature and the edge score as the input of each MLP selector. Specifically, given a training set $\{L, H\}$, where $L = \{l_n\}_{n=1\dots K}$ are the low-resolution (LR) images, $H = \{h_n\}_{n=1\dots K}$ are the high-resolution (HR) images. The training process of \mathcal{M}_W^A is to balance the overall SR performance and the BitOPs of each subnet. Since this part is not our contribution, we simply summarize the process. For the details, we refer readers to the recent works [12, 26].

Build a Supernet with CABM As illustrated in the motivation, the uniform sampling makes simple subnets overfitted and complicated subnets underfitted. Therefore, the bit configuration of MLP selectors is not optimal for the

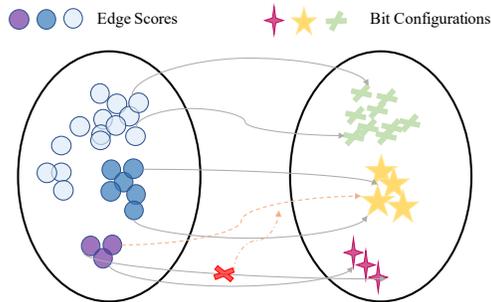


Figure 3. The illustration of Edge-to-Bit mappings.

given input patch. MLP selectors use two kinds of information to decide the bit for each layer. Among them, the standard deviation represents the feature importance of the current layer while the edge score of a patch is constant through layers. For different patches, the importance of layers might be different. However, for patches with the same edge score, the layer difference is almost negligible. This inspires us to build Edge-to-Bit lookup tables for determining the optimal bit configuration.

To be more specific, for all LR patches on the validation set of DIV2K, we compute the edge scores denoted as $E = \{e_i\}_{i=1\dots O}$. Then we use \mathcal{M}_W^A to generate the corresponding bit configurations for all the patches. Assume that the edge detection retains precision F , e.g. 0.01, we can then split the edge score interval $[0, \max(E)]$ to $R = \frac{10 \cdot \max(E) + F}{5 \cdot F}$ subintervals $S = \{s_r\}_{r=1\dots R}$ so that r^{th} subinterval can be defined as:

$$s_r = \left[\frac{F \cdot (r - 1)}{2}, \frac{F \cdot (5r - 1)}{10} \right]. \quad (3)$$

Given a patch \hat{p} with its edge score $D(\hat{p})$, we can determine the index of subinterval for $D(\hat{p})$ according to the R subintervals. We denote the process that determines the subinterval of \hat{p} as $s(\cdot)$. For those subintervals without corresponding bit configurations, we choose the bit configurations from the nearest subintervals for them. Therefore, we can build a simple mapping between edge scores and bit configurations using \mathcal{M}_W^A .

$$s(D(\hat{p})) \in S \rightarrow \bar{T} = \{\bar{t}_i\}_{i=1\dots O} \quad (4)$$

However, one subinterval might correspond to thousands of bit configurations. We design a simple yet effective strategy to determine one optimal bit configuration for each subinterval. We observe that high bit widths retain more information of features and improve the accuracy of quantized models. However, we need to minimize the computational cost using low bit widths. Therefore, we first sample the bit configuration with minimum BitOPs for each subinterval. However, for a small number of patches within a certain precision range, the bit configurations determined by MLPs may fall into the local minimum. Our solution to

Table 2. Quantitative comparison of full precision models, PAMS, CADyQ and our method on Urban100, Test2K and Test4K. Computational complexity is measured by BitOPs of the backbone network for generating a 720p/2K/4K image accordingly. Feature Average Bit (FAB) and PSNR (dB) /SSIM results are also reported for each model. The scaling factor is x4.

