AdderSR: Towards Energy Efficient Image Super-Resolution

Dehua Song[1*], Yunhe Wang[1*], Hanting Chen[1,2], Chang Xu[3], Chunjing Xu[1], Dacheng Tao[3]

[1] Noah's Ark Lab, Huawei Technologies   [2] Peking University   [3] The University of Sydney.

{dehua.song, yunhe.wang}@huawei.com;   dacheng.tao@sydney.edu.au;

Abstract

This paper studies the single image super-resolution problem using adder neural networks (AdderNets). Compared with convolutional neural networks, AdderNets utilize additions to calculate the output features thus avoid massive energy consumptions of conventional multiplications. However, it is very hard to directly inherit the existing success of AdderNets on large-scale image classification to the image super-resolution task due to the different calculation paradigm. Specifically, the adder operation cannot easily learn the identity mapping, which is essential for image processing tasks. In addition, the functionality of high-pass filters cannot be ensured by AdderNets. To this end, we thoroughly analyze the relationship between an adder operation and the identity mapping and insert shortcuts to enhance the performance of SR models using adder networks. Then, we develop a learnable power activation for adjusting the feature distribution and refining details. Experiments conducted on several benchmark models and datasets demonstrate that, our image super-resolution models using AdderNets can achieve comparable performance and visual quality to that of their CNN baselines with an about 2.5× reduction on the energy consumption. The codes are available at: https://github.com/huawei-noah/AdderNet.

1. Introduction

Single image super-resolution (SISR) is a typical computer vision task which aims at reconstructing a high-resolution (HR) image from a low-resolution (LR) image. SISR is a very popular image signal processing task in real-world applications such as smart phones and mobile cameras. Due to the hardware constrains of these portable devices, it is necessary to develop SISR models with low computation cost and high visual quality.

Recently, deep convolutional neural network (CNN) has dramatically boosted the performance of SISR. The first super-resolution convolutional neural network (SR-
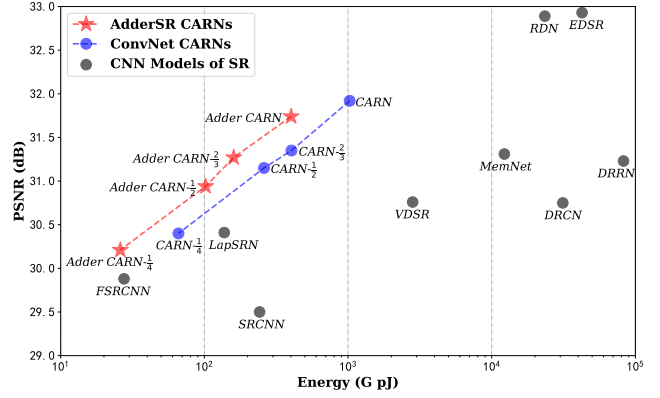
---

*Equal contribution

Figure 1. PSNR and energy trade-off with the SOTA SR methods on Urban100 database for ×2 scale. AdderSR networks achieve superior performance with moderate energy.

CNN) [4] contains only three convolutional layers with about 57K parameters. Then, the capacity of DCNN was amplified with the increasing of depth and width (channel number), resulting in notable improvement of super-resolution. The parameters and computation cost of recent DCNN are increased accordingly. For example, the residual dense network (RDN) [41] contains 22M parameters and requires about 10,192G FLOPs (floating-number operations) for processing only one image. Compared with neural networks for visual recognition (*e.g.*, ResNet-50 [11]), models for SISR have much higher computational complexities due to the larger feature map sizes. These massive calculations will consume much energy and reduce the enduration time of mobile devices.

In order to address the aforementioned problem, a series of approaches have been proposed to compress and accelerate deep convolutional neural networks. The prominent compression methods such as filter pruning [9, 35, 14, 22] and knowledge distillation [12, 7, 6, 38] reduce the computation by narrowing or shallowing the network. On the other hand, quantization methods [3, 23, 20, 37] devote to reducing the computation complexity of multiplications while preserving the architecture of the original neural network. Wherein, binarization is a specific case that weights and activations in networks are represented as $\{+1, -1\}$,

which can significantly reduce the energy and memory consumptions. However, the binarized network often cannot maintain the accuracy of full precision network, especially for super-resolution task [23]. Recently, Chen *et al*. [2] proposed a novel AdderNet which replaces the multiplication operations by additions. Since the complexity of additions is much lower than that of multiplications, this work motivates us to utilize AdderNet for constructing energy efficient SISR models.

