

# MuLUT: Cooperating Multiple Look-Up Tables for Efficient Image Super-Resolution

Jiacheng Li<sup>1\*</sup>, Chang Chen<sup>2\*</sup>, Zhen Cheng<sup>1</sup>, and Zhiwei Xiong<sup>1✉</sup>

<sup>1</sup> University of Science and Technology of China

{jcleee, mywander}@mail.ustc.edu.cn, zwxiong@ustc.edu.cn

<sup>2</sup> Huawei Noah's Ark Lab

chenchang25@huawei.com

**Abstract.** The high-resolution screen of edge devices stimulates a strong demand for efficient image super-resolution (SR). An emerging research, SR-LUT, responds to this demand by marrying the look-up table (LUT) with learning-based SR methods. However, the size of a *single* LUT grows *exponentially* with the increase of its indexing capacity. Consequently, the receptive field of a single LUT is restricted, resulting in inferior performance. To address this issue, we extend SR-LUT by enabling the cooperation of *Multiple* LUTs, termed MuLUT. Firstly, we devise two novel complementary indexing patterns and construct multiple LUTs in parallel. Secondly, we propose a re-indexing mechanism to enable the hierarchical indexing between multiple LUTs. In these two ways, the total size of MuLUT is *linear* to its indexing capacity, yielding a practical method to obtain superior performance. We examine the advantage of MuLUT on five SR benchmarks. MuLUT achieves a significant improvement over SR-LUT, up to 1.1dB PSNR, while preserving its efficiency. Moreover, we extend MuLUT to address demosaicing of Bayer-patterned images, surpassing SR-LUT on two benchmarks by a large margin.

**Keywords:** image super-resolution, look-up table, image demosaicing

## 1 Introduction

Single-image super-resolution (SR) aims to restore a high-resolution (HR) image with high-frequency details from its low-resolution (LR) observation. Recent methods based on deep neural network (DNN) [10,26,11,34,69,68,53] have made impressive progress in restoration performance, yet usually at a cost of heavy computational burden. Although this can be alleviated by elaborated model designs or dedicated computing engines (*e.g.*, GPU and NPU), the hardware cost and power consumption still limit the deployment of existing SR methods on edge devices. Therefore, the growing number of high-resolution screens on edge devices (*e.g.*, smartphones and televisions) calls for a practical SR solution.

---

\* Equal contribution.

An emerging research, SR-LUT [23], responds to this demand by replacing the expensive computing with relatively cheap memory access of cached index-value pairs. Different from the existing lightweight SR methods based on dedicated computing engines [11,2,31], SR-LUT utilizes a single look-up table (LUT) to cache the exhaustive values for later retrieval, which are computed in advance by a learned SR network. This contributes to the power efficiency and inference speed. However, in practice, the size of LUT is limited by the on-device memory. For a single LUT, the size grows exponentially as the dimension of indexing entries (*i.e.*, indexing capacity) increases. This imposes a restriction on the indexing capacity as well as the corresponding receptive field (RF) size of the SR network to be cached, which is the main obstacle for performance improvement.

In this paper, we embrace the merits of SR-LUT and propose MuLUT to overcome its intrinsic limitation, by enabling the cooperation of **M**ultiple LUTs. Firstly, we devise two novel *complementary indexing* patterns and cooperate multiple LUTs in parallel. Correspondingly, we propose a multi-branch network structure to learn and generate SR results for caching. Secondly, we devise a cascaded framework to enable the *hierarchical indexing* between multiple LUTs, where a re-indexing mechanism is proposed to link between LUTs from different hierarchies. In the above two ways, the total size of MuLUT is linear to its indexing capacity, yielding a practical method to obtain superior performance.

Extensive experiments demonstrate a clear advantage of our proposed MuLUT compared with SR-LUT. On five standard SR benchmarks, MuLUT achieves up to 1.1dB PSNR improvement, approaching the performance of the lightweight FSRCNN model [11]. Meanwhile, MuLUT preserves the efficiency of SR-LUT, for example, the theoretical energy cost is about 100 times less than that of FSRCNN.

Moreover, to evaluate the versatility of MuLUT, we extend MuLUT to address demosaicing of Bayer-patterned images. Although SR-LUT can be directly applied to demosaicing, it yields inferior performance due to the subpixel shift between Bayer-patterned and HR images. Instead, we cooperate multiple LUTs with complementary indexing and hierarchical indexing, addressing this misalignment problem of the single LUT solution. As a result, MuLUT achieves over 6.0dB PSNR gain compared with SR-LUT on two widely-used benchmarks.

The contributions of this work are summarized as follows:

- 1) We devise two novel indexing patterns and a corresponding multi-branch network to enable the complementary indexing of multiple LUTs.
- 2) We devise a cascaded framework with a re-indexing mechanism to enable the hierarchical indexing of multiple LUTs.
- 3) Extensive experiments on SR demonstrate that MuLUT achieves a significant improvement in performance over SR-LUT while preserving the clear advantage in efficiency over DNNs, showing its practicality for edge devices.
- 4) We adapt MuLUT to the image demosaicing task and demonstrate its superiority over SR-LUT, showing the versatility of the proposed method.

## 2 Related Works

**Classical SR methods.** Interpolation-based methods, including nearest, bilinear, and bicubic [25], often produce blurry results because the interpolation weights are calculated without considering the local structure inside the image. Exemplar-based methods leverage a dataset of LR-HR image patch pairs [13,14,43,59,60], or exploit self-similarity inside the LR image [16,62]. Sparse coding methods learn a compact representation of the patches, showing promising results [63,64,50,51,44]. But, computing the sparse representation of the input patch is time-consuming. Other fast SR methods based on random forests [45], gradient field sharpening [49], and displacement field [52] are also explored. Nevertheless, these classical SR methods suffer either unsatisfying visual quality or time-consuming computations.