Model	BitOPs	FAB	Urban100	BitOPs	FAB	Test2K	BitOPs	FAB	Test4K
CARN	90.88G	32.00	25.91/0.779	363.53G	32.00	27.54/0.772	1.26T	32.00	28.92/0.818
CARN-PAMS	5.68G	8.00	25.80/0.776	22.72G	8.00	27.49/0.770	78.75G	8.00	28.86/0.817
CARN-CADyQ	3.13G	4.99	25.88/0.778	10.55G	4.40	27.52/0.770	38.11G	4.36	28.87/0.817
CARN-CABM (Ours)	1.57G	4.21	25.90/0.779	6.11G	4.15	27.51/0.771	20.89G	4.12	28.88/0.817
IDN	81.88G	32.00	25.46/0.764	327.52G	32.00	27.40/0.766	1.13T	32.00	28.73/0.813
IDN-PAMS	5.12G	8.00	25.56/0.768	20.47G	8.00	27.43/0.767	70.62G	8.00	28.77/0.814
IDN-CADyQ	3.33G	5.14	25.58/0.769	11.33G	4.49	27.42/0.766	41.47G	4.46	28.75/0.813
IDN-CABM (Ours)	1.46G	4.28	25.57/0.768	5.77G	4.25	27.42/0.766	19.75G	4.23	28.74/0.813
SRResNet	146.44G	32.00	25.74/0.770	585.75G	32.00	27.50/0.770	2.03T	32.00	28.87/0.817
SRResNet-PAMS	9.15G	8.00	25.85/0.778	36.61G	8.00	27.52/0.771	126.88G	8.00	28.90/0.818
SRResNet-CADyQ	6.90G	6.07	25.87/0.778	24.97G	5.68	27.52/0.770	87.58G	5.57	28.88/0.817
SRResNet-CABM (Ours)	4.08G	5.34	25.86/0.778	15.29G	5.17	27.52/0.771	50.96G	5.07	28.91/0.818
EDSR	114.23G	32.00	26.03/0.784	456.92G	32.00	27.59/0.773	1.58T	32.00	28.99/0.820
EDSR-PAMS	7.14G	8.00	26.01/0.784	28.56G	8.00	27.59/0.773	98.75G	8.00	28.99/0.820
EDSR-CADyQ	6.64G	6.70	25.90/0.781	23.18G	6.11	27.54/0.771	82.37G	6.06	28.91/0.818
EDSR-CABM (Ours)	3.75G	5.80	25.95/0.782	14.24G	5.65	27.57/0.772	47.70G	5.56	28.96/0.819

Table 3. Quantitative comparison of full precision models, PAMS, CADyQ and our method on Urban100, Test2K and Test4K. Computational complexity is measured by BitOPs of the backbone network for generating a 720p/2K/4K image accordingly. Feature Average Bit (FAB) and PSNR (dB) /SSIM results are also reported for each model. The scaling factor is x2.

