Learning Series-Parallel Lookup Tables for Efficient Image Super-Resolution

Cheng Ma[1,2,*], Jingyi Zhang[1,2,*], Jie Zhou[1,2], and Jiwen Lu[1,2,†]

[1] Beijing National Research Center for Information Science and Technology, China
[2] Department of Automation, Tsinghua University, China
macheng17@tsinghua.org.cn; zhangjy20@mails.tsinghua.edu.cn
{jzhou, lujiwen}@tsinghua.edu.cn

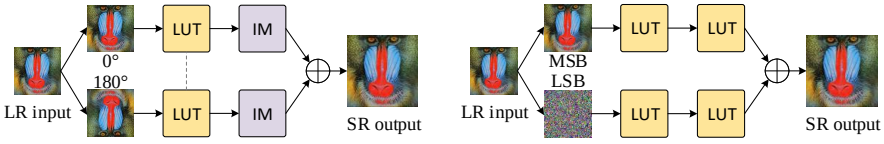
Abstract. Lookup table (LUT) has shown its efficacy in low-level vision tasks due to the valuable characteristics of low computational cost and hardware independence. However, recent attempts to address the problem of single image super-resolution (SISR) with lookup tables are highly constrained by the small receptive field size. Besides, their frameworks of single-layer lookup tables limit the extension and generalization capacities of the model. In this paper, we propose a framework of series-parallel lookup tables (SPLUT) to alleviate the above issues and achieve efficient image super-resolution. On the one hand, we cascade multiple lookup tables to enlarge the receptive field of each extracted feature vector. On the other hand, we propose a parallel network which includes two branches of cascaded lookup tables which process different components of the input low-resolution images. By doing so, the two branches collaborate with each other and compensate for the precision loss of discretizing input pixels when establishing lookup tables. Compared to previous lookup table-based methods, our framework has stronger representation abilities with more flexible architectures. Furthermore, we no longer need interpolation methods which introduce redundant computations so that our method can achieve faster inference speed. Extensive experimental results on five popular benchmark datasets show that our method obtains superior SISR performance in a more efficient way. The code is available at https://github.com/zhjy2016/SPLUT.

Keywords: Image super-resolution, look-up table, series-parallel network.

1 Introduction

As a fundamental task of computer vision, single image super-resolution (SISR) has attracted lots of research interests and plays an important role in wide applications, such as video surveillance, satellite imaging and high-definition televisions. SISR targets at recovering low-resolution (LR) images to high-resolution

* Equal contribution.
† Corresponding author.



(a) SR-LUT [11] (b) SPLUT (Ours)

Fig. 1. Comparison of SR-LUT and our SPLUT method. The former utilizes rotational ensemble to enlarge receptive fields from 2×2 to 3×3 and implement interpolation methods (IM) to improve recovery accuracy. The dashed line means weight sharing. For the sake of simplicity, we omit the rotations of 90° and 270°. In contrast, we stack multiple LUTs to significantly improve the receptive field size and design a new parallel architecture to compensate for the precision loss of discretizing input pixels.

(HR) ones by inferring high-frequency details. Along with the rapid development of deep learning techniques, various elaborately designed frameworks [18,19,43] based on convolutional neural networks (CNNs) have achieved encouraging progress in SISR. Most of these frameworks contain a large number of parameters and are time-consuming during testing. While several methods [17,21] have been proposed to reduce computation costs, they still rely on specific high-performance computing units, for example, GPUs and CPUs. Developing practical and real-time algorithms have been a growing trend in the SISR field.

Approaches [40,33,35,11] based on lookup tables (LUTs) have emerged in low-level vision tasks, including image enhancement and image super-resolution. These methods employ LUTs to establish the mapping relation between input pixels and the desired output pixels. In the testing phase, only a small number of parameters need storing and the inference processes are liberated from heavy computational burdens by replacing time-consuming calculations with fast memory accesses. As a result, the practicality of this kind of algorithm is significantly improved on mobile devices.

However, most existing LUT-based approaches only have a single layer of LUTs, which brings some major constraints. If n-dimensional LUTs (nD LUTs) are utilized and the n input entities for query all have v possible values, then the LUT scale is of v^n, where v and n are the two pivotal factors. While increasing the value of v and n may improve the restoration accuracy, a moderate improvement may lead to a rapid increase of the LUT scale. Thus, v and n are usually set to small values to avoid unbearably large LUTs, which severely limits the further enhancement of recovery abilities. In fact, receptive field (RF) size is a vital factor for deep learning and super-resolution. SISR is a well-known ill-posed problem since the same LR input may correspond to various high-resolution outputs with subtle differences. When we infer the missing details, a large context area on the input image should be considered to accurately capture the semantics and structures. In this way, we can effectively reduce the ambiguity of the estimated results. Therefore, how to enlarge receptive fields without exponentially increasing the storage and computation costs of LUTs is still an open issue.

Besides, due to the design of single-layer LUTs and the limited LUT scale, the extensibility and recovery capacity of existing LUT-based methods are highly constrained. Thus, a more powerful and flexible scheme is desired in order to further improve the inference ability of LUT-based methods.

To mitigate the above issues, we propose to learn series-parallel lookup tables (SPLUT) for SISR, as shown in Fig. 1. We cascade multiple lookup tables so that the query of latter LUTs are based on the outputs of former LUTs. In this way, the receptive field of each feature vector is gradually increased and the final SR results can be determined by larger local patches with more clear context information. For establishing LUTs, existing methods usually discretize the input values for reducing v and apply interpolation algorithms to improve the inference accuracy. However, such operations can only be implemented for small receptive fields. In our framework with large receptive fields, interpolations are inapplicable due to the exponentially increasing computational costs. In order to compensate for the precision loss of the discretized input pixel values, we propose a new parallel network which contains two branches. The first one processes the 4 most significant bits (MSBs) of the original 8-bit pixel values while the second one processes the 4 least significant bits (LSBs). The two branches of cascaded LUTs form the framework of series-parallel lookup tables.