To maximally excavate the potential for exploiting AdderNets to establish SISR models, we first analyze the theoretical difficulties for applying the additions into SISR tasks. Specifically, input and output features in any two neighbor layers in SISR models are very close with the similar global texture and color information as shown in Figure 2. However, the identify mapping cannot be learned by a one-layer adder network. Thus, we suggest to insert self-shortcuts and formulate new adder models for the SISR task. Moreover, we find that the high-pass filter is also hard to approximate by adder units. We then develop a learnable power activation. By exploiting these two techniques, we replace the conventional convolution filters in modern SISR networks by adder filters and establish AdderSR models accordingly. The effectiveness of the proposed SISR networks using additions is verified on several benchmark datasets. We can obtain comparable performance (*i.e.*, PSNR values and visual quality) using AdderSR models with that of the CNN baselines. Meanwhile, we can reduce more than 50% of the overall energy consumptions of these neural networks. Fig. 1 indicates the superior performance of AdderSR networks with moderate energy.

The rest of this paper is organized as follows. We briefly investigate the related works on neural network compression and energy-efficient approaches in Section 2. In Section 3, we present the motivation of using additions in SISR and establish AdderSR models. Section 4 illustrates both the quantitative and qualitative results on benchmarks and Section 5 concludes the paper.

2. Related Works

SISR is an important computer vision task, which has broad applications such as photo capture, surveillance and entertainment. In the last decades, numerous prominent approaches have been proposed to solve this ill-posed problem and obtained a tremendous progress in the performance. Unfortunately, current SISR methods require massive computations to process an input image. The energy consumption has become a thorny issue restricting the application of SISR on mobile devices.

2.1. Model Compression

Model compression has been investigated for many years and vast novel methods [10, 5, 17, 32, 19, 33] have been proposed. These methods can be roughly divided into four categories: network pruning, efficient filter design, neural architecture search (NAS) and knowledge distillation. Pruning [9, 35, 14, 22] aims at reducing the redundancy of filters so as to decrease the computation of the original model. Hou *et al*. [14] proposed a new pruning criterion for SISR which judged redundant channels with the discriminant information. The most conventional method is to design efficient block (*e.g.* GhostNet [10], CARN [1], IDN [15], MAFFSRN [25]) with efficient operators (*e.g.* group convolution, 1×1 convolution). Furthermore, NAS [30, 8, 6] has been employed to exploit efficient SR neural architecture automatically. In addition, knowledge distillation [12, 7, 6, 38] can transfer the information from large models to improve the performance of the tiny model. Gao *et al*. [7] proposed a novel knowledge distillation scheme for SISR and boosted the performance.

2.2. Low-Cost Computation

In addition to reduce the computation of models, there are two kinds of approaches that decrease the energy consumption [13] while preserving the architecture of original network. Model quantization [3, 23, 20, 37] saves energy [13] by reducing the number of bits required to represent each weight or feature component. Wherein, binarization is a specific case that weights and activations in networks are represented as $\{+1, -1\}$. Li *et al*. [20] proposed a novel quantization scheme for SISR to acquire a large dynamic quantization range. Xin *et al*. [37] designed a bit-accumulation mechanism to alleviate the quantization error of binary SISR networks. Unfortunately, the quantized network often cannot maintain the accuracy of super-resolution networks. On the other hand, many research works [13, 31, 40] have indicated that add addition consumes fewer energy than multiplication. Recently, Chen et al. [2] pioneered a novel method to reduce the power dissipation of network by taking the place of the multiplication with add operation. It achieved marginal loss of accuracy on classification tasks without any multiplication in convolutional layers. Then, kernel based progressive distillation [38] promoted the accuracy of AdderNets even superior to that of standard CNNs. It is attractive to construct energy efficient SISR models with AdderNet.