Model	BitOPs	FAB	Urban100	BitOPs	FAB	Test2K	BitOPs	FAB	Test4K
CARN	222.83G	32.00	31.93/0.926	891.33G	32.00	32.77/0.928	3.08T	32.00	34.34/0.943
CARN-PAMS	13.93G	8.00	31.97/0.927	55.71G	8.00	32.75/0.928	192.50G	8.00	34.36/0.943
CARN-CADyQ	5.25G	4.46	31.82/0.925	19.18G	4.22	32.69/0.927	67.53G	4.19	34.29/0.942
CARN-CABM (Ours)	3.61G	4.09	31.96/0.927	14.35G	4.06	32.75/0.928	49.58G	4.06	34.35/0.923
IDN	174.10G	32.00	31.22/0.919	696.41G	32.00	32.39/0.923	2.41T	32.00	34.00/0.940
IDN-PAMS	10.88G	8.00	31.28/0.920	43.53G	8.00	32.46/0.925	151.25G	8.00	34.04/0.941
IDN-CADyQ	5.96G	5.28	31.34/0.921	18.35G	4.45	32.48/0.925	66.11G	4.44	34.07/0.941
IDN-CABM (Ours)	3.01G	4.21	31.40/0.921	11.94G	4.19	32.50/0.925	41.49G	4.19	34.10/0.941
SRResNet	406.63G	32.00	31.51/0.922	1.63T	32.00	32.56/0.925	5.62T	32.00	34.15/0.941
SRResNet-PAMS	25.41G	8.00	31.55/0.923	101.88G	8.00	32.53/0.925	351.25G	8.00	34.17/0.942
SRResNet-CADyQ	17.04G	6.23	31.48/0.922	64.60G	6.04	32.50/0.925	222.34G	5.98	34.12/0.941
SRResNet-CABM (Ours)	11.84G	5.46	31.54/0.923	45.22G	5.33	32.55/0.925	150.12G	5.23	34.16/0.942
EDSR	316.25G	32.00	31.97/0.927	1.27T	32.00	32.75/0.928	4.37T	32.00	34.37/0.943
EDSR-PAMS	19.77G	8.00	32.06/0.928	79.38G	8.00	32.79/0.928	273.13G	8.00	34.42/0.944
EDSR-CADyQ	12.86G	6.03	31.84/0.925	51.01G	5.88	32.68/0.927	169.15G	5.79	34.28/0.942
EDSR-CABM (Ours)	9.65G	5.59	31.92/0.927	36.03G	5.39	32.74/0.927	120.33G	5.31	34.33/0.943

this problem is expanding the range of subintervals. Thus each expanded subinterval can contain more bit configurations. The adjusted subintervals can be formulated as:

$$s_r = \begin{cases} \left[\frac{F \cdot (r-1)}{2}, \frac{F \cdot (5r-1)}{10} \right], & r \in [0, \beta], \\ \left[\frac{F \cdot (r-1)}{2} - \Delta e, \frac{F \cdot (5r-1)}{10} + \Delta e \right], & r \in (\beta, R]. \end{cases} \quad (5)$$

where Δe and β are hyper-parameters. In this way, we can find one optimal bit configuration by choosing the bit configuration with minimum BitOPs for each expanded subinterval. We build the optimal one-to-one Edge-to-Bit lookup

tables $T = \{t_r\}_{r=1 \dots R}$ to reduce the BitOPs of MLP selectors. The process of CABM is illustrated by Fig. 3. After building the Edge-to-Bit lookup tables with CABM to construct the supernet $C_{W^*}^T$, we conduct a simple fine-tuning for the CABM supernet:

$$\min \sum_{n=1}^B \sum_{p=1}^{P_n} \|C_{W^*}^{t_s(p)}(l_p^n) - h_p^n\|_1, \quad (6)$$

where $\|\cdot\|_1$ denotes the commonly-used ℓ_1 -norm in the SISR task, $l_p \in l_n = \{l_p\}_{p=1 \dots P} \in \{l_n\}_{n=1 \dots B}$ and

$h_p \in h_n = \{h_p\}_{p=1\dots P} \in \{h_n\}_{n=1\dots B}$ are respectively the LR image patch and HR image patch in the current training iteration, B is the batch size and P_n is the patch number for each image. W^* is initialized from the weight W of \mathcal{M}_W^A .

3.4. CABM Supernet Inference

During inference, we first split the whole large input into local patches of a given size, and then use the Laplacian edge detection operator to calculate the edge scores. Based on the edge scores, we can quickly obtain the corresponding subnet for each patch from CABM supernet. After super-resolving all the patches, we merge the SR patches to the output. Compared with MLP selectors, CABM can achieve lower BitOPs with negligible additional computational cost. The whole pipeline of CABM is shown in Fig. 1.