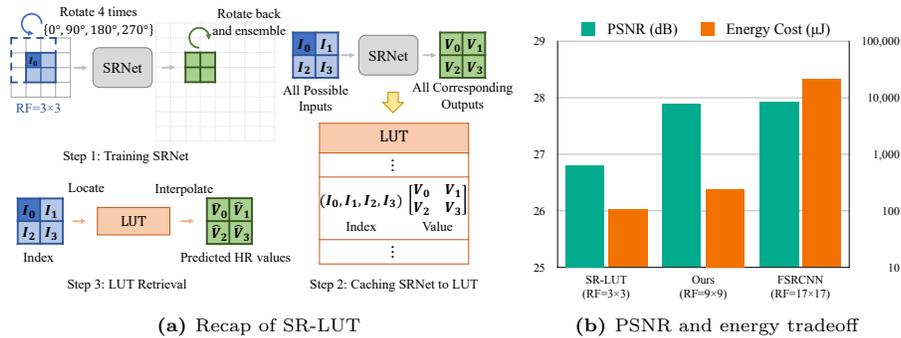
**Efficient SR networks.** With the rise of DNN methods, the community has made impressive progress in the task of SR [10,26,34,2,68,53,6,65,57,56,8,61,41]. However, it comes with a substantial computational burden of numerous floating-point operations. Thus, many efforts for efficient SR are conducted. Researchers elaborately design lightweight networks, including ESPCN [46], FSRCNN [11], CARN-M [2], IMDN [21], and LatticeNet [36], to name a few. General network compression methods like quantization [31,58], neural architecture search [9,48,30], network pruning [33], and AdderNet [7,47] have also been explored for efficient SR. Most recently, Jo *et al.* propose SR-LUT for practical SR [23]. They train a deep SR network with a restricted RF and then cache the output values of the learned SR network to the LUT, which are retrieved to obtain HR predictions at the test time. However, a single LUT yields inferior performance due to the restriction of the dimension of indexing entries, *i.e.*, the RF of the learned SR network. This is proved to be critical for SR [17]. Our method overcomes the intrinsic limitation of SR-LUT by enabling the cooperation of multiple LUTs.

**Image demosaicing.** Image demosaicing aims to produce colored observation from linear responses of light sensors inside the camera. It can be viewed as an SR problem with a particular color pattern. Interpolation-based methods like nearest and bilinear can also be used in image demosaicing. However, they tend to produce artifacts in the region with high-frequency signal changes. Classical methods taking advantage of the self-similarity inside the image [3,67,12,5] or relying on an optimization process [66,18,22] are proposed. Recently, DNN methods have been introduced to take advantage of powerful representations learned from large-scale datasets [54,15,29]. However, dedicated computing engines are required to execute numerous floating-point operations in DNNs. We adapt MuLUT to the image demosaicing task and show its versatility.

## 3 Cooperation of Multiple LUTs for SR

### 3.1 Preliminary

LUT is a widely-used mapping operator, especially for color manipulation and tone mapping modules in the image processing pipeline [40,37,27]. A LUT is

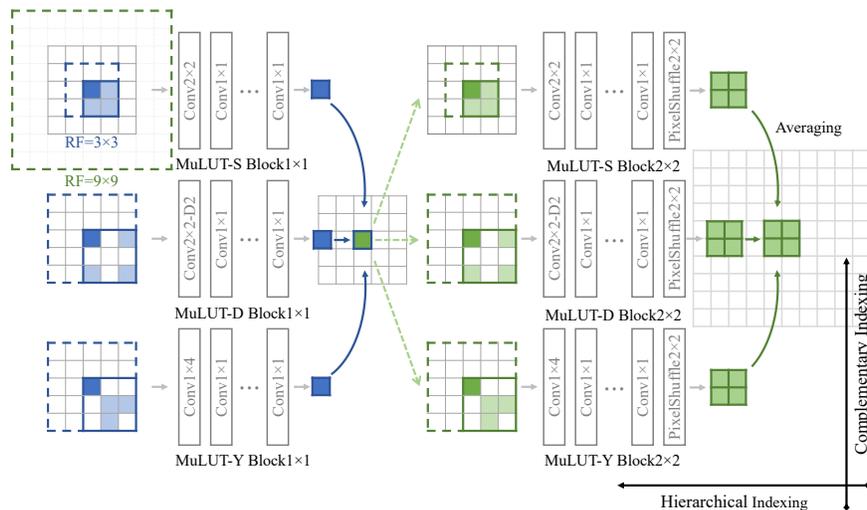


**Fig. 1:** (a) SR-LUT is obtained by caching the output values of a learned deep SR network with a restricted RF. At inference time, the precomputed HR output values are retrieved from the LUT for query LR input pixels. The indexing entries and corresponding HR values of a 4D LUT for  $2\times$  SR are marked in blue and green, respectively. The actual receptive area with the rotation ensemble trick are depicted with dashed lines. Please refer to the supplementary material and SR-LUT [23] for more details. (b) By cooperating multiple LUTs, we increase the RF from  $3\times 3$  to  $9\times 9$ , resulting in a significant performance improvement over SR-LUT while preserving its efficiency. The PSNR values are evaluated on Manga109 for  $4\times$  SR.

composed of pairs of indexes and values, which play as lookup indexing entries and interpolation candidates at the inference time, respectively. These paired indexing entries and values can be stored in the on-device memory, resulting in high execution efficiency. Recently, Jo *et al.* proposed SR-LUT, adopting LUT to the SR task. As illustrated in Fig. 1a, they firstly train a deep SR network. Then, the output values of the trained SR network are cached into a LUT via traversing all possible inputs. Finally, the HR predictions are obtained by locating LR input pixels and interpolating cached HR values. Due to the exponential growth of LUT size as the dimension of indexing entry increases, the authors impose a restriction on the indexing capacity of LUT, resulting in the limited RF of the SR network to be cached. Although with the rotation ensemble trick, where the input patch is rotated 4 times and the lookup results are ensemble, the RF size of SR-LUT is still limited to  $3\times 3$ . This limitation leads to inferior performance, since the RF size plays a critical role [17]. As shown in Fig. 1b, with increased RF size, the performance for SR can be significantly improved.

### 3.2 Overview

From the above observation, we propose to increase the indexing capacity by cooperating multiple LUTs, thus addressing the limitation of the RF. Specifically, as illustrated in Fig. 2, we propose two fundamental ways, *i.e.*, complementary indexing and hierarchical indexing, to generalize a single LUT to MuLUT, whose RF can be effectively enlarged by constructing multiple elementary components just like a neural network. To obtain multiple LUTs, we train a MuLUT network,



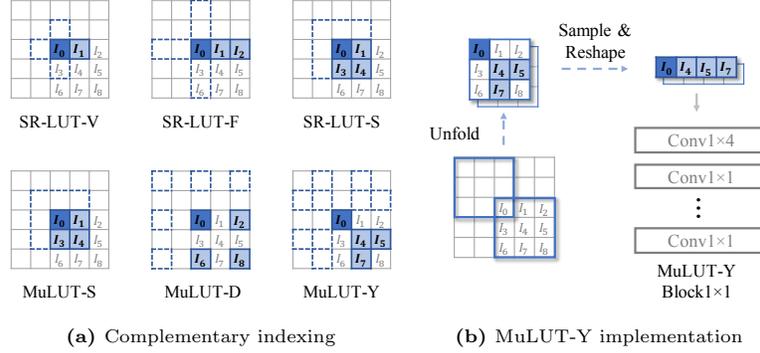
**Fig. 2:** Overview of MuLUT. Compared with a single LUT, MuLUT is able to greatly increase the RF size (*e.g.*, from  $3 \times 3$  to  $9 \times 9$ ). The MuLUT blocks are trained end-to-end and then cached to multiple LUTs. At inference time, these parallel and cascaded LUTs are retrieved with complementary indexing and hierarchical indexing in exactly the same order as the MuLUT blocks.

composed of multiple elementary MuLUT blocks. By parallelizing and cascading these MuLUT blocks, the RF and modeling capacity of the MuLUT networks increase, while the total size of cached LUTs grows linearly instead of exponentially. As shown in Fig. 2, the MuLUT network with 3 parallel blocks and 2 cascaded stages increases the RF size from  $3 \times 3$  to  $9 \times 9$  (9 times larger), while the total size of these LUTs is less than 4 times a single LUT. In contrast, the full size of a 25D LUT with an equivalent  $9 \times 9$  RF size is  $(2^8)^{25-4} = 2^{168}$  times a 4D LUT. In this way, MuLUT equips with a much larger indexing capacity without introducing the enormous cost of storage and computation.