3. AdderNet for Image Super-Resolution

AdderNet reduces the energy consumption of classification networks significantly while achieving comparable performance. We aim to inherit this huge success to the image super-resolution task, which often has higher energy consumption and computational complexity. In SISR, there are two important properties that should be ensured by adder neural networks: the similarity between the input and output features of each convolutional layer, and the enhance-
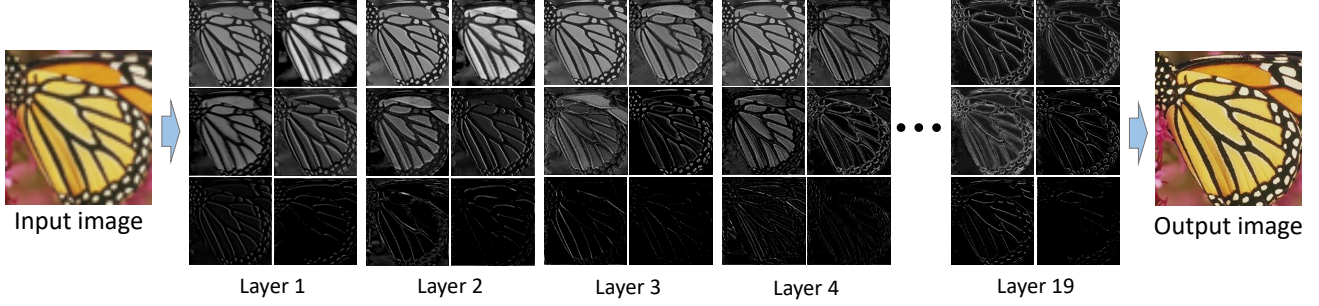
Figure 2. The output feature maps of different layers in VDSR [16] for the input image "Butterfly". The differences between any two adjacent layers are very similar with both texture and color information. The details w.r.t. the high-frequency information will be enhanced as the depth increases. These two important properties should be ensured by adder neural networks.

ment of details w.r.t. high frequency information. These two properties are illustrated in Fig. 2.

3.1. Preliminaries and Motivation

Here we firstly briefly introduce the single image super-resolution tasks using deep learning methods and then discuss the difficulty for directly using AdderNets to construct energy efficient SR models.

SISR aims at reconstructing a high-resolution image from a low-resolution image. It is a typical ill-posed reverse problem since an infinite number of high-resolution images could generate the same low-resolution image with downsampling. The objective function for the conventional SISR task can be formulated as:

$$\arg\min_{\mathbf{I_x}} \mathbf{I_y} = \mathbf{I_x} + \lambda \mathcal{R}(\mathbf{I_x}) \qquad (1)$$

where $\mathbf{I_y}$ is the observation data, *i.e.*, the low-resolution image, and $\mathbf{I_x}$ is the desired high-resolution image, $\mathcal{R}(\cdot)$ denotes the used priori such as smooth and additive noise, λ is the tradeoff parameter.

Recently, Dong *et al.* [4] first introduced deep learning method to super-resolution and achieved much better performance than traditional methods. Along with the improvement of super-resolution performance, the parameters and computations of SR networks grew rapidly, which seriously limited the efficiency of model executing on mobile devices. Hence, another research direction is to deploy efficient SR networks. Quantization, knowledge distillation, efficient operator designing and NAS have been explored to exploit efficient and accurate SR model. However, energy consumptions required by these portable SISR models are still much expensive for real-world mobile devices.

Distinguish to the existing model compression techniques such as pruning and quantization, Chen *et al.* [2] presented a fundamentally different way to reduce the energy consumption by replacing multiplication by cheap addition. AdderNet with the same architecture as that of ResNet-50 obtains a 91.7% top-5 accuracy while eliminating massive multiplications. For the parameters w.r.t. the filters

$\mathbf{W} \in \mathbb{R}^{k \times k \times c_{in} \times c_{out}}$ in an arbitrary layer in AdderNets, where k is the size of kernel, c_{in} and c_{out} are the number of input channels and output channels, respectively. Denote the input feature as $\mathbf{X} \in \mathbb{R}^{h \times w \times c_{in}}$, where w and h are the width and height of the input feature, respectively. The original adder operation is:

$$\mathbf{Y}(m, n, q) = -\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{g=1}^{c_{in}} |\mathbf{X}(m+i, n+j, g) - \mathbf{W}(i, j, g, q)|,$$
(2)

where $|\cdot|$ is the absolute value function. m and n denote the spatial location of features. q is the index of output channels. Although Eq. 2 shows comparable performance on image classification tasks, the SISR problem defined in Eq. 1 is quite different to the conventional recognition task. For example, we need to ensure that the output results maintain the original texture in $\mathbf{X}$(*i.e.*, $\mathbf{I_y}$), which cannot be easily learned by Eq. 2. Therefore, we should design a new framework of AdderNet for energy-efficient SISR models.