4. Experiment

4.1. Experimental Settings

To verify the generalization and effectiveness of our method, we conduct detailed experiments on four representative models. For plain models, we choose two widely used SISR models, *i.e.* EDSR [24] and SRResNet [19]. Since the quantization for activations mainly affects the flow of information between blocks, it is not enough to apply our CABM to plain models. Therefore, we apply CABM to IDN [15] and CARN [2] that have a hierarchical feature extraction mechanism. Besides, we notice that the leaning-based quantization method we choose performs better when bit-width is greater than 4, and Tensor Cores mainly support 4/8-bit, so it is reasonable to select 4/6/8-bit as our candidate bits. As for quantization details of activation and weight, we follow the settings of previous works [12, 36].

Implementation Details All models are trained on DIV2K datasets [1] which contains 800 images for training, 100 images for validation, and 100 images for testing. The lookup tables are built based on the validation set using the proposed CABM method and the precision of edge scores is $F = 0.01$. For CABM fine-tuning, we use the pre-trained weight from supernet \mathcal{M}_W^A . For testing, we use the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) as the metrics to evaluate the SR performance of all methods on three test datasets: benchmark [14], Test2K and Test4K [18]. In Test2k and Test 4k, images are generated following previous work from DIV8K datasets (index 1201-1400) [10]. Without special mention, all test input images are split to 96×96 LR patches with scaling factor x4. For EDSR and SRResNet, Δe and β are respectively set to 10 and 9000. For CARN and IDN, they are set to 10 and 6000.

Table 4. Ablation study of CABM Fine-tuning. FAB, PSNR (dB), and SSIM results are reported for each model on Set14 and Urban100 datasets.

Datasets	Set14		Urban100	
	FAB	PSNR/SSIM	FAB	PSNR/SSIM
EDSR- \mathcal{M}_W^A	6.37	28.53/0.780	6.70	25.90/0.781
EDSR- \mathcal{C}_W^T	5.80	28.47/0.777	5.80	25.77/0.774
IDN- \mathcal{M}_W^A	4.72	28.34/0.774	5.14	25.58/0.769
IDN- \mathcal{C}_W^T	4.18	27.08/0.750	4.28	24.32/0.732

4.2. Quantitative and Qualitative Results

To fully prove the effectiveness and generalization of our proposed CABM method, we compare our results with full precision models, PAMS [21] and CADyQ [12]. PAMS is a SISR quantization method, which uses fixed bit-width for network. CADyQ uses MLPs to adaptively adjust the bit-widths according to the input patch similar to our method. All the models are trained by ourselves using the official codebases and instructions to avoid unfair comparison. As shown in Tab. 2, our CABM method reduces the computational overhead and achieves the accuracy as full precision models on scaling factor x4. For SRResNet, CABM outperforms the full precision model by 0.11 dB (PSNR) and 0.008 (SSIM) with only 2.8% BitOPs. As for IDN, it is obvious that MLP bit selectors fail to learn the optimal bit configurations for the quantized models. Besides, the introduced MLPs brings additional computational cost to SR networks. Compared with them, our CABM is much more efficient when the BitOPs of quantized models are low enough since the lookup tables bring negligible additional computational cost. We also show the comparison on scaling factor x2 in Tab. 3. Although our method fails to achieve competitive results compared with full precision models, our method obtains similar results as PAMS and CADyQ while our FAB is much lower. Fig. 4 shows the qualitative results. To summarize, our CABM method achieves almost lossless model performance compared with PAMS and CADyQ while significantly reduces the computational overhead.

4.3. Ablation Study

CABM Fine-tuning After we get the MLP supernet \mathcal{M}_W^A and generate our CABM look-up tables, one straightforward choice is using the original weight W to build a CABM supernet \mathcal{C}_W^T . Instead, the proposed method conduct the fine-tuning after CABM. To verify the effectiveness of CABM fine-tuning, we show the results in Tab. 4. The experimental settings are consistent with our main results. Without fine-tuning, \mathcal{C}_W^T fails to achieve reasonable results after removing MLPs. Especially for IDN [15], a network with dense information, the PSNR value is 1.26 dB lower. This further provides experimental support to our motiva-