In each branch, we introduce three kinds of 4D LUTs whose input values for the query are from different dimensions. We further propose horizontal and vertical aggregation modules to enlarge the RF size of different dimensions. In the training procedure, we build a mapping module for each LUT and quantize the intermediate activation so that the mapping relationships of the inputs and outputs can be transferred to the corresponding LUTs. Different from previous methods [11], the whole inference procedure of our method only contains retrieval and addition operations without complex multiplications. Experimental results on benchmark datasets show that SPLUT can achieve better SR performance on the smartphone platform, which demonstrates the effectiveness and efficiency of our proposed method.

In summary, the contributions of this work are threefold:

1. To the best of our knowledge, we are the first to present cascaded LUTs for enlarging receptive fields in the SISR field.
2. We propose a new parallel network to compensate for the precision loss caused by the discretization when establishing LUTs with large RF sizes.
3. Quantitative and qualitative results show that our method can recover the missing details more precisely and efficiently. The comparison of different SPLUT models verifies the superior extensibility of our proposed method.

2 Related Work

Single Image Super-Resolution. Non-deep learning methods [6,41,30,31] and deep learning methods [43,2,25,18] have significantly promoted the development of SISR. While recent deep learning methods perform more encouragingly than non-deep learning methods, many of them have a deep neural architecture with

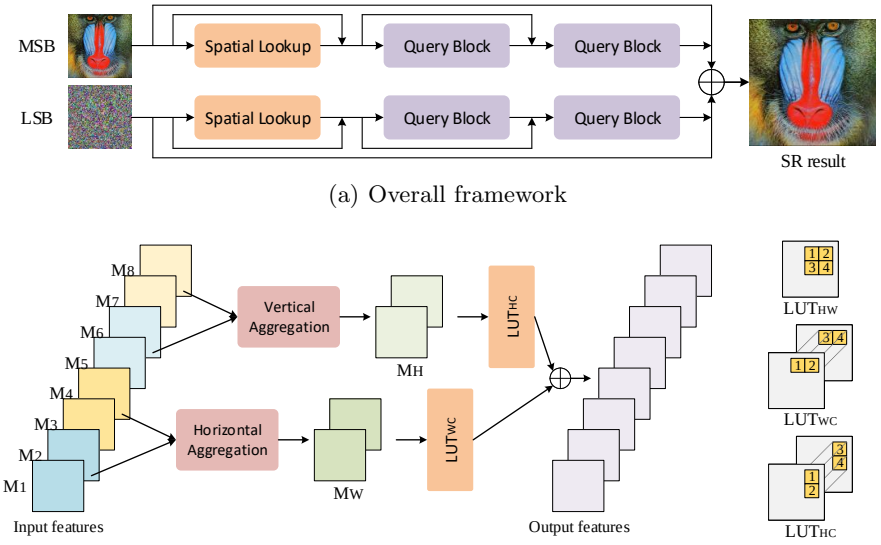
redundant parameters, which brings heavy computing costs and makes the training and inference rely on special computing devices. Therefore, efficient super-resolution has been a prevalent research interest of the community. In this field, methods based on various techniques have been proposed to improve the efficiency of SR algorithms. Lee *et al.* [17] and Zhang *et al.* [42] take advantage of the idea of knowledge distillation to compress the original deep teacher models to small student models with strong representation abilities. Wang *et al.* [34] explore the sparsity in image super-resolution by learning sparse masks to identify important regions and unimportant regions in images. Mei *et al.* [24] propose a non-local sparse attention module to achieve efficient and robust long-range modeling. Xin *et al.* [38] develop a binary neural network for SR by proposing a bit-accumulation mechanism to improve the precision of the quantized model. Some other methods [21,29,20] accomplish efficient SR inference by designing compact neural architectures. Lee *et al.* [16] search for appropriate architectures for both the generator and the discriminator by a neural architecture search approach. However, most of these methods are still based on convolutional layers and thus lack practicability on mobile devices.

Lookup Table. Lookup tables (LUTs) replace complex computations by simple and fast retrieval operations so that the efficiency of algorithms can be significantly improved. LUTs are widely used in a number of applications, such as numerical computation [5,27], video coding [15,32], pedestrian detection [4], RGB-to-RGBW conversion [14], *etc.* Besides, LUT is a classic and prevalent pixel adjustment tool in camera imaging pipeline [40] and photo editing software since it can easily manipulate the appearance of an image, such as color, exposure, saturation, *etc.* Recently deep learning methods based on LUTs have also emerged in low-level vision tasks [40,33,35,11]. In the image enhancement field, Zeng *et al.* [40] first propose image-adaptive 3D lookup tables and achieve high-performance photo enhancement. On this basis, Wang *et al.* [35] consider spatial information and further propose learnable spatial-aware 3D lookup tables. Wang *et al.* [33] model local context cues and propose pixel-adaptive lookup table weights for portrait photo retouching. As for the super-resolution field, Jo *et al.* [11] has developed SR-LUT by establishing the correspondence of LR input patterns and HR output patterns. However, as mentioned above, the RF size and the extensibility of LUT-based methods are still limited.