3.2. Learning Identity Mapping using AdderNet

Usually, an arbitrary super-resolution model using neural network learns the mapping from input LR image to HR image in an end-to-end manner. In addition to enhancing the high-frequency details, the overall texture and color information should be also maintained. Figure 2 illustrates the feature maps of different convolutional layers for a given LR image using VDSR [16]. It can be seen that the difference between the input feature map and output feature map of each convolutional layer are very similar. This observation reveals that the identity mapping (*i.e.*, $\mathbf{I_y} = \mathcal{F}(\mathbf{I_y})$) in SISR task using deep learning method is very essential.

For the traditional convolutional network, the identity mapping is very easy to learn when the weight is an identity matrix. However, adder neural network defined in Eq. 2 employs ℓ_1-norm to measure the distance between filter and input data. Although the ℓ_1 distance can perform well on the image classification task, the adder filter cannot approximate the identity mapping as described in Theorem 1.

Stacking more adder layers may alleviate this problem, but it will increase the model size and complexity significantly.

Theorem 1. *For the arbitrary low-resolution image $\mathbf{I}_y$, and an adder filter with its weight parameters $\mathbf{W}$. There exists no $\mathbf{W}$ satisfying the following equation:*

$$\mathbf{I_y} = \mathbf{I_y} \oplus \mathbf{W}, \tag{3}$$

where $\oplus$ denotes the adder operation defined in Eq. 2.

Proof. Here we consider a fully-connected layer for simplicity. Assuming that there exists a filter $\mathbf{W}$ that satisfies $\mathbf{I_y} = \mathbf{I_y} \oplus \mathbf{W}$ for any $\mathbf{I_y} \in \mathbb{R}^{d \times d \times c}$, where $\mathbf{W} \in \mathbb{R}^{d \times d \times c}$. Then for each element in $\mathbf{I_y}$, we have:

$$\mathbf{I_y}(m,n,q) = -\sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{g=1}^{c}|\mathbf{I_y}(m+i,n+j,g)-\mathbf{W}(i,j,g,q)|, \tag{4}$$

where $m \in \{1,\ldots,d\}, n \in \{1,\ldots,d\}, q \in \{1,\ldots,c\}$.

Then, we can select proper values for each element in $\mathbf{I_y^1}$ such that $\mathbf{I_y^1}(m,n,q) > \max_{i,j,k,g}|\mathbf{W}(i,j,k,g)|$, we have:

$$\mathbf{I_y^1}(m,n,q) = -\sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{g=1}^{c}[\mathbf{I_y^1}(m+i,n+j,g)-\mathbf{W}(i,j,g,q)]. \tag{5}$$

Let $\mathbf{I_y^2} = \mathbf{I_y^1} + 1$, then we have:

$$\begin{aligned}\mathbf{I_y^1}(m,n,q) + 1 &= -\sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{g=1}^{c}[\mathbf{I_y^1}(m+i,n+j,g)+1\\ &\quad - \mathbf{W}(i,j,g,q)]\\ &= -\sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{g=1}^{c}[\mathbf{I_y^1}(m+i,n+j,g)\\ &\quad - \mathbf{W}(i,j,g,q)] - d^2 \times c.\end{aligned} \tag{6}$$

Combining Eq. 5 and 6, we have $d^2 \times c + 1 = 0$, which is obviously impossible. In addition, the above proof can be easily extended to convolution layers in which the filter size of $\mathbf{W}$ is often much smaller than that of the input data $\mathbf{I_y}$. ☐

According to Theorem 1 and above analysis, the identity mapping cannot be directly learned using a one-layer adder neural network. To address this problem, we propose to refine the existing adder unit for adjusting the super-resolution task. In practice, we present a self-shortcut operation for each adder layer, *i.e.*,

$$\mathbf{Y}^l = \mathbf{X}^l + \mathbf{W}^l \oplus \mathbf{X}^l. \tag{7}$$

where $\mathbf{W}^l$ is the weights of adder filters in the l-th layer, $\mathbf{X}^l$ and $\mathbf{Y}^l$ are the input data and output data, respectively. Since the output of Eq. 7 contains the input data $\mathbf{X}^l$ itself,

we can utilize it to approximate the identity mapping by reducing the magnitude of $\mathbf{W}^l \oplus \mathbf{X}^l$.

