Look-Up Table Compression for Efficient Image Restoration

Yinglong Li Jiacheng Li Zhiwei Xiong*
University of Science and Technology of China
{yllee, jclee}@mail.ustc.edu.cn zwxiong@ustc.edu.cn

Abstract

Look-Up Table (LUT) has recently gained increasing attention for restoring High-Quality (HQ) images from Low-Quality (LQ) observations, thanks to its high computational efficiency achieved through a "space for time" strategy of caching learned LQ-HQ pairs. However, incorporating multiple LUTs for improved performance comes at the cost of a rapidly growing storage size, which is ultimately restricted by the allocatable on-device cache size. In this work, we propose a novel LUT compression framework to achieve a better trade-off between storage size and performance for LUT-based image restoration models. Based on the observation that most cached LQ image patches are distributed along the diagonal of a LUT, we devise a Diagonal-First Compression (DFC) framework, where diagonal LQ-HQ pairs are preserved and carefully re-indexed to maintain the representation capacity, while non-diagonal pairs are aggressively subsampled to save storage. Extensive experiments on representative image restoration tasks demonstrate that our DFC framework significantly reduces the storage size of LUT-based models (including our new design) while maintaining their performance. For instance, DFC saves up to 90% of storage at a negligible performance drop for $\times 4$ *super-resolution. The source code is available on GitHub: https://github.com/leenas233/DFC.*

1. Introduction

Image restoration, such as super-resolution, denoising, deblocking, and deblurring, aims at reconstructing high-quality (HQ) images with rich high-frequency details from low-quality (LQ) degraded observations. In recent years, with the rapid development of deep learning, methods based on deep neural networks (DNN) [2, 13, 19, 22, 29, 50, 51] have made impressive progress in image restoration. Nevertheless, these methods often require high computational costs or dedicated computing devices, *e.g.*, GPUs and TPUs. Such an excessive computational requirement limits the usage of DNN-based methods on edge devices with limited resources, such as smartphones and televisions.

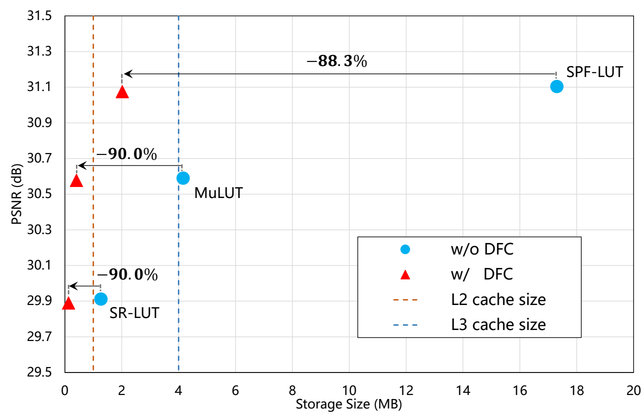


Figure 1. Performance-storage trade-off for $\times 4$ super-resolution on the Set5 dataset [3]. Our proposed DFC effectively compresses the storage size of the LUT-based models with a high compression ratio while maintaining performance. The orange dotted line and blue dotted line indicate the L2 cache and L3 cache sizes of a Qualcomm Snapdragon 888 Plus chip respectively.

Recently, Look-Up Table (LUT) has gained increasing attention for image restoration thanks to its high computational efficiency achieved through a "space for time" strategy of caching learned LQ-HQ pairs. Based on this strategy, Jo and Kim [21] propose SR-LUT, where LQ-HQ pairs are cached into a LUT by traversing all possible low-resolution (LR) inputs and pre-computing the corresponding high-resolution (HR) outputs of a trained DNN for super-resolution. Then, the HR prediction is retrieved from the LUT by querying the LR input at the inference phase. This way, the huge computational overhead of DNN is replaced by the storage size of the saved LUT, and the inference time is effectively reduced. Most recently, MuLUT [26] and SPLUT [32] propose to adopt multiple LUTs in a cascaded structure, showing a scaling law of obtaining better performance with more LUTs. In this paper, we design a new structure based on multiple LUTs, named SPF-LUT, extending the scaling law. Nevertheless, as more LUTs are used, the required storage grows rapidly, which is ultimately restricted by the allocatable on-device cache size. As shown in Fig. 1, advanced methods based on multiple

*Corresponding author.

LUTs can achieve superior performance, but they demand a rapid growth in storage size, which may exceed the limit of the mainstream cache size on smartphone chips nowadays.

In this work, we aim to address the dilemma between the performance improvement of the LUT model and the rapid growth in storage by proposing a novel LUT compression framework, named Diagonal-First Compression (DFC). By collecting occurrence statistics of LQ patches, we observe a diagonal-dominance property that most cached LQ patches distribute along the diagonal of a LUT, which reveals the sparsity of the cached pairs in LUT-based restoration models. Based on this observation, we first split diagonal and non-diagonal pairs according to the difference between pixels in each LQ patch. Then, we preserve the diagonal pairs with a diagonal re-indexing strategy and aggressively subsample the non-diagonal pairs with a large sampling interval to reduce the storage. This way, we convert one LUT into two LUTs with much smaller storage costs, while maintaining the representation capacity of the original LUT.

We examine the effectiveness of our DFC framework on representative image restoration tasks such as super-resolution, denoising, deblocking and deblurring. Extensive experiments demonstrate that DFC can significantly reduce the storage size of LUT-based models while maintaining their performance. For example, as shown in Fig. 1, the size of LUT-based models can be compressed with DFC to about 1/10 of the original size, facilitating their deployment on the cache of chips for efficient inference, yet with only a negligible drop on PSNR for $\times 4$ super-resolution.

The main contributions of this work are as follows:

1) We reveal the redundancy of the cached LQ-HQ pairs in LUT-based restoration models by observing that most occurrence statistics are distributed along the diagonal of the learned LUTs.