During training, the MuLUT network is trained in an end-to-end manner. After caching the trained MuLUT network, MuLUT shares exactly the same structure of the original SR network and can be retrieved through complementary indexing and hierarchical indexing. In these two principled ways, we extend SR-LUT both in the width and depth dimensions and empower it to cache more complicated neural networks, taking advantage of both elaborate designs of deep neural networks and the high efficiency of LUT retrieval.

### 3.3 Parallelizing LUTs with Complementary Indexing

The first way we propose to increase the indexing capacity is parallelizing LUTs with complementary indexing. In SR, the surrounding pixels provide critical information to restore the high-frequency details, making it essential to cover



**Fig. 3:** Complementary indexing of multiple LUTs. With the proposed two novel indexing patterns, MuLUT covers more pixels than different variants of SR-LUT. The covered pixels with the rotation ensemble trick are marked with dashed boxes.

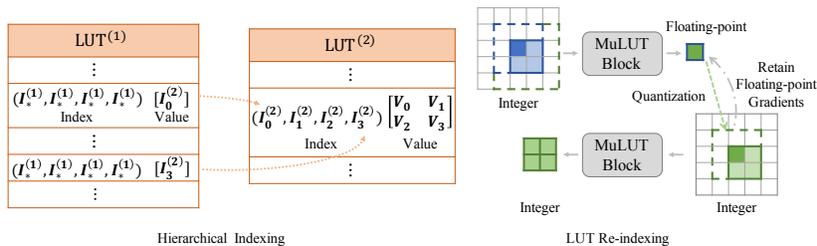
as many as input pixels for SR methods. Thus, we construct multiple LUTs with different indexing patterns in parallel, which are carefully designed to complement each other. For 4D LUTs, besides the standard indexing pattern, *i.e.*, MuLUT-S, we devise two novel indexing patterns (MuLUT-D and MuLUT-Y) with complementary covered pixels. As shown in Fig. 3a, the indexing pixels of MuLUT-S, MuLUT-D, and MuLUT-Y are  $(I_0, I_1, I_3, I_4)$ ,  $(I_0, I_2, I_6, I_8)$ , and  $(I_0, I_4, I_5, I_7)$ , respectively. Different from the variants of SR-LUT, our complementary design covers the whole  $5 \times 5$  area with the three types of MuLUT working together. Correspondingly, we propose a MuLUT network with multiple branches, where the parallel MuLUT blocks with complementary receptive areas are jointly trained. The cached LUTs, *i.e.*,  $LUT_S$ ,  $LUT_D$ , and  $LUT_Y$ , are retrieved in parallel, after which their predictions are averaged. Thus, for anchor  $I_0$ , the corresponding HR values  $\mathbf{V}$  are obtained by

$$\mathbf{V} = (LUT_S[I_0][I_1][I_3][I_4] + LUT_D[I_0][I_2][I_6][I_8] + LUT_Y[(I_0)[I_4][I_5][I_7])/3, \quad (1)$$

where  $LUT_*[\cdot]$  denotes the lookup and interpolation process in the LUT retrieval. In practice, the MuLUT-S block and MuLUT-D block can be implemented with standard convolutions, where the MuLUT-D block equips with an entry convolution with a dilation size of 2. As for MuLUT-Y, we implement it through the process illustrated in Fig. 3b. Precisely, we first unfold the input image by extracting  $3 \times 3$  patches with a sliding window. Then, we sample and reshape these “Y” shape pixels into  $1 \times 4$  vectors, which are fed into a standard convolution with a  $1 \times 4$  kernel. In summary, with complementary indexing of parallel LUTs, more surrounding pixels are involved to better capture the local structures, which help to predict the corresponding high-resolution observations.

### 3.4 Cascading LUTs with Hierarchical Indexing

The second way we propose to increase the indexing capacity is cascading LUTs with hierarchical indexing. As illustrated in Fig. 4 (left), with cascaded LUTs,



**Fig. 4:** Hierarchical indexing and LUT re-indexing mechanism. With LUT re-indexing, the behavior of LUT retrieval is involved in the learning process of the network with cascaded stages. Thus, the cascaded LUTs are able to reproduce its performance.

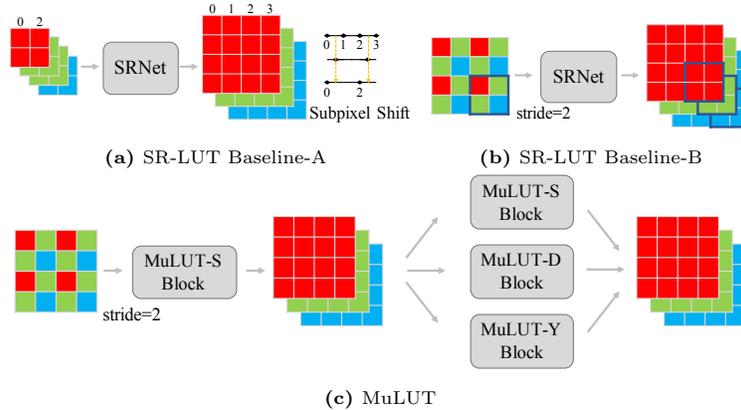
we conduct the lookup process in a hierarchical manner. The values ( $I_*^{(2)}$ ) in the previous LUT serve as the indexes of the following LUT. This hierarchical indexing process can be formulated as

$$\mathbf{V} = LUT^{(2)}[LUT^{(1)}[I_*]][LUT^{(1)}[I_*]][LUT^{(1)}[I_*]][LUT^{(1)}[I_*]]. \quad (2)$$

From the perspective of the RF, this cascaded framework is very similar to cascading multiple convolutional layers in a neural network. As shown in Fig. 4 (right), cascading two stages of MuLUT blocks increases the RF size from  $3 \times 3$  to  $4 \times 4$ . However, the indexes for image data are sampled and stored in the *int8* data type because of the constraint of LUT size, while training neural networks requires gradients in the *float* data type. Thus, we design a LUT re-indexing mechanism to integrate the behavior of hierarchical indexing in the learning process of MuLUT networks. Specifically, as shown in Fig. 4 (right), the prediction values of the previous MuLUT block are quantized to integers in the forward pass while their gradients are retained as floating-point values in the backward pass. This way, the cascaded LUTs are able to reproduce the performance of the cascaded MuLUT blocks. In practice, we adopt the dense connection [19] to help the convergence of MuLUT networks.