Different to the image recognition tasks, features in most of SISR problems should maintain a fixed size, *i.e.*, the width and height of $\mathbf{X}^l$ and $\mathbf{Y}^l$ are exactly the same. Thus, the new calculation as described in Eq. 7 can be embedded into most of conventional SISR models. In the following experiments, we will utilize Eq. 7 to replace most of convolutional layers whose output size is the same as that of the input size.

3.3. Learnable Power Activation

Besides the identity mapping, there is another important functionality of traditional convolution filters that cannot be easily ensured by adder filters. The goal of a SISR model is to enhance the details including color and texture information of the input low-resolution images. Therefore, the high-pass filter is also a very important component in most of existing SISR models. It can be found in Figure 2, the details of the input image are gradually enhanced when the network depth is increased.

Generally, natural images are composed of different-frequency information. For example, the background and a large area of grass are low-frequency information, where most of neighbor pixels are very closed. In contrast, edges of objects and some buildings are exactly high-frequency information for the given entire image. In practice, if we define an arbitrary image as the combination of high-frequency part and the low-frequency part as $\mathbf{I} = \mathbf{I}_H + \mathbf{I}_L$, an ideal high-pass filter $\Phi(\cdot)$ used in the super-resolution task and other image processing problems could be defined as

$$\Phi(\mathbf{I}) = \Phi(\mathbf{I}_H + \mathbf{I}_L) = \sigma(\mathbf{I}_H), \tag{8}$$

which only preserves the high-frequency part of the input image. Wherein, $\sigma(\mathbf{I}_H)$ is the convolution response of the high-frequency part. The above equation can help the SISR model removing redundant outputs and noise and enhancing the high-frequency details, which is also a very essential component in the SISR models.

Similarly, for the traditional convolution operation, the functionality of Eq. 8 can be directly implemented. For example, a 2×2 high-pass filter $\begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix}$ can be used for removing any flat areas in $\mathbf{I}$. However, for the adder neural network as defined in Eq. 2, it is impossible to achieve the functionality as described in Eq. 8.

Theorem 2. *Let $\mathbf{E} \in \mathbb{R}^{d \times d}$ be an input image for the given SR model using adder operation, where each element in $\mathbf{E}$ is equal to 1. $\mathbf{W}$ denotes the weights of an arbitrary adder filter. There exists no $\mathbf{W} \in \mathbb{R}^{d \times d}$ and a constant $a \in \mathbb{R}$*

satisfying the following equation:

$$(s * \mathbf{E}) \oplus \mathbf{W} = a, \qquad (9)$$

where $s \in \mathbb{R}$, and $\oplus$ denotes the adder operation which contains only additions as defined in 2.

Proof. Assuming that there exists an adder filter $\mathbf{W}$ that satisfies $(s * \mathbf{E}) \oplus \mathbf{W} = a$, for any $s \in \mathbb{R}$. We can find a s such that $s > max_{i,j} |\mathbf{W}(i,j)|$. Then we have:

$$-\sum_{i=1}^{d}\sum_{j=1}^{d}[(s+1)\mathbf{E}(i,j) - \mathbf{W}(i,j)] = a =$$
$$-\sum_{i=1}^{d}\sum_{j=1}^{d}[s\mathbf{E}(i,j) - \mathbf{W}(i,j)], \qquad (10)$$

which means $\sum_{i=1}^{d}\sum_{j=1}^{d}(s+1)\mathbf{E}(i,j) = \sum_{i=1}^{d}\sum_{j=1}^{d} s\mathbf{E}(i,j)$ and leads to a contradiction. □

According to the above theorem, the functionality of high-pass filter cannot be replaced by adder filter. Thus, the super-resolution process in the adder neural networks will involve more redundancy. To this end, we need to develop a new scheme for using adder neural networks to conduct the SISR tasks that can compensate this defect.