2) We propose a diagonal-first compression framework, which compresses one LUT into two smaller LUTs, by designing a diagonal re-indexing strategy to preserve representation capacity and a non-diagonal subsampling strategy to reduce redundancy.

3) We design a new structure based on multiple LUTs that achieves advanced performance, which is used together with existing LUT-based models for the evaluation of our compression framework.

4) Quantitative and qualitative results show that the proposed framework effectively reduces storage costs and maintains the performance of the original LUT-based models, achieving a better performance-storage trade-off.

2. Related Work

2.1. Image Restoration

Image restoration includes techniques for image super-resolution, denoising, deblocking, and deblurring, aims at enhancing image quality by improving resolution, reducing noise, and rectifying imperfections such as blurring and blocking. Classical methods [6, 10, 11, 14, 17, 24, 41, 42, 44] have been extensively studied. With the development of deep learning, many DNN-based methods [7–9, 12, 15, 28–30, 40, 43, 47–49] have achieved significant restoration performance. However, these methods bring heavy computational and storage costs by building strong network backbones with a large number of learnable parameters.

2.2. Look-Up Table

Look-Up Table (LUT) is a widely-used mapping operator, especially for color manipulation in the image processing pipeline [23, 33, 37]. A LUT is a data structure composed of index-value pairs. It can be saved as a high-dimensional matrix where the index plays the role of the coordinate and the value is stored in the matrix cell. Here, we categorize LUT into channel-wise LUT and spatial-wise LUT.

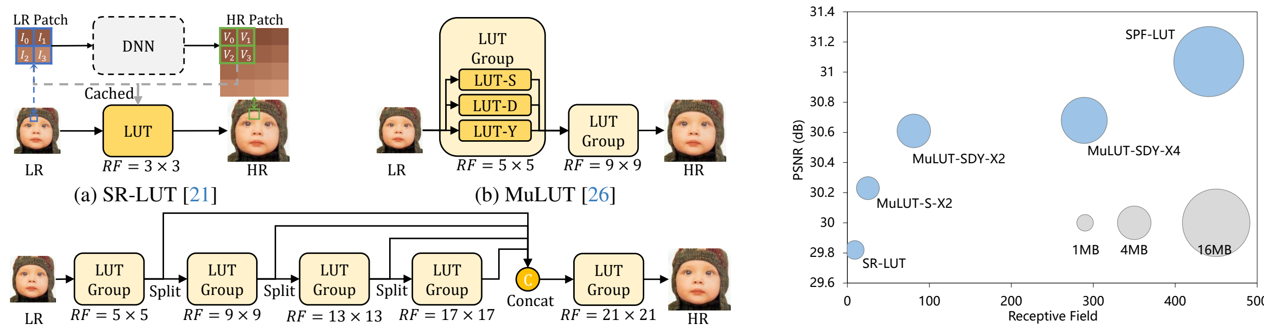
Channel-wise LUT. The widely-used channel-wise 3D LUT achieves color-to-color mapping by querying the source RGB color as a coordinate to locate the corresponding target RGB color saved in a LUT cell. Zeng et al. [45] propose image-adaptive 3D LUTs, which achieve flexible channel-wise mapping with learnable color-to-color LUTs. Zhang et al. [46] propose a compressed representation of image-adaptive 3D LUTs and build a CLUT-Net to reduce the redundancy in the LUTs. CLUT-Net explores the redundancy in channel-wise LUTs via the color-wise correlation, and thus it is not suitable for restoration tasks. In this work, we focus on the redundancy in spatial-wise LUTs.

Spatial-wise LUT. SR-LUT [21] is the first to introduce a spatial-wise LUT for patch-to-patch mapping, which uses a local LR patch composed of spatially adjacent pixels as the coordinate index to retrieve the corresponding HR patch cached in a LUT cell. To improve the performance of SR-LUT, MuLUT [26] adopts multiple SR-LUT variants as a LUT group and cascades the LUT groups. Since caching HR patches outputted by DNN into LUTs will cause performance degradation, MuLUT also proposes a LUT-aware finetuning strategy to reduce the performance gap. SPLUT [32] proposes a serial-parallel structure by using multiple LUTs, which processes different image information separately. RCLUT [31] proposes a plugin module to improve LUT-based models with a slight increase in size. To maintain performance with reduced storage size, our LUT compression framework exploits the compressibility of spatial-wise LUTs.

3. Motivation

3.1. Scaling Law of Spatial-wise LUT

As a pioneer structure, SR-LUT [21] is illustrated in Fig. 2a. First, a DNN is trained for super-resolution. Then, SR-



(a) SR-LUT [21] (b) MuLUT [26] (c) SPF-LUT (d) Scaling law of spatial-wise LUTs

Figure 2. Comparison of SR-LUT, MuLUT, and the proposed SPF-LUT. (a) A super-resolution network with a limited receptive field (RF) size for the $\times 2$ super-resolution task is trained, and then exhaustive LR-HR patches are cached in the SR-LUT. At the inference phase, SR-LUT predicts HR results by using LR input patches to retrieve HR patches. (b) MuLUT cascades multiple LUT groups composed of three variants of SR-LUT to expand the RF size, like DNN. (c) Based on LUT groups, SPF-LUT splits output channels for progressive RF size enlargement and multi-scale feature fusion, respectively. (d) The size of the bubble represents the storage size. The PSNR is evaluated on the Set5 $\times 4$ dataset [3].

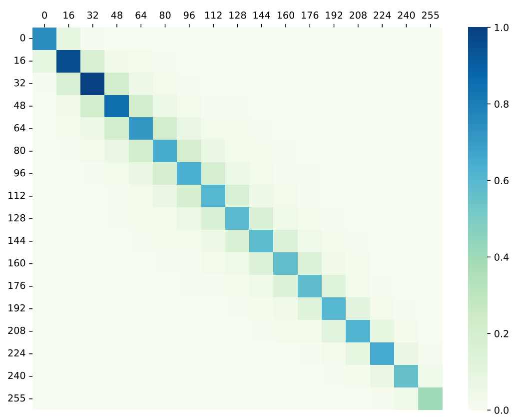


Figure 3. The visualization of the occurrence statistics of LQ patches in the downsampled DIV2K dataset [1], obtained by floor dividing each pixel value by 16 and then counting the occurrence frequency of adjacent pixel pairs. The darker color means a higher occurrence frequency.