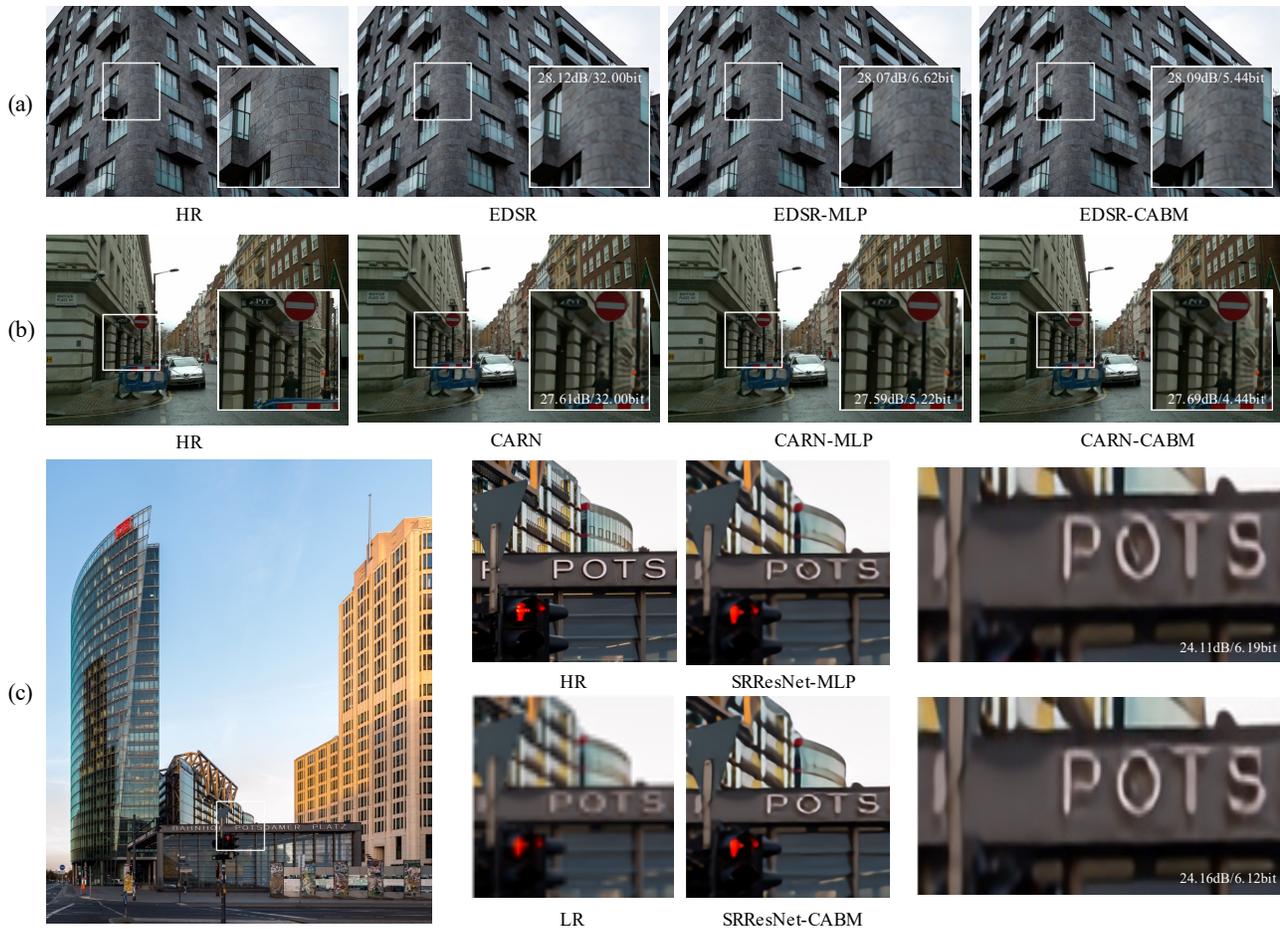


Figure 4. (a) (b) Qualitative comparison on two images from Urban100 with EDSR/CARN. (c) (d) Qualitative comparison on one image from Test4K with SRResNet. As can be seen, our method can achieve similar performance while reduce the FAB.