Admittedly, we can also add some parameters and filters to improve the capacity of the SR model using adder units, the save on energy and calculation will be reduced. Fortunately, Sharabati and Xi [29] applied the Box-Cox transformation [28] in the image denoising task and found that this transformation can achieve the similar functionality to that of the high-pass filters without adding massive parameters and calculations. Oliveira *et al.* [26] further discussed the functionality of Box-Cox transformation in image super-resolution. In addition, a sign-preserving power law point transformation is also explored for emphasizing the areas with abundant details in the input image [27]. Therefore, we propose a learnable power activation function to solve the defect of AdderNet and refine the output images, *i.e.*,

$$\mathcal{P}(\mathbf{Y}) = \text{sgn}(\mathbf{Y}) \cdot |\mathbf{Y}|^{\alpha}, \qquad (11)$$

where $\mathbf{Y}$ is the output features, $\text{sgn}(\cdot)$ is the sign function, $\alpha > 0$ is a learnable parameter for adjusting the information and distribution. When $\alpha > 1$, the above activation function can enhance the contrast of output images and emphasize the high-frequency information. When $0 < \alpha < 1$, Eq. 11 can smooth all signals in the output image and remove artifacts and noise. In addition, the above function can be easily embedded into the conventional ReLU in any SISR models.

By exploiting these two methods described in Eq. 7 and Eq. 11, we can address the aforementioned problems on

using adder networks for conducting the SISR task. Although there are some additional computations introduced, *e.g.*, Eq. 7 needs a shortcut to maintain the information in the input data, and the learnable parameter α in Eq. 11 leads to some additional multiplications. However, compared with the massive operations required by either convolutional layer or adder layer, they are very trivial. For example, the number of adder operations defined in Eq. 2 is about $2k^2 c_{in} c_{out} hw$, and the additional computations required by Eq. 7 and Eq. 11 are both equal to $c_{out} hw$. Considering that $2k^2 c_{in}$ is usually a relatively large value in modern deep neural architectures (*e.g.*, $k = 3$ and $c_{in} = 64$), the additional computation for each layer is over $1000\times$ lower than that of the original method. In the next section, we will conduct extensive experiments to illustrate the superiority of the proposed method on both the visual quality and energy consumptions.

4. Experiments

In the above section, we have developed a series of new operations for establishing SISR models using adder neural networks. Here we will conduct experiments to verify the effectiveness of the proposed AdderSR networks.

Datasets. To evaluate the performance of adder neural networks on super-resolution tasks, we select several benchmark image datasets to conduct the experiments. Following the setting of VDSR [16], *291* dataset is employed to train adder VDSR networks. It consists of 91 images from Yang *et al.* [39] and 200 images from Berkeley Segmentation Dataset [24]. In addition, DIV2K dataset [34] is utilized to train adder EDSR networks in the following experiments. This dataset consists of 800 training images and 100 validation images. In order to compare with other state-of-the-art methods, four relatively small benchmarks are also selected including Set5, Set14, B100 and Urban100. The LR image is generated with bicubic downsampling. Super-resolution results are evaluated using both peak signal-to-noise ratio (PSNR [4]) and structure similarity index (SSIM [36]) on Y channel (*i.e.*, luminance) of YCbCr space.