LUT caches exhaustive LR-HR patches in its LUT cells by traversing LR inputs and pre-computing HR outputs of the DNN. Finally, an LR local patch with a shape of 2×2 is used as the index of the 4-dimensional (4D) LUT to obtain an HR patch as a prediction. Since a full-size LUT is not practical, with a total of 256^4 possible indexes (64GB for $\times 4$ super-resolution tasks), the 4D SR-LUT is uniformly subsampled with a sampling interval.

The receptive field (RF) size of SR-LUT is determined by index dimension. Since the size of the LUT increases exponentially with the index dimension, this dimension is set very small. Although the rotation ensemble trick is adopted to rotate a small 2×2 input patch 4 times and ensemble the lookup results, the RF size of SR-LUT is still limited to 3×3. The limited RF size results in a large gap between the performance of SR-LUT and advanced DNN models. As illustrated in Fig. 2b, MuLUT [26] proposes to parallelize and cascade multiple LUTs to enlarge the RF size.

As observed in Fig. 2d, MuLUT-S-X2 and MuLUT-SDY-X2 (hereinafter denoted as MuLUT) significantly outperform SR-LUT by expanding the RF size, which explores a scaling law of the spatial-wise LUT: more LUTs with a larger RF size lead to better performance.

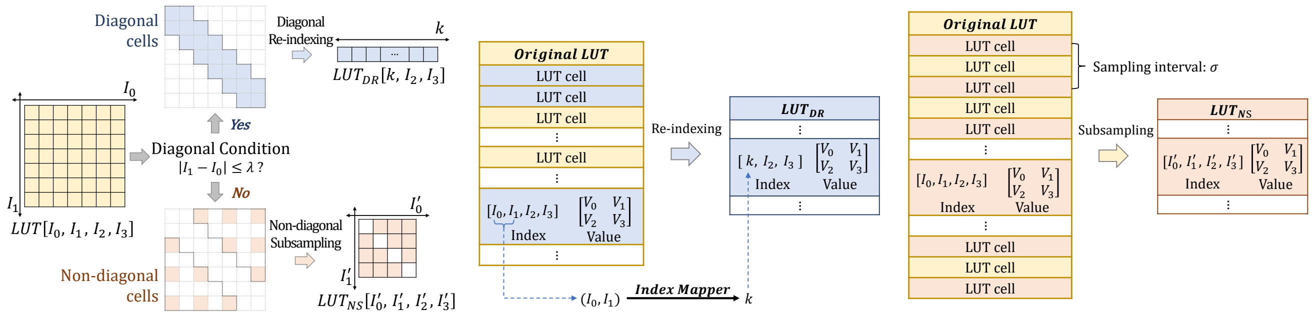
3.2. Spatial Progressive Fusion LUT

As shown in Fig. 2d, although the RF size of MuLUT grows linearly with the number of LUT groups, the performance improvement achieved by expanding the RF size with more LUTs is diminishing (see MuLUT-SDY-X4). We attribute this phenomenon to the bottleneck structure in the design of MuLUT, where the LUT groups only process single-channel features and thus restrict the scaling law due to the lack of feature diversity for image restoration.

Inspired by the network architecture in IMDN [20], here we design a new structure based on multiple LUTs, named Spatial Progressive Fusion LUT (SPF-LUT), to extend the scaling law from the perspective of both RF size and feature diversity. As illustrated in Fig. 2c, different from MuLUT and SR-LUT, SPF-LUT predicts multiple-channel features from LUT groups, and these features are split into two parts: one part is left for multi-scale feature fusion, and the other part is fed to the next LUT group to progressively expand the RF size. Then, we concatenate feature maps with different RF sizes for multi-scale feature fusion to output the target super-resolution results. As shown in Fig. 2d, our SPF-LUT obtains advanced restoration performance thanks to a larger RF size of 21×21 and the improved diversity of multi-scale features, compared to existing LUT-based models. More details about SPF-LUT are provided in the supplementary material.

3.3. The Dilemma and Our Observation

Nevertheless, the LUT-based models face the dilemma that the increasing use of LUTs to improve performance re-



(a) Diagonal-First Compression (b) Diagonal Re-indexing Strategy (c) Non-diagonal Subsampling Strategy

Figure 4. Overview of our Diagonal-First Compression (DFC) framework. The figure illustrates the process of compressing the first two dimensions of $LUT[I_0, I_1, I_2, I_3]$, where (I_0, I_1, I_2, I_3) are the indexes of the four dimensions. (a) Our DFC divides the diagonal and non-diagonal cells according to a diagonal condition and obtains two LUTs with smaller storage sizes, $LUT_{DR}[k, I_2, I_3]$ and $LUT_{NS}[I'_0, I'_1, I'_2, I'_3]$, respectively, through the diagonal re-indexing strategy and the non-diagonal subsampling strategy. (b) In the diagonal re-indexing strategy, values in the cells whose indexes satisfy the diagonal condition are preserved and carefully re-indexed using Index Mapper, resulting in a lower-dimensional LUT_{DR}. (c) In the non-diagonal subsampling strategy, LUT cells are aggressively subsampled at large intervals to save storage, resulting in a sparser LUT_{NS} that can retrieve the values cached in non-diagonal LUT cells.

sults in rapidly growing storage size, which is ultimately restricted by the allocatable on-device memory, hindering the deployment of these models on edge devices. As depicted by the bubble size in Fig. 2d, the storage size of SPF-LUT is 17.284MB for $\times 4$ super-resolution, which is 4 times the storage size of MuLUT (4.062MB) and 14 times that of SR-LUT (1.274MB).