### 3.5 The LUT-aware Finetuning Strategy

In SR-LUT [23], due to the constraint of storage, the indexes of a LUT are uniformly sampled to reduce the LUT size, and the nonsampled indexes are approximated with nearest neighbors. Also, an interpolation process is performed to compute final predictions from weighted LUT values during LUT retrieval. The aforementioned loss of information causes a performance gap between the SR network and the cached LUT (See Table 4). Thus, we propose a LUT-aware finetuning strategy to address this issue. Specifically, we treat the values stored inside LUTs as trainable parameters and finetune them in a similar process to LUT re-indexing. After finetuning, the values inside LUTs are adapted to the sampling and interpolation process. This strategy is universal and serves as a practical improvement for MuLUT.



**Fig. 5:** The MuLUT network and SR-LUT baselines for image demosaicing. The co-operation of multiple LUTs enables the flexible design of the processing pipeline.

## 4 Extension of MuLUT to Demosaicing

MuLUT enables the flexible design of the processing pipeline for different vision tasks. Here, we take demosaicing Bayer-patterned images as an example. Image demosaicing can be viewed as an SR problem with a particular color pattern. However, there are grave obstacles in adopting SR-LUT to this task. Two straightforward solutions are shown in Fig. 5a and Fig. 5b, respectively. In Baseline-A, the pixels in the bayer pattern are treated as four independent channels, which are processed separately, and then the two green channels are averaged. As shown in Fig. 5a, this solution suffers from a subpixel shift of center points due to the misalignment between the HR pixels and the Bayer-patterned sampled ones. In Baseline-B, at a stride of 2, the  $2 \times 2$  Bayer-patterned blocks are upsampled into colored patches directly. But the limited RF of SR-LUT leads to the independent processing of these bayer blocks, thus failing to capture the inter-block patterns. In contrast, MuLUT is easy to be adapted to the characteristics of Bayer-patterned images. As shown in Fig. 5c, our MuLUT network resembles three color channels like Baseline-B in the first stage, and then integrates the surrounding pixels with three indexing patterns in the second stage. This multi-stage and multi-branch structure enabled by the cooperation of multiple LUTs addresses the above obstacles of adapting LUT to the task of image demosaicing effectively, showing the versatility of MuLUT.

## 5 Experiments and Results

### 5.1 Experimental Settings

**Datasets.** We train the MuLUT networks on the DIV2K dataset [1], which is widely used in the task of SR. The DIV2K dataset contains 800 training images and 100 validation images with 2K resolution. It covers multiple scenes and

PSNR/SSIM	Method	RF Size	Set5	Set14	BSDS100	Urban100	Manga109
Interpolation	Nearest	$1 \times 1$	26.25/0.7372	24.65/0.6529	25.03/0.6293	22.17/0.6154	23.45/0.7414
	Bilinear	$2 \times 2$	27.55/0.7884	25.42/0.6792	25.54/0.6460	22.69/0.6346	24.21/0.7666
	Bicubic	$4 \times 4$	28.42/0.8101	26.00/0.7023	25.96/0.6672	23.14/0.6574	24.91/0.7871
LUT	SR-LUT-S [23]	$3 \times 3$	29.82/0.8478	27.01/0.7355	26.53/0.6953	24.02/0.6990	26.80/0.8380
	MuLUT-SDY	$5 \times 5$	30.40/0.8600	27.48/0.7507	26.79/0.7088	24.31/0.7137	27.52/0.8551
	MuLUT-SDY-X2	$9 \times 9$	<b>30.60/0.8653</b>	<b>27.60/0.7541</b>	<b>26.86/0.7110</b>	<b>24.46/0.7194</b>	<b>27.90/0.8633</b>
Sparse coding	NE + LLE [4]	-	29.62/0.8404	26.82/0.7346	26.49/0.6970	23.84/0.6942	26.10/0.8195
	Zeyde et al. [64]	-	26.69/0.8429	26.90/0.7354	26.53/0.6968	23.90/0.6962	26.24/0.8241
	ANR [50]	-	29.70/0.8422	26.86/0.7368	26.52/0.6992	23.89/0.6964	26.18/0.8214
	A+ [51]	-	30.27/0.8602	27.30/0.7498	26.73/0.7088	24.33/0.7189	26.91/0.8480
DNN	FSRCNN [11]	$17 \times 17$	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
	CARN-M [2]	$45 \times 45$	31.82/0.8898	28.29/0.7747	27.42/0.7305	25.62/0.7694	29.85/0.8993
	RRDB [53]	$703 \times 703$	32.68/0.8999	28.88/0.7891	27.82/0.7444	27.02/0.8146	31.57/0.9185

**Table 1:** The comparison with other methods for  $4\times$  SR on standard benchmark datasets. With increased RF size, MuLUT achieves a significant improvement in restoration performance over SR-LUT.

encapsulates diverse patches. We evaluate our method with 5 well-recognized benchmark datasets: Set5, Set14, BSDS100 [38], Urban100 [20], and Manga109 [39]. For quantitative evaluation, we report peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [55], which are widely used for image quality assessment in terms of restoration fidelity. Besides, we compute the theoretical energy cost following AdderSR [47] to evaluate the efficiency tradeoff of our method as well as other solutions.

**Comparison methods.** We compare our method with various single-image SR methods, including interpolation-based methods (nearest neighbor, bilinear, and bicubic interpolation), sparse coding methods (NE + LLE [4], Zeyde et al. [64], ANR [50], and A+ [51]), SR-LUT [23], and DNN methods (FSRCNN [11], CARN-M [2], and RRDB [53]). Besides, to evaluate the efficiency of our method, we also compare the computation cost with the AdderNet [47] version and the quantized versions of VDSR [26] and CARN [2].

**Implementation details.** We train MuLUT networks with the Adam optimizer [28] in the cosine annealing schedule [35]. We use the mean-squared error (MSE) loss function as the optimization target. The MuLUT networks are trained for  $2 \times 10^5$  iterations at a batch size of 32. The cached LUTs are uniformly sampled with the interval  $2^4$ , *i.e.*, from LUT[256][256][256][256] to LUT[17][17][17][17]. After locating coordinates, the final prediction is obtained with 4D simplex interpolation [23], a 4D equivalent of 3D tetrahedral interpolation [24]. We further finetune the cached LUTs on the same training dataset for 2000 iterations with the proposed LUT-aware finetuning strategy.