3 Method

3.1 Network Architecture

Given an input LR image I^{LR}, our goal is to recover the missing details and yield the SR image I^{SR} which is as similar as possible to the HR image I^{HR}. As shown in Fig. 2 (a), we design a series-parallel lookup table (SPLUT) network which contains two parallel branches processing different components of I^{LR}. We treat the RGB channels equally and separate the original input pixels with 8-bit values into two maps, I_{MSB} with 4 most significant bits (MSBs) and I_{LSB} with 4 least significant bits (LSBs). The two parallel branches take I_{MSB}



(a) Overall framework

(b) Query block (c) LUTs

Fig. 2. Details of the proposed SPLUT method. (a) The overall framework of our method. The input LR images are split into I_{MSB} and I_{LSB}, which are fed into two parallel branches, respectively. Each branch includes cascaded LUTs to extend receptive fields. (b) We take SPLUT-M with $C_f = 8$ as the example and display the details of the proposed query block, which further enlarge the RF size by aggregation modules and different kinds of LUTs. LUT_{WC} and LUT_{HC} can also model the correlations between different channels. (c) Illustration of different LUTs: LUT_{HW}, LUT_{WC} and LUT_{HC}. Their input values for the query are in different dimensions.

and I_{LSB} as inputs, respectively. Then we merge the outputs of the two parallel branches to compensate for the loss of quantization when establishing LUTs. In this way, the super-resolution capacities can be significantly enhanced. In each branch, there is a spatial lookup block, query blocks and skip connections. The spatial lookup block and the query blocks increase the RF size of extracted features gradually. The query blocks include horizontal and vertical aggregation modules which enlarge the RF size by the width and height dimensions, respectively. During training, we replace each LUT with a mapping block which is built on convolutional layers. Then we establish LUTs according to the mapping relations of the inputs and outputs of these mapping blocks. During inference, we retrieve the outputs of each LUT according to the indices computed by the input patterns, which are defined as the combinations of the n input entities for the query.

The LUT scale is mainly influenced by three factors, the number of pixels for retrieval n, the number of possible pixel values v, and the length of output vectors c. Then the LUT size can be computed by $v^n \cdot c$. In previous work, n is usually not more than 4 to keep a small LUT scale. The original pixels with 256

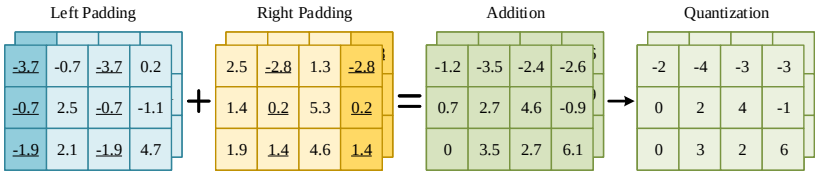


Fig. 3. Illustration of the proposed horizontal aggregation module. The underlined numbers in dark squares represent the results of the reflection padding operations. After adding the two padded feature maps together, the receptive field of the obtained features is enlarged on the width dimension.

different values are also discretized to obtain $v = 16$ or $v = 32$ bins for retrieval. The choice of c is determined by the practical tasks. For generating the output of $\times 4$ super-resolution, c is set to 16. The increase of n and v results in a significant increase in LUT scales while c only brings a linear growth of LUT scales. In our framework, we set $n = 4$ and $v = 16$ for all the LUTs so that $v^n = 65536$ is a relatively small constant. Different from previous methods which only contain a single layer of LUTs with a small RF size and lack the model extensibility, we cascade multiple LUTs to improve the RF size and flexibly control the trade-off between efficiency and accuracy by changing the network depth and the channel number of intermediate features C_f. In practice, we design three models, SPLUT-S, SPLUT-M and SPLUT-L with $C_f = 4$, $C_f = 8$ and $C_f = 16$, respectively.

Since n is set to 4, the indices for retrieving LUTs are computed by 4 adjacent entities. We design 3 kinds of LUTs whose input patterns are of different dimensions. For an intermediate feature, there are mainly three dimensions, W, H, and C, representing width, height, and channel, respectively. The 3 kinds of designed LUTs are LUT_{WH}, LUT_{HC} and LUT_{WC}, as depicted in Fig. 2 (c). The input pattern of LUT_{WH} is a 2×2 area in the spatial dimensions. The 2×2 input pattern of LUT_{HC} is along the height and channel dimensions while that of LUT_{WC} is along the width and channel dimensions. These LUTs can capture local dependency and enlarge receptive fields of different dimensions. In our framework, they are placed in different modules for specific functions. Next, we take the model SPLUT-M as an example to describe the details of each component of our framework.

Spatial LUT Block. The two branches of I_{MSB} and I_{LSB} have similar architectures. In the beginning, spatial correlations are more important than channel correlations. Hence, we use a spatial LUT block to exploit the spatial dependency of neighboring pixels in the input images. In this block, we employ reflection padding to keep the spatial dimensions unchanged after retrievals.

Query Blocks. Following the spatial LUT block, there are two query blocks. The details of the query block are shown in Fig. 2 (b). For the model of SPLUT-M, C_f is set to 8 and thus we have 8 intermediate feature maps, named $M_1, ..., M_8$. We split them into 4 groups, each adjacent two in one group. The first two groups are fed into the horizontal aggregation module while the last two are fed into the

vertical aggregation module. The two modules enlarge the RF size by the width dimension and the height dimension, respectively. M_W and M_H are obtained by the two modules and they both have 2 channels. Since only exploring spatial information severely affects the representation ability of the network, we use $\mathrm{LUT}_{\mathrm{WC}}$ and $\mathrm{LUT}_{\mathrm{HC}}$ to model the correlations between different channels. We retrieve the output of a width-channel $\mathrm{LUT}_{\mathrm{WC}}$ by computing the query indices according to the input patterns on M_W. Similarly, we retrieve $\mathrm{LUT}_{\mathrm{HC}}$ according to M_H. Finally, the outputs of two kinds of LUTs are added to get the output of the query block.