Table 5. The comparison of different calibration settings on EDSR with CABM. FAB, PSNR (dB), and SSIM results are reported for each setting on Urban100 datasets.

Δe	FAB	Uniform		BitOPs	
		PSNR	SSIM	PSNR	SSIM
10	5.80	25.91	0.781	25.95	0.782
20	5.71	25.91	0.781	25.93	0.781
30	5.65	25.90	0.781	25.91	0.780
40	5.60	25.90	0.780	25.89	0.779
80	5.54	25.90	0.780	25.87	0.778

tion in Sec. 3.2 that bit configurations and edge scores are not always positively correlated. Therefore, in order to adapt each layer, fine-tuning is necessary after CABM.

Different Calibration Settings To evaluate our proposed CABM method in Sec. 3.3, we conduct more experiments to evaluate the impact of different settings. Specifically, we choose EDSR as our backbone and the β is set

Table 6. The comparison of different strategies for bit selecting on EDSR with CABM. FAB, PSNR (dB), and SSIM results are reported for each model on Set14 and Urban100 datasets.

Datasets	Set14		Urban100	
	FAB	PSNR/SSIM	FAB	PSNR/SSIM
EDSR-baseline	32.00	28.58/0.781	32.00	26.03/0.784
$C_{W^*}^{T1}$	5.92	28.56/0.780	5.90	25.95/0.782
$C_{W^*}^{T2}$	6.69	28.58/0.781	6.64	26.04/0.784
$C_{W^*}^{T3}$	6.21	28.57/0.781	6.20	26.01/0.783

to 9000 with edge precision $F = 0.01$. Tab. 5 reports the quantitative results of different expanding settings. As we have mentioned, the results obtained by MLP selectors are often not optimal due to the uniform subnet sampling. As we increase the expanding range Δe , the possibilities of finding better solutions and worse solutions increase at the same time. By analyzing the results of expanding ranges, we can empirically set the optimal Δe for the proposed CABM method. In Fig. 5, we have compared the visual