We argue that this dilemma can be significantly mitigated by reducing the redundancy of saved LUTs. Intrinsically, the spatial-wise LUT realizes the "space for time" strategy by traversing all possible combinations of input local patches. This strategy ignores the low-dimensional manifold distribution of natural image data, which is empirically observed in previous works [4, 5, 16, 25, 36], resulting in redundancy in storage space.

To validate the above argument, we reveal this redundancy in the learned spatial-wise LUT by observing statistics of patch occurrence. First, we collect retrieval statistics in the LUT by counting the occurrence frequency of pairs of two spatially adjacent pixels in LQ patches. The occurrence frequency of adjacent pixel pairs reflects the frequency at which the indexes of the LUT cells are used for retrieval. The high occurrence frequency means that the values cached in the corresponding LUT cells are frequently accessed. Next, we visualize the retrieval statistics in Fig. 3. As illustrated in Fig. 3, cells with a darker color, which represents a higher occurrence frequency, are primarily distributed along the diagonal. We refer to this observation as the diagonal-dominance property of the spatial-wise LUT. Since the coordinate index of a spatial-wise LUT is composed of the adjacent pixels in an LQ patch, the diagonal-dominance property means that values of adjacent pixels in most local patches are very close, which is consistent with the low-manifold distribution of natural image data [4, 5, 16, 25]. Thus, we make a local smoothness assumption on LQ patches, and we split diagonal and non-diagonal LUT cells according to the difference between indexes of the LUT cells (*i.e.* adjacent pixels).

4. Diagonal-First Compression

4.1. Overview

A spatial-wise LUT can be saved as a high-dimensional matrix, whose storage size can be calculated as

$$S = (2^{8-q} + 1)^n \times mB, \tag{1}$$

where q is the uniform sampling interval, n is the number of the coordinate index dimension of the LUT, and m is the number of cached values in each LUT cell, *e.g.*, $m = 4 \times 4 = 16$ for $\times 4$ super-resolution. Here, we take the 4D spatial-wise LUT with an original sampling interval of 4 as an example (*i.e.* $n = 4$ and $q = 4$ in the Eq. 1), which is also adopted by existing methods [21, 26]. Following these methods, we formulate a 4D spatial-wise LUT as $LUT[I_0, I_1, I_2, I_3]$, with the shape of $L \times L \times L \times L$, where I_0, I_1, I_2, and I_3 are the coordinate indexes of the four dimensions, respectively, and L is the size of each dimension. Here, L can be calculated as $L = 2^{8-q} + 1$.

Our proposed LUT compression framework, DFC, compresses the original spatial-wise LUT to achieve a smaller storage size in two steps: diagonal re-indexing and non-diagonal subsampling. As illustrated in Fig. 4a, we take the first two dimensions (I_0 and I_1) of a 4D spatial-wise LUT as an example, visualizing the LUT as a 2D grid. First, in the diagonal re-indexing step, the values cached in diagonal LUT cells whose indexes satisfy a diagonal condition are preserved and re-indexed, resulting in a lower-dimensional LUT_{DR}. Then, in the non-diagonal subsampling step, the values cached in non-diagonal LUT cells are aggressively subsampled with larger sampling intervals than the original interval q, resulting in a sparser LUT_{NS}.

4.2. Diagonal Re-indexing Strategy

The first step of our framework is to reduce the number of dimensions of a spatial-wise LUT. Based on the diagonal-dominance property, we propose a diagonal re-indexing strategy to map indexes satisfying a diagonal condition to a new dimension using an Index Mapper, thereby reducing the dimension of a LUT.

As illustrated in Fig. 4a, the following diagonal condition is used to judge whether a LUT cell is a diagonal LUT cell:

$$|I_0 - I_1| \leq \lambda, \tag{2}$$

where λ is defined as the diagonal width. As shown in Fig. 4b, all the values cached in diagonal LUT cells whose indexes of I_0 and I_1 satisfy Eq. 2 are sequentially preserved in a Diagonal Re-indexing LUT, *i.e.*, LUT_{DR}. LUT_{DR} is formulated as $LUT_{DR}[k, I_2, I_3]$ with the shape of $K \times L \times L$, where k is the new index generated by counting while enumerating all diagonal LUT cells, and K is the number of diagonal cells. Here, K can be calculated as $K = (2\lambda + 1)L - \lambda(\lambda + 1)$. Thus, we compress the first two dimensions of the 4D LUT to one dimension by mapping the index (I_0, I_1) to the index k. An Index Mapper acts as a function establishing the re-indexing relationship between the index (I_0, I_1) and the index k by the following rule:

$$k = I_1 \times (2\lambda + 1) + r_1 - 1, \tag{3}$$

where r_1 can be viewed as the relative distance between I_0 and I_1, which is calculated as $r_1 = I_0 - I_1 + \lambda$ $(0 \leq r_1 \leq 2\lambda)$. Then, the index (k, I_2, I_3) is used to retrieve the 3D $LUT_{DR}[k, I_2, I_3]$. In practice, we can easily extend this strategy to compress more dimensions.

4.3. Non-diagonal Subsampling Strategy

The second step of our framework is to aggressively subsample the values cached in the non-diagonal LUT cells at a large sampling interval, since the values cached in the non-diagonal LUT cells are rarely accessed, according to our observation in Fig. 3.

As illustrated in Fig. 4c, by non-diagonal subsampling at a large sampling interval, we reduce the size of each dimension from L to D, resulting in a Non-diagonal Subsampling LUT, *i.e.*, LUT_{NS}, with the shape of $D \times D \times D \times D$. D is calculated as $D = 2^{8-\sigma} + 1$, where σ is a large sampling interval. LUT_{NS} is formulated as $LUT_{NS}[I'_0, I'_1, I'_2, I'_3]$. It should be noted that values in some diagonal LUT cells are also subsampled into LUT_{NS} to predict the values whose indexes do not satisfy the diagonal condition but are near the diagonal boundary.