## 5.2 Quantitative Evaluation

**Restoration performance.** The quantitative comparisons with other methods are listed in Table 1. The PSNR and SSIM values are computed at the Y-channel

	int8 Add.	int8 Mul.	int32 Add.	int32 Mul.	float32 Add.	float32 Mul.	Energy Cost(pJ)	Set14 PSNR	BSDS100 PSNR	Urban100 PSNR
Bilinear					7.4M	2.8M	29.8M	29.15	28.65	25.95
Bicubic					12.0M	10.1M	53.5M	30.23	29.53	26.86
SR-LUT-F [23]	13.6M	0.5M	11.8M	19.1M			61.0M	31.88	30.77	28.49
SR-LUT-S [23]	19.1M	0.5M	28.6M	22.8M			74.2M	31.73	30.64	28.50
MuLUT-SDY	56.9M	0.5M	118.0M	68.0M			224.3M	32.35	31.17	29.10
MuLUT-SDY-X2	80.6M	0.9M	109.4M	85.5M			278.5M	<b>32.49</b>	<b>31.23</b>	<b>29.31</b>
FSRCNN [11]					6.1G	6.1G	28.1G	32.69	31.49	29.87
A-VDSR-8bit [47]	1224.1G	1.1G					36.9G	32.85	31.66	30.07
A-VDSR [47]					1224.1G	1.1G	1105.6G	32.93	31.81	30.48
VDSR [26]					612.6G	612.6G	2817.9G	33.03	31.90	30.76
A-CARN-1/4 [47]					28.9G	0.1G	26.3G	-	-	30.21
CARN-1/4 [47]					14.5G	14.5G	66.5G	-	-	30.40
CARN-M [2]					91.2G	91.2G	419.5G	33.26	31.92	30.83

**Table 2:** The comparison of energy cost and performance for producing a  $1280 \times 720$  HD image through  $2 \times$  SR. The statistics of operations not involved in a method are leaved blank. A-VDSR denotes the AdderNet version of VDSR [26,47]. A-VDSR-8bit denotes performing 8bit quantization for A-VDSR. Our method shows superior performance (0.6~0.8dB) over SR-LUT, and a clear energy cost advantage (about  $100 \times$  less) compared with DNN methods, even with their AdderNet and quantized versions.

in the YCbCr color space. As can be seen, MuLUT boosts the performance of SR-LUT significantly. For example, with 2 cascaded stages and 3 parallel blocks, MuLUT-SDY-X2 improves the PSNR performance of a single LUT up to 1.1dB on the Manga109 dataset and exceeds FSRCNN in terms of SSIM. With only complementary indexing, MuLUT-SDY increases the RF size from  $3 \times 3$  to  $5 \times 5$ , boosting the PSNR value by about 0.6dB on the Set5 dataset. Overall, MuLUT obtains comparable or better performance compared with FSRCNN.

**Computational analysis.** Following the protocol in AdderSR [47], we estimate the theoretical energy cost of MuLUT. We calculate the statistics of multiplications and additions in different data types needed by each method and estimate their total energy cost. The detailed comparison is listed in Table 2. As can be seen, our method shows superior performance compared with interpolation methods and SR-LUT, while maintaining similar energy cost. For example, MuLUT-SDY-X2 exceeds SR-LUT by 0.6~0.8dB while maintaining comparable energy cost. On the other hand, MuLUT maintains the clear energy cost advantage over DNN methods, even with their AdderNet and quantized versions. Compared with FSRCNN, A-VDSR-8-bit, and A-CARN-1/4, MuLUT costs about  $100 \times$  less energy while achieving comparable restoration performance. In summary, MuLUT achieves a better performance and efficiency tradeoff, boosting the performance of SR-LUT significantly with similar computation cost.

We also report the running times of different methods in Table 3. As listed, MuLUT maintains the efficiency of SR-LUT, showing a clear advantage compared to sparse coding methods and DNN methods. Note that the CPU computing architecture is not optimized for LUT, which can be embedded into on-device

	Interpolation			LUT			Sparse Coding*		DNN		
	Nearest	Bilinear	Bicubic	SR-LUT-S	MuLUT-SDY	MuLUT-SDY-X2	ANR	A+	FSRCNN	CARN-M	RRDB
runtime(ms)	9	20	97	137	228	242	1715	1748	350	3300	23377
PSNR(dB)	23.45	24.21	24.91	26.80	27.52	27.90	26.18	26.91	27.91	29.85	31.57

**Table 3:** Runtime comparison for generating a  $1280 \times 720$  HD image through  $4 \times$  SR. \* denotes that the runtimes of sparse coding methods are from SR-LUT [23], which are tested on a desktop computer. All the other runtimes are measured on a Xiaomi 11 smartphone, among which the DNN methods are implemented in the CPU-version of the PyTorch library [42]. For SR-LUT-S, we test the official implementation provided by the authors. PSNR values are evaluated on Manga109.

memory such as those of image processors in consumer cameras for low-latency execution. Moreover, MuLUT can be implemented without modern computing libraries like PyTorch, thus having better practicality on edge devices.

### 5.3 Qualitative Evaluation

We compare the visual quality of our method (MuLUT-SDY-X2) with other methods in Fig. 6. In the first two examples, SR-LUT-S produces notable artifacts, *e.g.*, along the border of the hat (*baby* from Set5). Our method achieves similar visual quality as computation-heavy methods like A+ and FSRCNN. In the last two examples, our method is able to generate sharper edges and obtain better visual quality than A+ and FSRCNN, *e.g.*, the eyebrow of the character (*TetsuSan* from Manga109). To sum up, MuLUT achieves better visual quality than SR-LUT-S and comparable visual quality with A+ and FSRCNN. More visual results are provided in the supplementary material.

### 5.4 Ablation Studies

We conduct several ablation experiments to verify the effectiveness of MuLUT.

**The effectiveness of complementary indexing.** We conduct an experiment with combinations of different indexing patterns of parallel LUTs. As listed in Table 4, with MuLUT-S and MuLUT-D working together, MuLUT-SD is able to cover a region of  $5 \times 5$ , but not all pixels are covered. Still, it improves the performance of SR-LUT. Further, involving the novel MuLUT-Y with a “Y” shape indexing pattern, the MuLUT-SDY covers all pixels in a  $5 \times 5$  region and improves the performance, showing the effectiveness of complementary indexing.