Aggregation Modules. Here we describe the details of the horizontal and vertical aggregation modules. As shown in Fig. 3, we take the horizontal aggregation module as an example. The two input feature maps both have two channels and have the same receptive fields. First, we pad one feature map on the left by reflection padding and pad the other one on the right. Then we obtain two feature maps whose receptive fields have a shift of one pixel along the width dimension. After merging the two feature maps by addition, the obtained feature map has a larger spatial receptive field. In order to transfer the real-value responses to the query indices for the following LUTs, we need to quantize the real values to form $v = 16$ discrete values. Specifically, we set the quantization interval to 1 so that the quantization can be achieved by a simple rounding operation. By doing this, we avoid complex multiplication computations and improve the efficiency of the proposed module. The operations are similar for the vertical aggregation module. Differently, vertical aggregation enlarges the receptive field along the height dimension.

Parallel Branches. In prior arts [40,11,35,33], pixel values are quantized for reducing the possible values and decreasing LUT scales. However, the original continuously changing pixels become discrete, which may cause blocking effects in the SR results. Therefore, interpolation algorithms [12] are usually applied to smooth the output textures. However, they introduce additional multiplications and comparison operations. Moreover, these algorithms are only available for LUTs with a small RF size. If we use r to represent the RF size of a feature, then 2^r nearest bounding vertices need considering for interpolating the retrieval results. In our framework with a large RF size, such a computation complexity is unacceptable. We propose a new parallel framework to alleviate this issue. The framework includes two branches with the same architecture of cascaded LUTs. One branch processes I_{MSB} and mainly focuses on capturing context semantic information. The other branch processes I_{LSB} and provides high-frequency details. By Merging the outputs of the two branches, we are able to compensate for the loss of quantization when establishing LUTs and hence boost SR performance.

Skip Connections. In order to improve the representation abilities of the network, we store low-precision real numbers in LUTs. Since the above mentioned operations of quantization and index computation sacrifice the precision of intermediate features, we introduce skip connections to fuse the real-value inputs and the retrieval outputs to improve the precision. Besides, identity map-

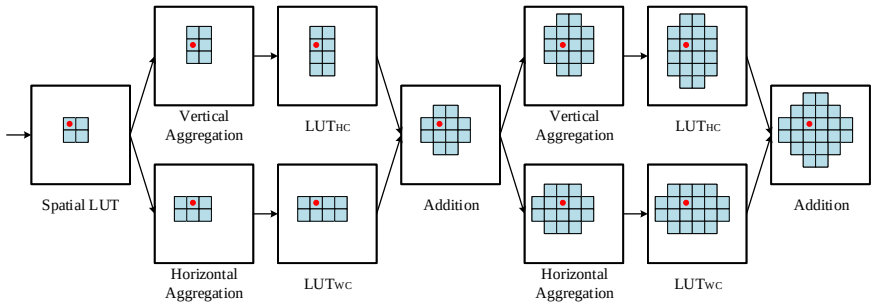


Fig. 4. The visualization of receptive fields with respect to the feature marked in red after each module or operation. Horizontal aggregation modules and $\mathrm{LUT_{WC}}$ increase the RF size by the width dimension while vertical aggregation modules and $\mathrm{LUT_{HC}}$ increase the RF size by the height dimension. The addition operations further enlarge receptive fields by fusing the two different areas of receptive fields.

ping [8] is a pivotal component of SR networks. Thus we adopt a skip connection between the input image and the output of the last query block to simplify optimization and enhance recovery accuracy.

3.2 Training Strategy

For training the network, we replace the LUTs by mapping modules, which are comprised of a convolutional layer with a kernel size of $k_h \times k_w$, GELU [9] layers, and 1×1 convolutional layers. All mapping modules output feature maps with C_f channels except the last one. The last mapping module outputs $C_{sr} = s^2$ channels where s is the upscaling factor. A pixel-shuffle layer [28] maps the outputs with 16 channels to the final results for $\times 4$ SR. In mapping modules of different LUTs, k_h and k_w are different. For the spatial LUT block, the input channel number is 1 and $k_h = k_w = 2$. For $\mathrm{LUT_{HC}}$ and $\mathrm{LUT_{WC}}$, the input channel number is 2. $k_h = 2$ and $k_w = 1$ for $\mathrm{LUT_{HC}}$ while $k_h = 1$ and $k_w = 2$ for $\mathrm{LUT_{WC}}$. The consecutive 1×1 convolutions followed by GELU layers strengthen the nonlinearity and representative abilities of the mapping modules. We jointly train the MSB and LSB branches by imposing Mean Squared Error (MSE) loss on the final SR outputs. For the quantized activations, we use the identity straight-through estimator (STE) [39] to achieve end-to-end back-propagation.

3.3 Analysis of SPLUT

Receptive Field Size. We visualize the changes of RF sizes through the whole framework in Fig. 4. After the first spatial LUT block, each feature has an RF size of 2×2. Then in the first query block, the horizontal aggregation module and $\mathrm{LUT_{WC}}$ enlarge the RF size along the width dimension. The vertical aggregation module and $\mathrm{LUT_{HC}}$ enlarge the RF size along the height dimension. After fusing

the two outputs by addition, we get an RF size of 12. In the second query block, we implement similar operations and further increase the RF size. Finally, we get an RF size of 24, which is much bigger than the RF size of $3 \times 3 = 9$ in SR-LUT [11] by rotational ensemble trick.