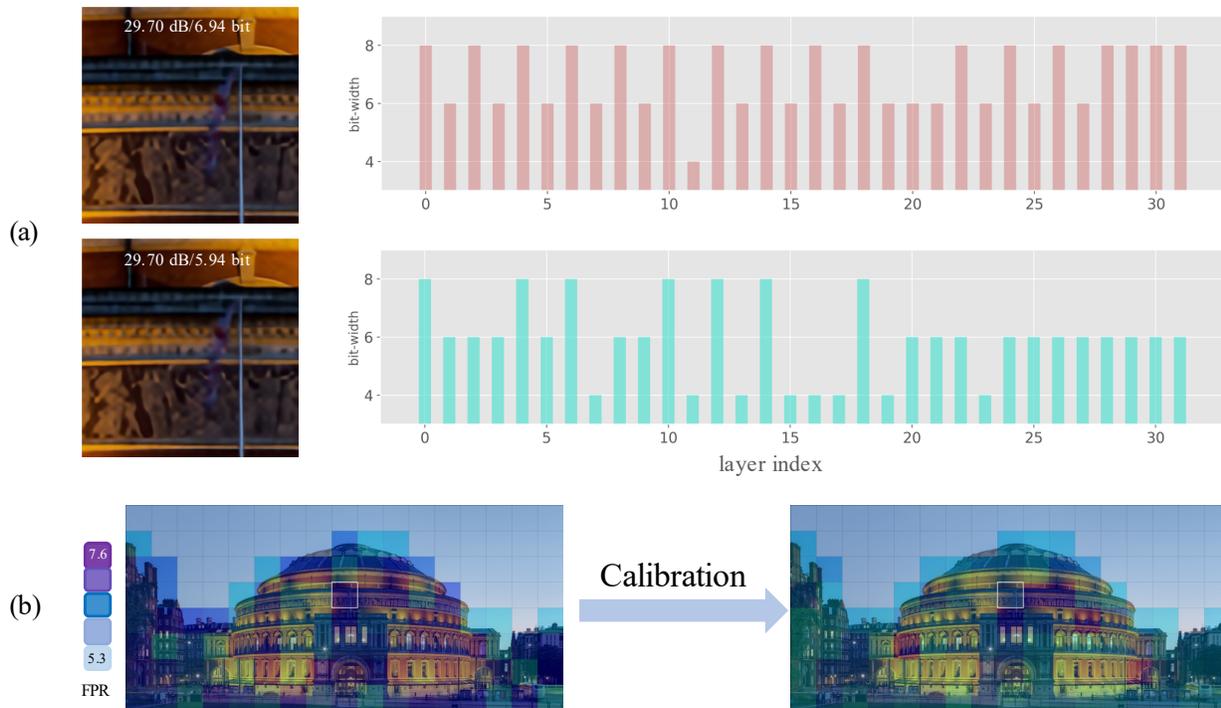


Figure 5. Quantitative comparison on an image in Test4K with backbone network EDSR. (a) Comparison between patches before and after calibration, and the corresponding bit configuration of different layers. (b) FAB heat maps before and after calibration.

results of the mappings before and after our strategy, and Δe is set to 80 here. As can be seen, both mappings can achieve similar performance, while our strategy can calibrate the mapping and further reduce 1 FAB. This demonstrate that our calibration method can improve the original one-to-many mapping and reduce the computational cost.

Bit Selection Strategies Our method expands the range of subintervals and select one bit configuration with minimum BitOPs for each subinterval. We denote our minimum sampling strategy as S1. The other two alternatives are sampling one bit configuration with maximum BitOPs (S2) and randomly sampling one bit configuration (S3). To evaluate these bit selection strategies, we analyze and compare S1, S2 and S3. We fine-tune different supernets $\mathcal{C}_{W^*}^T$ and denote them as $\mathcal{C}_{W^*}^{T1}$, $\mathcal{C}_{W^*}^{T2}$ and $\mathcal{C}_{W^*}^{T3}$, where $T1$, $T2$ and $T3$ are different look-up tables generated by S1, S2, and S3 respectively. It can be observed from Tab. 6 that $\mathcal{C}_{W^*}^{T2}$ achieves the best performance. However, from the perspective of balancing performance and computation, $\mathcal{C}_{W^*}^{T2}$ may not be an appropriate choice since the PSNR on Set14 is only 0.02 dB higher than $\mathcal{C}_{W^*}^{T1}$ while the FAB is 0.77 higher. Therefore, we choose the S1 strategy for the proposed CABM.

5. Limitation

Although CABM can further reduce the computational overhead of existing SISR methods, our method still has

some limitations. For example, our network uses mixed precision quantization, which requires specific hardware support to achieve practical speedup. Besides, applying CABM to Video Super-Resolution (VSR) is not a trivial task since VSR needs to consider the temporal information and has more modules compared with SISR. Mixed precision quantization of VSR networks is still unexplored to the best of our knowledge. These problems will be our future works.