4.4. Theoretical Analysis of Compression Ratio

Our proposed DFC framework compresses a single spatial-wise LUT into two LUTs, LUT_{DR} and LUT_{NS}, with a smaller total storage size. The shape of LUT_{DR} is $K \times L^{n-p}$, where p is the number of compressed dimensions, and the shape of LUT_{NS} is D^n. Then, the storage size of LUT_{DR} can be calculated as

$$S_{DR} = K \times L^{n-p} \times mB = K \times (2^{8-q} + 1)^{n-p} \times mB, \tag{4}$$

where $0 \leq p \leq n$, and the number of diagonal LUT cells K is determined by the diagonal width λ and the number of compressed dimensions p. The storage size of LUT_{NS} can be calculated as

$$S_{NS} = D^n \times mB = (2^{8-\sigma})^n \times mB, \tag{5}$$

where $q \leq \sigma < 8$, and q is the sampling interval of the uncompressed original LUT. Finally, we define the Compression Ratio (CR) as

$$CR = (S_{DR} + S_{NS})/S \times 100\%, \tag{6}$$

where S is the storage size of the original LUT in Eq. 1. As indicated in Eq. 4 and Eq. 5, CR monotonically increases with λ and monotonically decreases with p and σ.

5. Experiments and Results

5.1. Evaluation Settings

In the evaluation of our DFC framework, we conduct experiments on representative image restoration tasks, where we train three LUT-based models (SR-LUT [21], MuLUT [26], and our SPF-LUT) on the widely used DIV2K [1] dataset for super-resolution, denoising, and deblocking, and train them on GoPro [38] training set for deblurring. We adapt these LUT-based models to denoising, deblocking, and deblurring by removing the PixelShuffle layer [27]. The three LUT-based models are trained for 2×10^5 iterations using Adam optimizer in the cosine annealing schedule with learning rate of 1×10^{-4} and batch size of 16. We randomly crop images into 48×48 patches, and augment the dataset with random rotation and flipping.

We compress the trained original LUT-based models using our proposed DFC to obtain their compressed versions, denoted as +DFC. The LUT-aware finetuning strategy [26] is adopted after DFC. In order to keep a small compression ratio, we set a configuration of ($p = 4$, $\lambda = 2$, $\sigma = 5$) as default for the +DFC version when evaluating performance. We report the main results as follows, and more comparisons and analyses are in the supplementary material.

5.2. Image Super-Resolution

We evaluate our DFC framework on five commonly used benchmark datasets for $\times 4$ super-resolution: Set5, Set14, BSDS100 [34], Urban100 [18], and Manga109 [35]. The degraded images are generated by bicubic downsampling. We test peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) for quantitative evaluation.

We select SR-LUT [21], MuLUT [26], and our SPF-LUT as the original versions of LUT-based models for compression and compare the performance with their +DFC ver-

Table 1. Quantitative comparison of PSNR/SSIM and storage size on standard benchmark datasets for $\times 4$ super-resolution. The gray background means LUT-based models are compressed using DFC. The storage size is greatly reduced with DFC, while the performance is maintained. For LUT-based models, the best and second-best results are depicted with red and blue, respectively.

	Method	*Storage Size*	*Set5*	*Set14*	*BSDS100*	*Urban100*	*Manga109*
Classical	Bicubic	-	28.42/0.8101	26.00/0.7023	25.96/0.6672	23.14/0.6574	24.91/0.7871
	NE + LLE [6]	1.434MB	29.62/0.8404	26.82/0.7346	26.49/0.6970	23.84/0.6942	26.10/0.8195
	ANR [41]	1.434MB	29.70/0.8422	26.86/0.7368	26.52/0.6992	23.89/0.6964	26.18/0.8214
	A+ [42]	15.17MB	30.27/0.8602	27.30/0.7498	26.73/0.7088	24.33/0.7189	26.91/0.8480
LUT	SR-LUT [21]	1.274MB	29.94/0.8524	27.18/0.7416	26.59/0.6999	24.09/0.7053	26.94/0.8454
	SR-LUT [21] +DFC	0.128MB	29.88/0.8501	27.14/0.7394	26.57/0.6982	24.05/0.7021	26.87/0.8423
	MuLUT [26]	4.062MB	30.60/0.8653	27.60/0.7541	26.86/0.7110	24.46/0.7194	27.90/0.8633
	MuLUT [26] +DFC	0.407MB	30.55/0.8642	27.56/0.7532	26.83/0.7104	24.41/0.7177	27.82/0.8613
	SPF-LUT	17.284MB	31.11/0.8764	27.92/0.7640	27.10/0.7197	24.87/0.7378	28.68/0.8796
	SPF-LUT +DFC	2.018MB	31.05/0.8755	27.88/0.7632	27.08/0.7190	24.81/0.7357	28.58/0.8779
DNN	RRDB [43]	63.942MB	32.68/0.8999	28.88/0.7891	27.82/0.7444	27.02/0.8146	31.57/0.9185
	EDSR [30]	164.396MB	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148

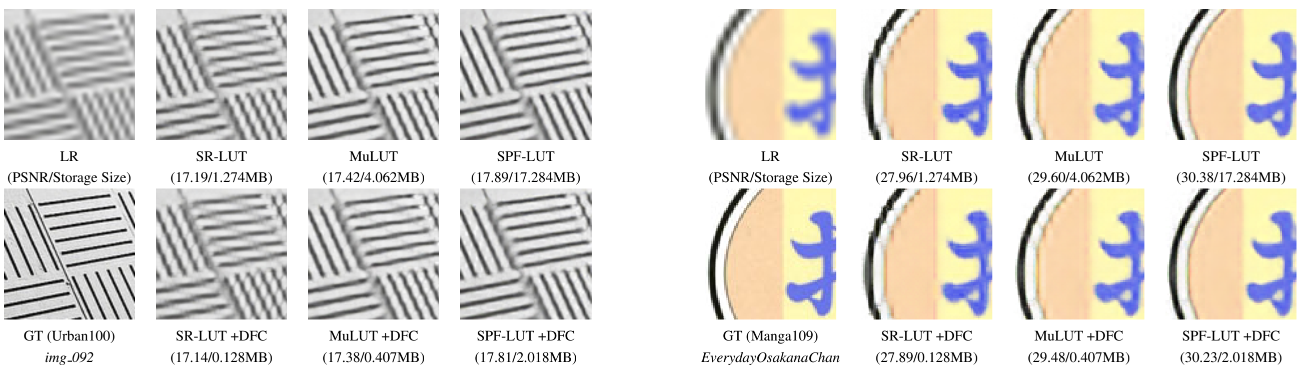


Figure 5. Qualitative comparison for $\times 4$ super-resolution on standard benchmark datasets [18, 35].