**The effectiveness of hierarchical indexing.** We conduct an experiment with cascading different stages of LUTs. As listed in Table 4, cascading more stages increases the RF size steadily, and the performance improves accordingly. Without LUT re-indexing, the performance drops due to the inconsistency between the SR network and the cached LUT. Note that cascading LUTs involves *sub-linear* extra computational burden and storage space, since all LUTs except the ones in the last stage cache only one value for each index entry. Furthermore,

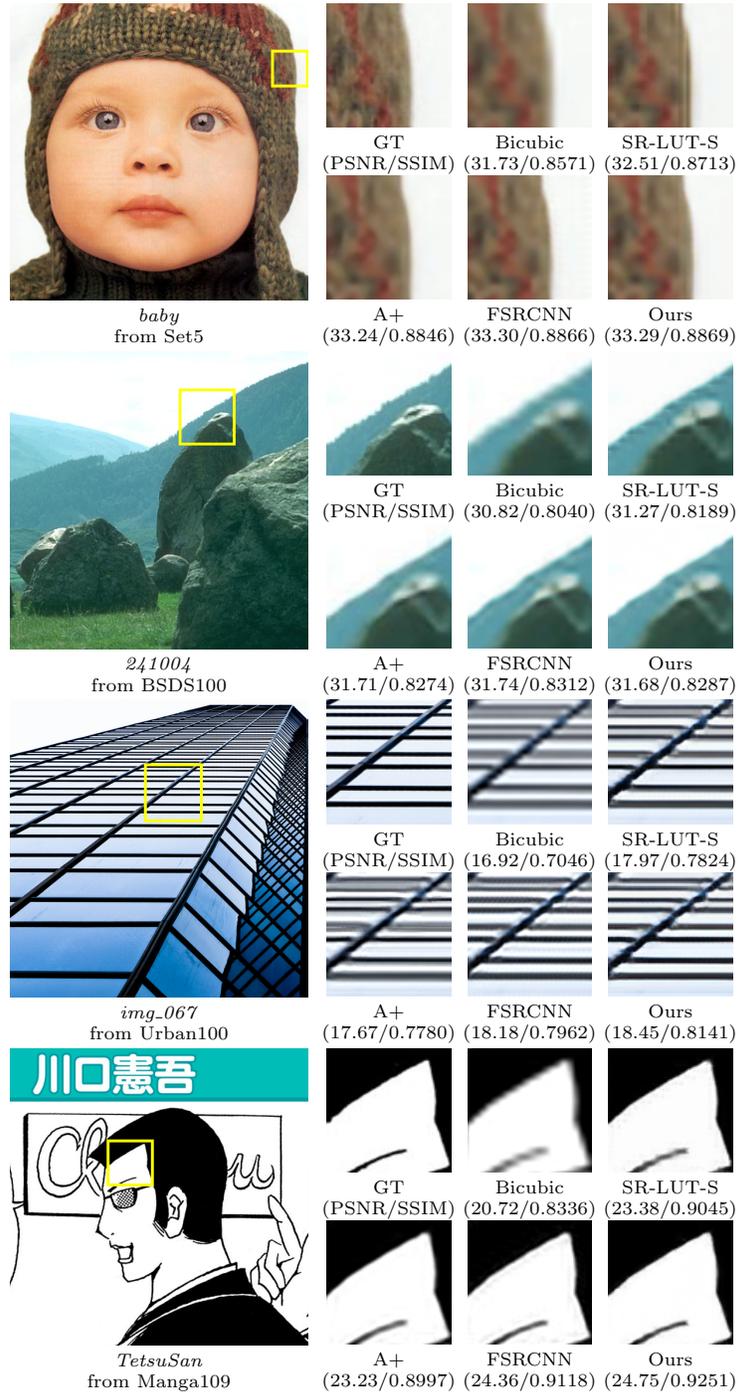


Fig. 6: Visual comparison for  $\times 4$  SR on standard benchmark datasets.

PSNR(dB)	Energy(pJ)	LUT Size	RF Size	Set5	Set14	BSDS100	Urban100	Manga109
SR-LUT-S	72.5M	1.274MB	$3 \times 3$	29.82	27.01	26.53	24.02	26.80
MuLUT-SD	149.2M	2.549MB	$5 \times 5$	30.31	27.41	26.75	24.25	27.38
MuLUT-SDY	222.3M	3.823MB	$5 \times 5$	30.40	27.48	26.79	24.32	27.52
MuLUT-S-X2 w/o re-index	78.0M	1.354MB	$4 \times 4$	30.11	27.26	26.64	24.15	27.02
MuLUT-S-X2	78.0M	1.354MB	$4 \times 4$	30.23	27.39	26.70	24.28	27.39
MuLUT-S-X3	83.4M	1.434MB	$5 \times 5$	30.31	27.42	26.73	24.31	27.54
MuLUT-S-X4	88.9M	1.513MB	$6 \times 6$	30.40	27.47	26.76	24.36	27.66
SR-LUT-S net.	-	-	$3 \times 3$	29.88	27.14	26.57	24.04	26.86
SR-LUT-S (4bit LUT) w/o ft.	72.5M	1.274MB	$3 \times 3$	29.82	27.01	26.53	24.02	26.80
SR-LUT-S (4bit LUT) w/ ft.	72.5M	1.274MB	$3 \times 3$	29.94	27.18	26.59	24.09	26.94
SR-LUT-S (3bit LUT) w/o ft.	72.5M	102.5KB	$3 \times 3$	29.58	26.99	26.49	23.95	26.76
SR-LUT-S (3bit LUT) w/ ft.	72.5M	102.5KB	$3 \times 3$	29.87	27.13	26.56	24.04	26.85
MuLUT-SDY-X2 net.	-	-	$9 \times 9$	30.61	27.61	26.86	24.47	27.93
MuLUT-SDY-X2 w/o ft.	233.6M	4.062MB	$9 \times 9$	30.52	27.55	26.83	24.40	27.83
MuLUT-SDY-X2	233.6M	4.062MB	$9 \times 9$	30.60	27.60	26.86	24.46	27.90

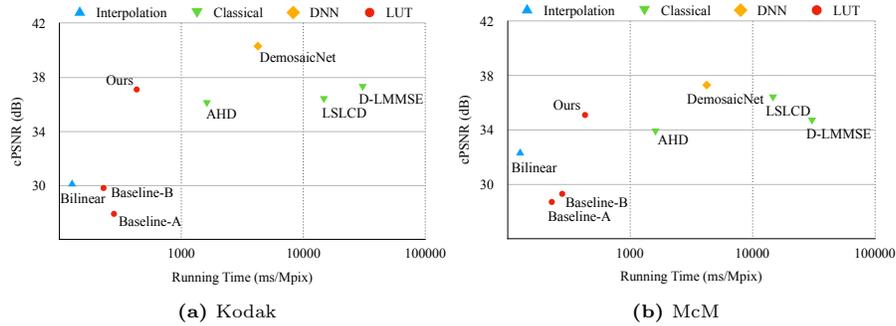
**Table 4:** Ablation studies on MuLUT for  $4 \times$  SR, where re-index denotes the LUT re-indexing mechanism, net. denotes the performance of corresponding neural network, 4bit denotes the sampling interval is  $2^4$  and 3bit the  $2^5$ , and ft. denotes the LUT-aware finetuning strategy.

with both complementary indexing and hierarchical indexing, MuLUT-SDY-X2 achieves better restoration performance, showing their orthogonal improvement.