Computational Cost. In SR-LUT [11], rotational ensemble is proposed to extend the RF size of 2×2 to 3×3. The computational burdens of SR-LUT mainly include image rotation, retrieval of 4 input images with different orientations, and interpolation methods which contain heavy multiplication and comparison operations. While our SPLUT model has more LUTs, we do not need the rotation and interpolation operations. Therefore, our method can achieve a faster inference speed than SR-LUT.

4 Experiments

4.1 Implementation Details

Datasets and Metrics. We train the proposed serial-parallel lookup table (SPLUT) model on the DIV2K dataset [1] and evaluate the effectiveness of our method on 5 widely used benchmarks: Set5 [3], Set14 [41],BSD100 [22], Urban100 [10] and Manga109 [23]. We focus on the upscaling factor of $\times 4$ in our experiments. We use Peak Signal-to-Noise Ratio (PSNR) and structural similarity index (SSIM) [37] as the evaluation metrics for prediction accuracy. To compare the computation efficiency, we measure and report the runtime of super-resolving 320×180 LR images on mobile phones.

Training Setting. We design three SPLUT models with different model sizes, namely SPLUT-S, SPLUT-M, and SPLUT-L. The three models have the same architecture depicted in Fig. 2 (a). The difference between the three models lies in the value of C_f, the number of lookup tables per query block and the grouping strategy of input feature maps. We have introduced the details of SPLUT-M. The details of the other two models are described in the supplementary material. We train SPLUT models with PyTorch [26] on Nvidia 2080Ti GPUs. We use Adam Optimizer [13] with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$ to jointly train the MSB and LSB branches. The learning rate is set to 10^{-3}. We randomly crop LR images into 48×48 patches with a mini-batch size of 32. We enhance the dataset by randomly rotating and flipping.

4.2 Results and Analyses

Quantitative Comparison. We compare our method with SR methods based on sparse coding which include NE+LLE [6], Zeyde *et al.* [41], ANR [30] and A+ [31], SR methods based on deep learning including CARN-M [2], FMEN [7] and RRDB [36], and SR method based on LUTs, SR-LUT [11]. Since the source code of SR-LUT [11] is not released, we reproduce the SR-LUT algorithm and compare our method with it under the same environment. Since the implementation of sparse coding based methods [6,41,30,31] rely on Matlab, we evaluate

Table 1. Quantitative comparisons of different SR methods on 5 benchmark datasets. The best results among LUT-based methods are **highlighted**. Running time is measured by super-resolving 320 × 180 LR images on the mobile phone. * represents the running time is measured on computer CPUs. Size denotes the storage space or the parameter number of each model.

Method	Time	Size	Set5		Set14		BSDS100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NE+LLE	7016ms*	1.434MB	29.62	0.840	26.82	0.735	26.49	0.697	23.84	0.694	26.10	0.820
Zeyde *et al.*	8797ms*	1.434MB	26.69	0.843	26.90	0.735	26.53	0.697	23.90	0.696	26.24	0.824
ANR	1715ms*	1.434MB	29.70	0.842	26.86	0.737	26.52	0.699	23.89	0.696	26.18	0.821
A+	1748ms*	15.17MB	30.27	0.860	27.30	0.750	26.73	0.709	24.33	0.719	26.91	0.848
CARN-M	4955ms	1.593MB	31.82	0.890	28.29	0.775	27.42	0.730	25.62	0.769	29.85	0.899
FMEN	3101ms	1.395MB	32.24	0.896	28.70	0.784	27.63	0.738	26.28	0.791	30.70	0.911
RRDB	31717ms	63.83MB	32.60	0.900	28.88	0.790	27.76	0.743	26.73	0.807	31.16	0.916
SR-LUT	279ms	**1.274M**	29.82	0.848	27.01	0.736	26.53	0.695	24.02	0.699	26.80	0.838
SPLUT-S	**242ms**	5.5M	30.01	0.852	27.20	0.743	26.68	0.702	24.13	0.706	27.00	0.843
SPLUT-M	265ms	7M	30.23	0.857	27.32	0.746	26.74	0.704	24.21	0.709	27.20	0.848
SPLUT-L	545ms	18M	**30.52**	**0.863**	**27.54**	**0.752**	**26.87**	**0.709**	**24.46**	**0.719**	**27.70**	**0.858**

these methods on the CPUs of computers which may be faster than mobile phones. The quantitative comparisons are shown in Table 1. As observed, our SPLUT models achieve much faster inference than both sparse coding based methods and deep learning based methods. SPLUT-S, SPLUT-M and SPLUT-L all obtain higher PSNR and SSIM than NE+LLE, Zeyde *et al.*and ANR. The SPLUT-M is comparable to A+ and SPLUT-L is superior to A+ on all benchmarks. While deep learning based methods have the best PSNR and SSIM performance, their inference speed is much slower than our method. As a method based on LUTs, SR-LUT is much faster than the other compared methods. However, it is still slower than our SPLUT-S and SPLUT-M methods. Besides, SPLUT models all outperform SR-LUT by a large margin on PSNR and SSIM metrics. Our SPLUT-M model achieves a better trade-off between efficiency and accuracy. Compared to SR-LUT, SPLUT-M improves PSNR by 0.4 dB on Set5 and Manga109 in a faster speed. By comparing model sizes, we see that our SPLUT method only brings a linear increase in storage costs. We believe the LUT size of our method is acceptable for current mobile phones. Therefore, we regard the runtime as a more important factor for evaluating efficiency. Besides, SPLUT-L presents more powerful SR abilities than SPLUT-S and SPLUT-M. These comparisons verify that our SPLUT framework is more powerful and more flexible than the previous framework of single-layer LUTs whose scale increases exponentially with RF sizes.