sions. Besides, we include classical models (Bicubic, NE + LLE [6], ANR [41], and A+ [42]), and DNN models (RRDB [43] and EDSR [30]) as references. We also report the storage size of DNN. It is worth noting that DNN models usually require a dedicated computing framework, *e.g.*, PyTorch libraries, and incur enormous computational overhead compared to LUT-based models [21, 26, 32].

The comparison of different models is listed in Table 1. As can be seen, the +DFC versions of LUT-based models significantly reduce the storage size compared to the original versions, with only negligible performance degradation in PSNR. For example, SPF-LUT +DFC achieves a compression ratio of 11.7%, and only a slight decrease of 0.06dB on the Set5 dataset. Furthermore, our DFC enables advanced LUT-based models to achieve better performance at a smaller storage size, *e.g.*, SPF-LUT +DFC outperforms MuLUT in PSNR with a smaller storage size (2.018MB vs. 4.062MB). Further efficiency evaluation and discussion on deployment are provided in the supplementary material.

We compare the visual quality of LUT-based models and their +DFC versions for the $\times 4$ super-resolution task in Fig. 5. The horizontal comparison validates our effort in extending the scaling law of LUT-based models that a larger RF size leads to better performance, which is consistent with Fig. 2d. For example, SPF-LUT (RF = 21×21) generates a cleaner and smoother edge (image *EverydayOsakanaChan* from Manga109) than SR-LUT (RF = 3×3) and MuLUT (RF = 9×9). The vertical comparison in Fig. 5 shows that images generated by the +DFC versions exhibit insignificant differences compared to those generated by the original versions. To summarize, our DFC maintains visual quality when compressing the storage size of LUT-based models.

5.3. Image Denoising

We evaluate LUT-based models on two benchmark datasets, Set12 [47] and BSD68 [34], for grayscale image denoising at a noise level of 15. The degraded images are generated with Additive Gaussian White Noise.

In Table 2, we report PSNR and the storage size of LUT-based models, providing the quantitative comparison. Classical models (BM3D [11], WNNM [17], and TNRD [10]) and DNN models (DnCNN [47], FFDNet [48], and SwinIR [29]) are also included as references. As

Table 2. The comparison of PSNR and storage size on standard benchmark datasets for grayscale image denoising at a noise level of 15. The gray background means LUT-based models are compressed using DFC. For LUT-based models, the best and second-best results are depicted with red and blue, respectively.

	Method	*Storage Size*	*Set12*	*BSD68*
LUT	SR-LUT [21]	81.563KB	30.42	29.78
	SR-LUT [21] +DFC	8.172KB	30.39	29.76
	MuLUT [26]	489.381KB	31.50	30.63
	MuLUT [26] +DFC	49.031KB	31.38	30.54
	SPF-LUT	3017.849KB	32.11	31.17
	SPF-LUT +DFC	595.926KB	32.01	31.09
Classical	BM3D [11]	-	32.37	31.07
	WNNM [17]	-	32.70	31.37
	TNRD [10]	-	32.50	31.42
DNN	DnCNN [47]	2239.117KB	32.86	31.73
	FFDNet [48]	1978.423KB	32.75	31.63
	SwinIR [29]	116.422MB	33.36	31.97

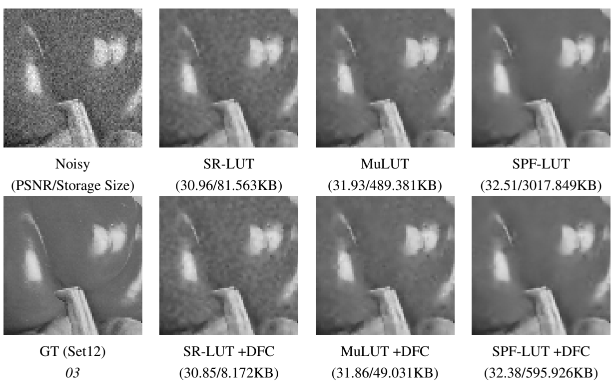


Figure 6. Qualitative comparison for grayscale image denoising at a noise level of 15 on standard benchmark datasets [47].

observed, the storage size of the LUT-based models is greatly reduced by using DFC, while the performance drops slightly on the two benchmark datasets. For example, MuLUT +DFC drops only 0.09dB with a compression ratio of 10.0% (49.031KB/489.381KB) on the BSD68 dataset, but achieves better denoising performance in a smaller storage size than the original SR-LUT (49.031KB vs. 81.563KB).

We provide the qualitative evaluation in Fig. 6. It yields a conclusion consistent with that of super-resolution, illustrating the generalizability of our framework.

5.4. Image Deblocking

Table 3 reports the quantitative comparison of PSNR-B of LUT-based models on two benchmark sets (Classic5 [14] and LIVE1 [39]) for image deblocking with a JPEG quality factor of 10, where the PSNR-B evaluates the blocking effects in images. We also include classical models (SA-DCT [14]) and DNN models (ARCNN [12] and SwinIR [29]) as references. We provide the qualitative evaluation in Fig. 7. This result indicates the adaptability of our framework to deblocking.