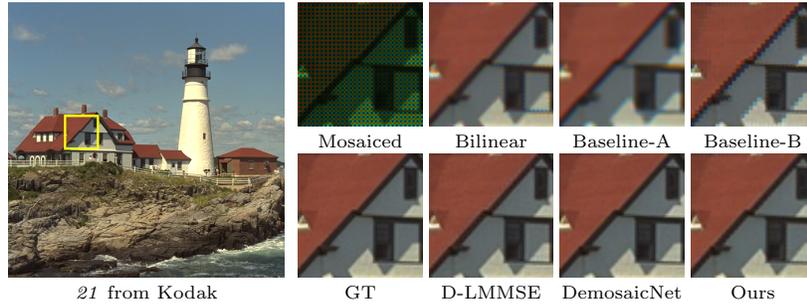
**The effectiveness of LUT-aware finetuning.** We compare SR-LUT and MuLUT-SDY-X2 with or without the LUT-aware finetuning strategy. We also report the corresponding network performance. As can be seen in Table 4, there is a performance drop from network predictions to LUT results, especially for the one with larger sampling intervals (3bit LUT). The proposed LUT-aware finetuning strategy is able to fill this gap consistently for different sampling intervals. Especially, after finetuning, a 3bit LUT achieves similar performance compared with a 4bit LUT, while taking  $10 \times$  less storage. Further, the proposed MuLUT-SDY-X2 also benefits from the finetuning strategy, showing its effectiveness and universality.

## 5.5 Results in Image Demosaicing

For the task of image demosaicing, we train baseline methods and MuLUT on the synthetic data pairs from the DIV2K dataset [1], where the mosaiced images are simulated by applying color masks on the original images. We compare our method with bilinear interpolation, classical solutions (D-LMMSE [66], AHD [18], and LSLCD [22]), and DNN (DemosaicNet [15]) on the widely-used Kodak dataset [32] and McMaster dataset [67]. As illustrated in Fig. 7a and Fig. 7b, MuLUT improves the performance of SR-LUT baselines by a large margin, *e.g.*, over 6.0dB on the Kodak dataset, achieving a better performance and efficiency tradeoff. Besides, as shown in Fig. 8, the results of Baseline-A are blurry, and Baseline-B produces noticeable blocking artifacts due to limited RF, while MuLUT obtains comparable visual quality with computation-heavy classical solutions and DNN.



**Fig. 7:** MuLUT achieves a better performance and efficiency tradeoff, compared with interpolation, classical solutions, single-LUT baselines, and DNN methods. The cPSNR values are average over 3 color channels. The runtimes of classical solutions are from DemosaicNet [15], where they are tested on a desktop computer. The runtimes of other methods are measured on a Xiaomi 11 smartphone.



**Fig. 8:** Visual comparison for image demosaicing on the Kodak dataset.

## 6 Conclusion Remarks

In this work, we propose MuLUT to generalize the SR-LUT by enabling the cooperation of multiple LUTs. Our method overcomes the limitation of the receptive field of a single LUT, empowering LUTs to be constructed like a neural network. Extensive experiments on both image super-resolution and image demosaicing demonstrate that MuLUT achieves significant improvement in restoration performance over SR-LUT while preserving its efficiency. Overall, the proposed MuLUT shows its versatility to serve as a universal caching framework and an efficient solution to avoid deploying heavy DNNs on edge devices.

## Acknowledgments

We acknowledge funding from National Key R&D Program of China under Grant 2017YFA0700800, and National Natural Science Foundation of China under Grants 62131003 and 62021001.

## References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1122–1131 (2017)
2. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: European Conference on Computer Vision (ECCV). vol. 11214, pp. 256–272 (2018)
3. Buades, A., Coll, B., Morel, J., Sbert, C.: Self-similarity driven color demosaicking. IEEE Trans. Image Process. **18**(6), 1192–1202 (2009)
4. Chang, H., Yeung, D., Xiong, Y.: Super-resolution through neighbor embedding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 275–282 (2004)
5. Chang, K., Ding, P.L.K., Li, B.: Color image demosaicking using inter-channel correlation and nonlocal self-similarity. Signal Process. Image Commun. **39**, 264–279 (2015)
6. Chen, C., Xiong, Z., Tian, X., Zha, Z., Wu, F.: Camera lens super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1652–1660 (2019)
7. Chen, H., Wang, Y., Xu, C., Shi, B., Xu, C., Tian, Q., Xu, C.: Addernet: Do we really need multiplications in deep learning? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1465–1474 (2020)
8. Cheng, Z., Xiong, Z., Chen, C., Liu, D., Zha, Z.: Light field super-resolution with zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10010–10019 (2021)
9. Chu, X., Zhang, B., Ma, H., Xu, R., Li, Q.: Fast, accurate and lightweight super-resolution with neural architecture search. In: International Conference on Pattern Recognition (ICPR). pp. 59–64 (2020)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision (ECCV). vol. 8692, pp. 184–199 (2014)
11. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision (ECCV). vol. 9906, pp. 391–407 (2016)
12. Duran, J., Buades, A.: Self-similarity and spectral correlation adaptive algorithm for color demosaicking. IEEE Trans. Image Process. **23**(9), 4031–4040 (2014)
13. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications **22**(2), 56–65 (2002)
14. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. Int. J. Comput. Vis. **40**(1), 25–47 (2000)
15. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics **35**(6), 191:1–191:12 (2016)
16. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: IEEE International Conference on Computer Vision (ICCV). pp. 349–356 (2009)
17. Gu, J., Dong, C.: Interpreting super-resolution networks with local attribution maps. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9199–9208 (2021)
18. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. IEEE Trans. Image Process. **14**(3), 360–369 (2005)

19. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269 (2017)
20. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5197–5206 (2015)
21. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 723–731 (2018)
22. Jeon, G., Dubois, E.: Demosaicking of noisy bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Trans. Image Process.* **22**(1), 146–156 (2013)
23. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 691–700 (2021)
24. Kasson, J.M., Nin, S.I., Plouffe, W., Hafner, J.L.: Performing color space conversions with three-dimensional linear interpolation. *J. Electronic Imaging* **4**(3), 226–250 (1995)
25. Keys, R.: Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **29**, 1153–1160 (1981)
26. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1646–1654 (2016)
27. Kim, S.J., Lin, H.T., Lu, Z., Süsstrunk, S., Lin, S., Brown, M.S.: A new in-camera imaging model for color computer vision and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(12), 2289–2302 (2012)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
29. Kokkinos, F., Lefkimiatis, S.: Iterative joint image demosaicking and denoising using a residual denoising network. *IEEE Trans. Image Process.* **28**(8), 4177–4188 (2019)
30. Lee, R., Dudziak, L., Abdelfattah, M.S., Venieris, S.I., Kim, H., Wen, H., Lane, N.D.: Journey towards tiny perceptual super-resolution. In: European Conference on Computer Vision (ECCV). vol. 12371, pp. 85–102 (2020)
31. Li, H., Yan, C., Lin, S., Zheng, X., Zhang, B., Yang, F., Ji, R.: PAMS: quantized super-resolution via parameterized max scale. In: European Conference on Computer Vision (ECCV). vol. 12370, pp. 564–580 (2020)
32. Li, X., Gunturk, B.K., Zhang, L.: Image demosaicing: a systematic survey. In: *Electronic Imaging* (2008)
33. Li, Y., Gu, S., Zhang, K., Gool, L.V., Timofte, R.: DHP: differentiable meta pruning via hypernetworks. In: European Conference on Computer Vision (ECCV). vol. 12353, pp. 608–624. Springer (2020)
34. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1132–1140 (2017)
35. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) (2017)
36. Luo, X., Xie, Y., Zhang, Y., Qu, Y., Li, C., Fu, Y.: Latticenet: Towards lightweight image super-resolution with lattice block. In: European Conference on Computer Vision (ECCV). vol. 12367, pp. 272–289 (2020)

37. Mantiuk, R., Daly, S.J., Kerofsky, L.: Display adaptive tone mapping. *ACM Trans. Graph.* **27**(3), 68 (2008)
38. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *IEEE International Conference on Computer Vision (ICCV)*. pp. 416–425 (2001)
39. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. *Multim. Tools Appl.* **76**(20), 21811–21838 (2017)
40. Mukherjee, J., Mitra, S.K.: Enhancement of color images by scaling the DCT coefficients. *IEEE Trans. Image Process.* **17**(10), 1783–1794 (2008)
41. Pan, Z., Li, B., He, D., Yao, M., Wu, W., Lin, T., Li, X., Ding, E.: Towards bidirectional arbitrary image rescaling: Joint optimization and cycle idempotence. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 17389–17398 (2022)
42. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019)
43. Qiu, G.: Interresolution look-up table for improved spatial magnification of image. *J. Vis. Commun. Image Represent.* **11**(4), 360–373 (2000)
44. Romano, Y., Isidoro, J., Milanfar, P.: RAISR: rapid and accurate image super resolution. *IEEE Trans. Computational Imaging* **3**(1), 110–125 (2017)
45. Schuler, S., Leistner, C., Bischof, H.: Fast and accurate image upscaling with super-resolution forests. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 3791–3799 (2015)
46. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1874–1883 (2016)
47. Song, D., Wang, Y., Chen, H., Xu, C., Xu, C., Tao, D.: Addersr: Towards energy efficient image super-resolution. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 15648–15657 (2021)
48. Song, D., Xu, C., Jia, X., Chen, Y., Xu, C., Wang, Y.: Efficient residual dense block search for image super-resolution. In: *Conference on Artificial Intelligence (AAAI)*. pp. 12007–12014 (2020)
49. Song, Q., Xiong, R., Liu, D., Xiong, Z., Wu, F., Gao, W.: Fast image super-resolution via local adaptive gradient field sharpening transform. *IEEE Trans. Image Process.* **27**(4), 1966–1980 (2018)
50. Timofte, R., Smet, V.D., Gool, L.V.: Anchored neighborhood regression for fast example-based super-resolution. In: *IEEE International Conference on Computer Vision (ICCV)*. pp. 1920–1927 (2013)
51. Timofte, R., Smet, V.D., Gool, L.V.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: *Asian Conference on Computer Vision (ACCV)*. vol. 9006, pp. 111–126 (2014)
52. Wang, L., Wu, H., Pan, C.: Fast image upsampling via the displacement field. *IEEE Trans. Image Process.* **23**(12), 5123–5135 (2014)
53. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.C.: ESRGAN: enhanced super-resolution generative adversarial networks. In: *European Conference on Computer Vision Workshops (ECCVW)*. vol. 11133, pp. 63–79 (2018)

54. Wang, Y.: A multilayer neural network for image demosaicking. In: IEEE International Conference on Image Processing (ICIP). pp. 1852–1856 (2014)
55. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
56. Xiao, Z., Fu, X., Huang, J., Cheng, Z., Xiong, Z.: Space-time distillation for video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2113–2122 (2021)
57. Xiao, Z., Xiong, Z., Fu, X., Liu, D., Zha, Z.: Space-time video super-resolution using temporal profiles. In: ACM Multimedia Conference. pp. 664–672 (2020)
58. Xin, J., Wang, N., Jiang, X., Li, J., Huang, H., Gao, X.: Binarized neural network for single image super resolution. In: European Conference on Computer Vision (ECCV). vol. 12349, pp. 91–107 (2020)
59. Xiong, Z., Sun, X., Wu, F.: Robust web image/video super-resolution. *IEEE Trans. Image Process.* **19**(8), 2017–2028 (2010)
60. Xiong, Z., Xu, D., Sun, X., Wu, F.: Example-based super-resolution with soft information and decision. *IEEE Trans. Multim.* **15**(6), 1458–1465 (2013)
61. Xu, R., Xiao, Z., Yao, M., Zhang, Y., Xiong, Z.: Stereo video super-resolution via exploiting view-temporal correlations. In: ACM Multimedia Conference. pp. 460–468 (2021)
62. Yang, J., Lin, Z., Cohen, S.: Fast image super-resolution based on in-place example regression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1059–1066 (2013)
63. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **19**(11), 2861–2873 (2010)
64. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces - 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers. vol. 6920, pp. 711–730 (2010)
65. Zhang, H., Liu, D., Xiong, Z.: Two-stream action recognition-oriented video super-resolution. In: IEEE International Conference on Computer Vision (ICCV). pp. 8798–8807 (2019)
66. Zhang, L., Wu, X.: Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Trans. Image Process.* **14**(12), 2167–2178 (2005)
67. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electronic Imaging* **20**(2), 023016 (2011)
68. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: European Conference on Computer Vision (ECCV). vol. 11211, pp. 294–310 (2018)
69. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2472–2481 (2018)