Qualitative Comparison. Fig. 5 illustrates the qualitative comparisons of Bicubic interpolation, A+, SR-LUT, our SPLUT models, and ground-truth images. We can see SR-LUT fails to present natural details for sharp edges. In

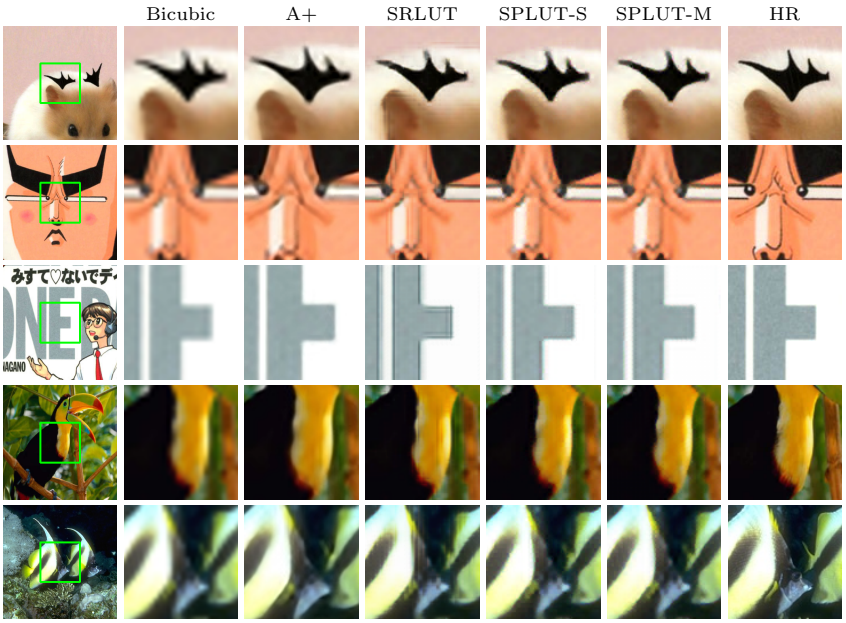


Fig. 5. Qualitative Comparisons of bicubic interpolation, A+ [31], SRLUT [11], our SPLUT method and HR images. The results show our method can generate sharp edges without severe artifacts.

some areas with continuously changing colors, there are often blocking artifacts. In the third row, the SR results of SR-LUT have severe ringing artifacts near edges. While A+ introduces fewer artifacts than SR-LUT, it may generate more blurry edges, as shown in the second row. On the contrary, our SPLUT models restore more natural textures. It can be seen that the expansion of the receptive field in SPLUT helps the network grasp the texture and structure information of context regions to achieve better reconstruction accuracy.

Ablation: Parallel Network vs. Interpolation. We take SPLUT-M as the baseline model and further investigate the effectiveness of our proposed parallel network by comparing it with interpolation algorithms. Specifically, we remain only one branch of the SPLUT model and use full-precision LR images as inputs to train this model. In the inference phase, we follow SR-LUT [11] to extract I_{MSB} for retrieval and store I_{LSB} for interpolation. We call the one-branch model without interpolation "OBM w/o interpolation". Since the cascaded LUTs in this model brings a large RF size of r, it is intractable to consider all bounding vertices and simply implement interpolations due to the computational complexity of 2^r. To improve the SR accuracy of this model, we design two interpolation methods for the input images. For a position of (x, y), SR-LUT implements 4-simplex interpolation for 4D LUTs by exploring the relation of $I_{LSB}^{(x,y)}$, $I_{LSB}^{(x+1,y)}$, $I_{LSB}^{(x,y+1)}$ and $I_{LSB}^{(x+1,y+1)}$. A weighted sum of the retrieval results for the 16 bound-

Table 2. Comparison of parallel network and interpolation methods. The results show OBM w/o interpolation and the models with interpolation methods cannot achieve comparable performance to our SPLUT method.

method	Size	Set5		Set14	
		PSNR	SSIM	PSNR	SSIM
OBM w/o interpolation	3.5M	27.24	0.8217	25.30	0.7175
Tail-layer Interpolation	3.5M	27.24	0.8217	25.30	0.7175
First-layer Interpolation	3.5M	27.24	0.8218	25.30	0.7176
SPLUT	7M	**30.23**	**0.8567**	**27.32**	**0.7460**

Table 3. Comparison of SPLUT models with different quantization precision v_f of intermediate features. We choose $v_f = 16$ as the optimal setting considering accuracy and efficiency.

v_f	Size	Set5		Set14		BSDS100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
8	1.375M	29.95	0.849	27.13	0.738	26.63	0.698	24.05	0.702	26.83	0.839
12	2.898M	30.11	0.854	27.26	0.743	26.71	0.702	24.15	0.706	27.06	0.844
16	7M	30.23	0.857	27.32	0.746	26.74	0.704	24.21	0.709	27.20	0.848
20	15.648M	30.22	0.856	27.31	0.746	26.74	0.704	24.22	0.710	27.25	0.848
24	31.375M	30.25	0.858	27.34	0.747	26.75	0.705	24.24	0.711	27.24	0.849

ing vertices is computed as the final output since SR-LUT has only one layer of LUT. In our methods, we also utilize I_{LSB} for interpolation but we cannot get the final SR output by directly fusing the retrieval results of the first layer of spatial LUT since we still have other following LUTs for retrieval. Therefore, we concatenate the 16 bounding vertices of all input pixels to form 16 index maps, which reduce the complexity of 2^r to 2^4. In our first interpolation method, we take these 16 index maps as the inputs to the spatial lookup blocks and get 16 SR results through the whole network. We interpolate the 16 SR results by I_{LSB} using the 4-simplex method. We call this method tail-layer interpolation. In our second method, we feed the 16 index maps to the spatial lookup blocks and get 16 intermediate feature maps. We fuse the 16 feature maps by I_{LSB} to get one feature map. By feeding the feature map to the following layers, we can get a final SR output. We call this method first-layer interpolation. More implementation details are described in the supplementary. The results are shown in Table 2. Our SPLUT with the parallel network achieves an improvement of more than 2.9 dB over the two interpolation methods and OBM w/o interpolation. This proves that applying interpolation methods fails to compensate for the precision loss, which is caused by discretizing pixel values when establishing lookup tables. In contrast, our parallel network is superior to interpolation algorithms in compensating for the precision loss for large RF sizes. It can also be inferred that our parallel network is inherently robust to different receptive fields.