Table 3. The comparison of PSNR-B on standard benchmark datasets for image deblocking under a quality factor of 10. The gray background means LUT-based models are compressed using DFC. For LUT-based models, the best and second-best results are depicted with red and blue, respectively.

	Method	*Storage Size*	*Classic5*	*LIVE1*
LUT	SR-LUT [21]	81.563KB	27.58	27.69
	SR-LUT [21] +DFC	8.172KB	27.55	27.64
	MuLUT [26]	489.381KB	28.29	28.39
	MuLUT [26] +DFC	49.031KB	28.24	28.33
	SPF-LUT	3017.849KB	28.63	28.62
	SPF-LUT +DFC	595.926KB	28.62	28.61
Classical	JPEG	-	25.21	25.33
	SA-DCT [14]	-	28.15	28.01
DNN	ARCNN [12]	415.812KB	28.76	28.77
	SwinIR [29]	97.560MB	29.95	29.50

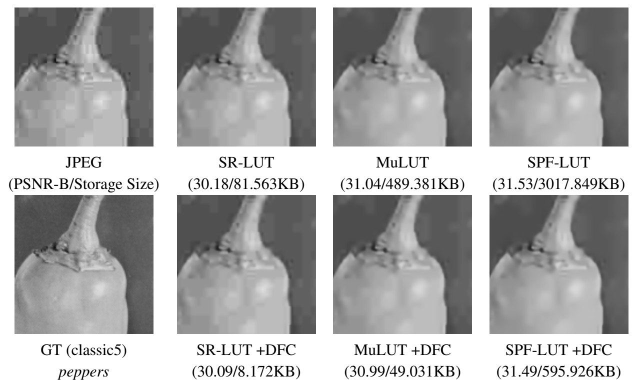


Figure 7. Qualitative comparison for image deblocking under the quality factor of 10 on standard benchmark datasets [14].

5.5. Image Deblurring

Table 4 provides the quantitative comparison of LUT-based models and also includes classical models (Xu et al. [44] and Kim and Lee [24]) and DNN models (Gong et al. [15] and DBGAN [49]) as references. We test PSNR and SSIM on the benchmark dataset, GoPro [38], for image deblurring. We provide the qualitative evaluation in Fig. 8. This result validates the generalizability of our framework again.

6. Ablation Analysis

In order to independently reveal the impact of different configurations of our DFC, the following ablation experiments are conducted without LUT-aware finetuning.

Diagonal width. We conduct experiments with different diagonal widths λ on the Set5 $\times 4$ dataset by setting $\sigma = 6$ and $p = 4$. As illustrated in Fig. 9, when λ increases from 1 to 5, the performance in PSNR rapidly improves, indicating that the information in the diagonal cells is crucial for maintaining performance. As λ increases from 5 to 11, the rate

Table 4. The comparison of PSNR/SSIM on the GoPro test set for image deblurring. The gray background means LUT-based models are compressed using DFC. For LUT models, the best and second-best results are depicted with red and blue, respectively.

	Method	*Storage Size*	*GoPro*
LUT	SR-LUT [21]	81.563KB	25.69/0.8598
	SR-LUT [21] +DFC	8.172KB	25.68/0.8592
	MuLUT [26]	489.381KB	25.74/0.8604
	MuLUT [26] +DFC	49.031KB	25.73/0.8604
	SPF-LUT	3017.849KB	25.94/0.8640
	SPF-LUT +DFC	595.926KB	25.92/0.8627
Classical	Xu et al. [44]	-	21.00/0.7410
	Kim and Lee [24]	-	23.64/0.8239
DNN	Gong et al. [15]	-	26.06/0.8632
	DBGAN [49]	44.318MB	31.10/0.9420

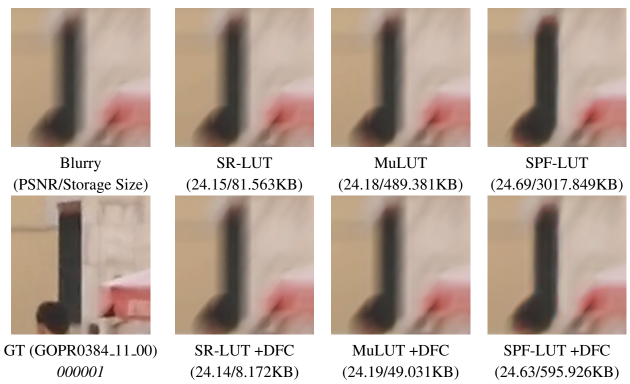


Figure 8. Qualitative comparison for image deblurring on standard benchmark datasets [38].

of performance improvement slows down. When $\lambda = 9$, the performance is saturated, and the compression ratio is about 60%, which indicates that at least 40% of the storage size in the uncompressed SPF-LUT is redundant, and the redundancy is present in the non-diagonal LUT cells.

The number of compressed dimensions. We conduct experiments with different numbers of compressed dimensions p. As listed in Table 5, when λ is set to 11, a lot of key information in diagonal LUT cells is preserved, so there is little impact on performance for different p, *e.g.*, only a 0.01dB change on the Set14 dataset. When λ is set to 5, key information in diagonal LUT cells is not preserved enough. In this case, when p increases from 2 to 4, the performance drops significantly, *e.g.*, a 0.38dB drop on the Set5 dataset.

Sampling interval of non-diagonal subsampling. We conduct experiments with different non-diagonal subsampling intervals σ. We change σ to 5, 6, and 7, corresponding to the LUT_{NS} shape of $9 \times 9 \times 9 \times 9$, $5 \times 5 \times 5 \times 5$ and $3 \times 3 \times 3 \times 3$, respectively. As shown in Table 5, the compressed versions of SPF-LUT show no significant change in performance on Set5 and Set14 datasets, *e.g.*, little difference in PSNR between $\sigma = 5$ and $\sigma = 7$ on Set5, indicating that a diagonal range with a certain diagonal width can contain almost all the crucial information.