6. Conclusion

To summarize, we propose a novel Content-Aware Bit Mapping (CABM) method for SISR with large input. Existing methods learn the MLP selectors to determine the bit configuration for a given patch. However, they uniformly sample the subnets and fail to obtain optimal bit configuration. Instead of using the MLPs, CABM builds the Edge-to-Bit lookup tables to determine the bit configuration. In order to further reduce the computational cost, we present a novel calibration strategy to find a better mapping between edge score and bit configuration. Our method can achieve similar performance as existing methods with negligible additional computational and storage cost. We hope our method can inspire future works on SR techniques with large input.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2, 6
- [2] Namhyuk Ahn, Byungkon Kang, and Kyung ah Sohn. Fast, accurate, and, lightweight super-resolution with cascading residual network. *ArXiv*, abs/1803.08664, 2018. 6
- [3] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5918–5926, 2017. 3
- [4] Bohong Chen, Mingbao Lin, Kekai Sheng, Mengdan Zhang, Peixian Chen, Ke Li, Liujuan Cao, and Rongrong Ji. Arm: Any-time super-resolution method. *arXiv preprint arXiv:2203.10812*, 2022. 1, 3
- [5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 3
- [6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 1, 2
- [7] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. *Advances in neural information processing systems*, 33:18518–18529, 2020. 3
- [8] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019. 3
- [9] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 3
- [10] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3512–3516, 2019. 6
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2
- [12] Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq: Content-aware dynamic quantization for image super-resolution. *arXiv preprint arXiv:2207.10345*, 2022. 1, 3, 4, 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2675–2684, 2022. 3
- [14] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015. 6
- [15] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 723–731, 2018. 6
- [16] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4359, 2019. 3
- [17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 1, 2
- [18] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12016–12025, 2021. 1, 3, 6
- [19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2, 6
- [20] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6448–6457, 2021. 3
- [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*, 2020. 3, 6
- [22] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *Advances in Neural Information Processing Systems*, 33, 2020. 2
- [23] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 2
- [24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 6
- [25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: 918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

972	Hierarchical vision transformer using shifted windows. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pages 10012–10022, 2021. 2	1026
973		1027
974		1028
975	[26] Zhenhua Liu, Yunhe Wang, Kai Han, Siwei Ma, and Wen Gao. Instance-aware dynamic neural network quantization. <i>2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)</i> , pages 12424–12433, 2022. 3, 4	1029
976		1030
977		1031
978		1032
979	[27] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In <i>Proceedings of the IEEE conference on computer vision and pattern recognition</i> , pages 1874–1883, 2016. 1, 2	1033
980		1034
981		1035
982		1036
983		1037
984	[28] Shizun Wang, Ming Lu, Kaixin Chen, Xiaoqi Li, Jiaming Liu, and Yandong Guo. Adaptive patch exiting for scalable single image super-resolution. <i>arXiv preprint arXiv:2203.11589</i> , 2022. 1, 3	1038
985		1039
986		1040
987		1041
988	[29] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pages 1905–1914, 2021. 2	1042
989		1043
990		1044
991		1045
992		1046
993	[30] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In <i>Proceedings of the European conference on computer vision (ECCV) workshops</i> , pages 0–0, 2018. 2	1047
994		1048
995		1049
996		1050
997	[31] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In <i>International Conference on Machine Learning</i> , pages 11875–11886. PMLR, 2021. 3	1051
998		1052
999		1053
1000		1054
1001		1055
1002	[32] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pages 5728–5739, 2022. 2	1056
1003		1057
1004		1058
1005		1059
1006		1060
1007		1061
1008	[33] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In <i>Proceedings of the IEEE/CVF conference on computer vision and pattern recognition</i> , pages 3217–3226, 2020. 2	1062
1009		1063
1010		1064
1011	[34] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In <i>Proceedings of the European conference on computer vision (ECCV)</i> , pages 286–301, 2018. 1, 2	1065
1012		1066
1013		1067
1014		1068
1015		1069
1016	[35] 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. <i>arXiv preprint arXiv:2203.03844</i> , 2022. 3	1070
1017		1071
1018		1072
1019		1073
1020		1074
1021	[36] 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. <i>ArXiv</i> , abs/2203.03844, 2022. 6	1075
1022		1076
1023		1077
1024	[37] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards	1078
1025		1079