Table 4. Ablation study on LUT number. Extending the depth and width of the SPLUT network both boost the SR performance, which demonstrates the effectiveness and extensibility of the proposed method.

Layer num	Size	Set5		Set14		BSDS100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SPLUT_{1-2}	5M	29.77	0.846	27.04	0.737	26.56	0.696	23.95	0.697	26.68	0.836
SPLUT_{1-4}	10M	30.01	0.852	27.21	0.742	26.66	0.700	24.12	0.705	27.05	0.844
SPLUT_{1-2-2}	7M	30.23	0.857	27.32	0.746	26.74	0.704	24.21	0.709	27.20	0.848
SPLUT_{1-4-4}	18M	30.52	0.863	27.54	0.752	26.87	0.709	24.46	0.719	27.70	0.858

Ablation: Quantization Precision. In SPLUT, we uniformly quantize the real-value activations in the aggregation modules during training to control the size of LUTs. Since we set $n = 4$ for all the LUTs, the prediction accuracy and model scale are mainly determined by v. For the spatial lookup blocks, we fix the sampling interval to 16. We change the precision of quantizing the real-value intermediate features, v_f, and investigate the influence of it. Table 3 presents the performance of SPLUT-M with different v_f. The results indicate that increasing v_f constantly improves the SR accuracy but the model size also increases. When v_f is less than 16, the accuracy improves rapidly. However, SPLUT only has a minor improvement when v_f is greater than 16. Hence we choose $v_f = 16$ as the appropriate value which presents appealing SR performance with a relatively small model size. In practice, the quantization precision can be determined by the application scenarios to achieve the flexible model design.

Ablation: LUT Number. We conduct ablation studies on the number of LUTs in each parallel branch to further investigate the extensibility of our SPLUT architecture. As shown in Table 4, we compare 4 models with different model depths and widths. The model is named according to the number of LUTs in each block. SPLUT_{1-2} represents there are two layers of LUTs. The first layer is the spatial lookup block and the second layer is a query block which contains one LUT_{WC} and one LUT_{HC}. For SPLUT_{1-4}, the second layer contains two different LUT_{WC} and two different LUT_{HC}. In this model, the channel number of intermediate feature maps is $n_{in} = 16$. SPLUT_{1-2-2} and SPLUT_{1-4-4} have the similar architectures to SPLUT_{1-2} and SPLUT_{1-4} but have two query blocks in each branch. SPLUT_{1-2-2} and SPLUT_{1-4-4} are actually the same as SPLUT-M and SPLUT-L, respectively.

In Table 4, we observe a huge improvement when the network width increases by comparing the first two rows and the last two rows. This indicates that more LUTs per layer can extract more information by the limited number of channels. In this way, the model can obtain a stronger representation ability and better SR reconstruction performance. As the number of query blocks increases, we see SPLUT_{1-2-2} achieves an improvement of 0.46 dB over SPLUT_{1-2} while SPLUT_{1-4-4} outperforms SPLUT_{1-4} by about 0.51 dB on the PSNR performance of Set5. It is inferred that cascading multiple layers of LUTs is very

Table 5. Effects of of horizontal and vertical aggregation modules. After removing the aggregation modules, the RF size gets smaller and the restoration ability is degraded.

	Set5		Set14		BSDS100		Urban100		Manga109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SPLUT w/o AM	29.64	0.839	26.95	0.724	26.08	0.676	23.50	0.677	25.72	0.812
SPLUT	**30.23**	**0.857**	**27.32**	**0.746**	**26.74**	**0.704**	**24.21**	**0.709**	**27.20**	**0.848**

effective in enlarging the receptive fields and boosting recovery abilities. Comparing SPLUT_{1-2-2} and SPLUT_{1-4}, both models have 5 lookup tables. However, SPLUT_{1-2-2} gains a boost of about 0.22 dB. This indicates that the network depth is more important than network width in SPLUT. Besides, the comparisons demonstrate the effectiveness of enlarging receptive fields.

Ablation: Aggregation Modules. Table 5 shows the performance comparison between the original SPLUT model and the SPLUT model without aggregation modules, SPLUT w/o AM. From the table we see there is a gap of 0.59 dB between the PSNR performance of SPLUT w/o AM and SPLUT on Set5. The key insight is that we expand the receptive field of the intermediate features by fusing the feature maps padded in opposite directions in the aggregation modules. Thus the features become stronger for the subsequent lookup processes. When aggregation modules are removed, the overall receptive field of the final output is significantly reduced compared with that of the original network, leading to performance degradation correspondingly. The experimental results demonstrate the effectiveness of the proposed horizontal and vertical aggregation modules.