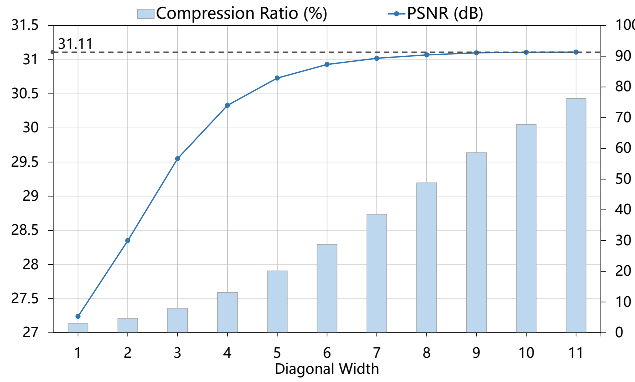


Figure 9. Ablation study on different diagonal widths λ of diagonal re-indexing strategy. The PSNR results are obtained by evaluating SPF-LUT +DFC on the Set5 [3] dataset for $\times 4$ super-resolution.

Table 5. Ablation study on different sampling intervals σ of non-diagonal subsampling and numbers of compressed dimensions p for the task of $\times 4$ super-resolution. λ means diagonal width.

	Configuration			*Storage Size*	*Set5*	*Set14*
	p	λ	σ			
SPF-LUT	-	-	-	17.284MB	31.11	27.92
SPF-LUT +DFC	2	11	6	15.650MB	31.11	27.92
	3	11	6	14.269MB	31.11	27.92
	4	11	6	13.176MB	31.11	27.91
	2	5	6	9.662MB	30.96	27.80
	3	5	6	5.650MB	30.81	27.68
	4	5	6	3.476MB	30.73	27.64
	4	11	5	14.381MB	31.11	27.92
	4	11	7	13.066MB	31.10	27.89

7. Conclusion

We propose a LUT compression framework, DFC, for efficient image restoration, addressing the dilemma between the performance improvement and the rapidly growing storage size of LUT-based models. Additionally, we also design a new structure, SPF-LUT, to further improve the performance of LUT-based models. Extensive experiments on four representative image restoration tasks demonstrates that our proposed LUT compression framework facilitates the deployment of advanced LUT-based models on resource-limited edge devices.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 62131003 and 62021001.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017. 3, 5

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 1

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 1, 3, 8

[4] Matthew Brand. Charting a manifold. In *NeurIPS*, 2002. 4

[5] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vis.*, 76:1–12, 2008. 4

[6] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 2, 6

[7] Chang Chen, Zhiwei Xiong, Xinmei Tian, and Feng Wu. Deep boosting for image denoising. In *ECCV*, 2018. 2

[8] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *CVPR*, 2019.

[9] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Real-world image denoising with deep boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(12):3071–3087, 2020. 2

[10] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6): 1256–1272, 2016. 2, 6, 7

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007. 2, 6, 7

[12] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *ICCV*, 2015. 2, 7

[13] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 1

[14] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Process.*, 16(5):1395–1411, 2007. 2, 7

[15] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 2, 7, 8

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 4

[17] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 2, 6, 7

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5, 6

[19] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *CVPR*, 2018. 1

[20] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 3

[21] Younghyun Jo and Seon Joo Kim. Practical single-image super-resolution using look-up table. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1

[23] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(12):2289–2302, 2012. 2

[24] Tae Kim and Kyoung Lee. Segmentation-free dynamic scene deblurring. In *CVPR*, 2014. 2, 7, 8

[25] Ann B Lee, Kim S Pedersen, and David Mumford. The non-linear statistics of high-contrast patches in natural images. *Int. J. Comput. Vis.*, 54:83–103, 2003. 4

[26] Jiacheng Li, Chang Chen, Zhen Cheng, and Zhiwei Xiong. Mulut: Cooperating multiple look-up tables for efficient image super-resolution. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[27] Jiacheng Li, Chang Chen, Zhen Cheng, and Zhiwei Xiong. Toward dnn of luts: Learning efficient image restoration with multiple look-up tables. *arXiv preprint arXiv:2303.14506*, 2023. 5

[28] Jiacheng Li, Chang Chen, Wei Huang, Zhiqiang Lang, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Learning steerable function for efficient image resampling. In *CVPR*, 2023. 2

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021. 1, 6, 7

[30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, 2017. 2, 6

[31] Guandu Liu, Yukang Ding, Mading Li, Ming Sun, Xing Wen, and Bin Wang. Reconstructed convolution module based look-up tables for efficient image super-resolution. In *ICCV*, 2023. 2

[32] Cheng Ma, Jingyi Zhang, Jie Zhou, and Jiwen Lu. Learning series-parallel lookup tables for efficient image super-resolution. In *ECCV*, 2022. 1, 2, 6

[33] Rafał Mantiuk, Scott Daly, and Louis Kerofsky. Display adaptive tone mapping. In *ACM SIGGRAPH 2008 papers*. 2008. 2

[34] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5, 6

[35] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multim. Tools Appl.*, 76:21811–21838, 2017. 5, 6

[36] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.*, 21(12):4695–4708, 2012. 4

[37] Jayanta Mukherjee and Sanjit K Mitra. Enhancement of color images by scaling the dct coefficients. *IEEE Trans. Image Process.*, 17(10):1783–1794, 2008. 2

[38] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 5, 7, 8

[39] H Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005. 7

[40] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015. 2

[41] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013. 2, 6

[42] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2015. 2, 6

[43] 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 *ECCV Workshops*, 2018. 2, 6

[44] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 2, 7, 8

[45] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(4):2058–2073, 2020. 2

[46] Fengyi Zhang, Hui Zeng, Tianjun Zhang, and Lin Zhang. Clut-net: Learning adaptively compressed representations of 3dluts for lightweight image enhancement. In *ACM MM*, 2022. 2

[47] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017. 2, 6, 7

[48] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.*, 27(9):4608–4622, 2018. 6, 7

[49] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020. 2, 7, 8

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

[51] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1