5 Conclusion

In this paper, we have proposed a series-parallel lookup table network to achieve efficient image super-resolution. On the one hand, we cascade multiple LUTs to enlarge the receptive field size progressively and enhance the representation capacity of the whole network. On the other hand, we design a parallel architecture to fuse the information of MSB inputs and LSB inputs. By doing so, we compensate for the precision loss caused by quantization when establishing LUTs and improve the prediction accuracy. Comprehensive experiments have demonstrated the effectiveness, efficiency, and flexibility of the proposed method.

Acknowledgement This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603 and Grant U1813218, in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPR. pp. 126–135 (2017) 9
2. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV. pp. 252–268 (2018) 3, 9
3. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012) 9
4. Bilal, M., Khan, A., Karim Khan, M.U., Kyung, C.M.: A low-complexity pedestrian detection framework for smart video surveillance systems. TCSVT **27**(10), 2260–2273 (2017) 4
5. Chang, C.C., Chou, J.S., Chen, T.S.: An efficient computation of euclidean distances using approximated look-up table. TCSVT **10**(4), 594–599 (2000) 4
6. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR. vol. 1, pp. I–I. IEEE (2004) 3, 9
7. Du, Z., Liu, D., Liu, J., Tang, J., Wu, G., Fu, L.: Fast and memory-efficient network towards efficient image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 853–862 (2022) 9
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 8
9. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016) 8
10. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. pp. 5197–5206 (2015) 9
11. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: CVPR. pp. 691–700 (2021) 2, 3, 4, 7, 9, 11
12. Kasson, J.M., Nin, S.I., Plouffe, W., Hafner, J.L.: Performing color space conversions with three-dimensional linear interpolation. Journal of Electronic Imaging **4**(3), 226–250 (1995) 7
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
14. Lee, C., Monga, V.: Power-constrained rgb-to-rgbw conversion for emissive displays: Optimization-based approaches. TCSVT **26**(10), 1821–1834 (2016) 4
15. Lee, J.Y., Lee, J.J., Park, S.: New lookup tables and searching algorithms for fast h.264/avc cavlc decoding. TCSVT **20**(7), 1007–1017 (2010) 4
16. Lee, R., Dudziak, Ł., Abdelfattah, M., Venieris, S.I., Kim, H., Wen, H., Lane, N.D.: Journey towards tiny perceptual super-resolution. In: ECCV. pp. 85–102. Springer (2020) 4
17. Lee, W., Lee, J., Kim, D., Ham, B.: Learning with privileged information for efficient image super-resolution. In: ECCV. pp. 465–482. Springer (2020) 2, 4
18. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: CVPR. pp. 3867–3876 (2019) 2, 3
19. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. pp. 136–144 (2017) 2
20. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. In: ECCV. pp. 41–55. Springer (2020) 4
21. Luo, X., Xie, Y., Zhang, Y., Qu, Y., Li, C., Fu, Y.: Latticenet: Towards lightweight image super-resolution with lattice block. In: ECCV. pp. 272–289. Springer (2020) 2, 4

22. 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: ICCV. pp. 416–425 (2001) 9
23. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications **76**(20), 21811–21838 (2017) 9
24. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: CVPR. pp. 3517–3526 (2021) 4
25. Park, S.J., Son, H., Cho, S., Hong, K.S., Lee, S.: Srfeat: Single image super-resolution with feature discrimination. In: ECCV. pp. 439–455 (2018) 3
26. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017) 9
27. Rizvi, S., Nasrabadi, N.: An efficient euclidean distance computation for vector quantization using a truncated look-up table. TCSVT **5**(4), 370–371 (1995) 4
28. Shi, W., Caballero, J., Huszár, 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: CVPR. pp. 1874–1883 (2016) 8
29. Song, D., Wang, Y., Chen, H., Xu, C., Xu, C., Tao, D.: Addersr: Towards energy efficient image super-resolution. In: CVPR. pp. 15648–15657 (2021) 4
30. Timofte, R., De Smet, V., Van Gool, L.: Anchored neighborhood regression for fast example-based super-resolution. In: ICCV. pp. 1920–1927 (2013) 3, 9
31. Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: ACCV. pp. 111–126. Springer (2014) 3, 9, 11
32. Tsang, S.H., Chan, Y.L., Siu, W.C.: Region-based weighted prediction for coding video with local brightness variations. TCSVT **23**(3), 549–561 (2013) 4
33. Wang, B., Lu, C., Yan, D., Zhao, Y.: Learning pixel-adaptive weights for portrait photo retouching. arXiv preprint arXiv:2112.03536 (2021) 2, 4, 7
34. Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., Guo, Y.: Exploring sparsity in image super-resolution for efficient inference. In: CVPR. pp. 4917–4926 (2021) 4
35. Wang, T., Li, Y., Peng, J., Ma, Y., Wang, X., Song, F., Yan, Y.: Real-time image enhancer via learnable spatial-aware 3d lookup tables. In: ICCV. pp. 2471–2480 (2021) 2, 4, 7
36. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCVW. pp. 0–0 (2018) 9
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004) 9
38. Xin, J., Wang, N., Jiang, X., Li, J., Huang, H., Gao, X.: Binarized neural network for single image super resolution. In: ECCV. pp. 91–107. Springer (2020) 4
39. Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., Xin, J.: Understanding straight-through estimator in training activation quantized neural nets. arXiv preprint arXiv:1903.05662 (2019) 8
40. Zeng, H., Cai, J., Li, L., Cao, Z., Zhang, L.: Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. TPAMI (2020) 2, 4, 7
41. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: ICCS. pp. 711–730. Springer (2010) 3, 9

42. Zhang, Y., Chen, H., Chen, X., Deng, Y., Xu, C., Wang, Y.: Data-free knowledge distillation for image super-resolution. In: CVPR. pp. 7852–7861 (2021) 4
43. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. pp. 2472–2481 (2018) 2, 3