Training setting. We evaluate the performance of the proposed AdderSR networks using two famous neural architectures for super-resolution, *i.e.*, VDSR [16] and EDSR [21], which are shown extraordinary performance for generating images with high visual quality. Following the setting of the conventional AdderNet, we do not replace the first and the last convolutional layers in these networks, and the batch normalization is employed on each adder layer. To accelerate convergence speed of AdderSR networks, the learning rate for adder layers in our models is enlarged 10 times than that of convolutional layers in baselines. Specifically, the learning rates of adder layer and convolutional
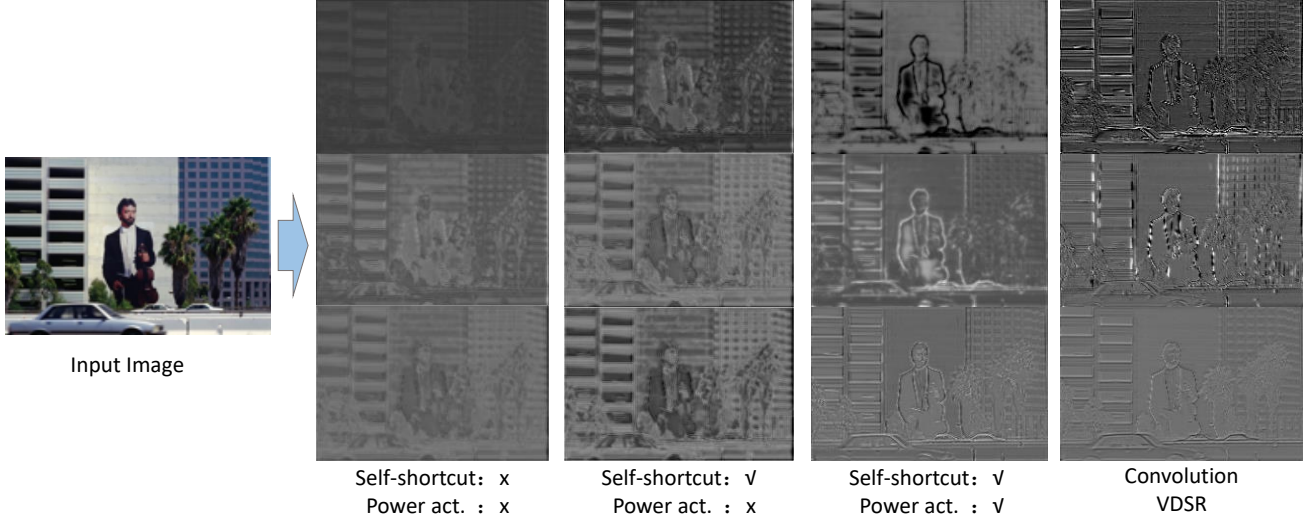
Figure 3. The output feature maps of adder layers in AdderSR networks with different strategies. Feature maps of adder layer without self-shortcut (Eq. 7) cannot maintain the overall texture information of their input features. The power activation function (Eq. 11) effectively enhances high-frequency regions.

Table 1. Ablation study of the proposed AdderSR network. Wherein, results of VDSR model on the four datasets are directly reported from the original paper [16].

Model	Architecture		Set5	Set14	B100	Urban100
	Self-shortcut (Eq. 7)	Power Act. (Eq. 11)	PSNR	PSNR	PSNR	PSNR
Adder VDSR 1	×	×	35.11	31.40	30.63	27.77
Adder VDSR 2	×	✓	35.44	31.62	30.78	27.96
Adder VDSR 3	✓	×	37.10	32.77	31.69	30.05
Adder VDSR 4	✓	✓	37.37	32.93	31.81	30.48
Conv. VDSR [16]	−	−	37.53	33.03	31.90	30.76

layer are initialized as 3×10^{-3} and 3×10^{-4}, respectively. The optimizer utilized in adder neural network for super-resolution task is ADAM [18] due to its fast convergence speed. Hyper-parameters here are set as $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Other settings such as patch size and data augmentation strategies are totally the same as those in baseline VDSR and EDSR.

Ablation study. To ensure the performance of adder neural networks on the SISR tasks, we have thoroughly designed two new operations in Eq. 7 and Eq. 11. Here we will first conduct the detailed ablation study to illustrate their functionalities. In practice, the VDSR using convolutional layers is selected as baseline, and we replace all intermediate layers by adder layers as described in Eq. 2. Table 1 shows results of ablation experiments. Without Eq. 7 and Eq. 11, the PSNR of AdderSR is 2.08 *dB* lower than that of conventional VDSR network on the average of four benchmark datasets. Such a PSNR value decline will increase the artifacts in the resulting high-resolution images. In Eq. 2, the self-shortcut makes it possible to optimize an identity mapping. Thus, the performance of AdderSR net-

work using Eq. 2 can obtain an about 1.67 *dB* PSNR enhancement on average. In addition, Eq. 11 is proposed to emphasize the high-frequency information in intermediated features and reconstructed images. By embedding Eq. 11 into AdderSR network, the PSNR value can be further improved with about 0.24 *dB* on benchmark datasets. If both Eq. 7 and Eq. 11 are employed on AdderSR network, the performance can be improved with 1.91 *dB* on the average of four datasets. The final result of AdderSR network is pretty close to that of conventional VDSR. Detailed visualization results of these models can be found in the supplementary materials.

Feature Visualizations. To have an explicit illustration on the functionalities of different components in our models using adder layers, we visualize the feature maps of Adder VDSR model 1, 3 and 4 reported in Table 1. From Figure 3, we can see that the feature maps of AdderSR network without Eq. 7 are quite different from the input image. The texture information such as human face, cars and trees in the input image are blurred or distorted. In contrast, the features maps of AdderSR network using Eq. 7 preserve more

Figure 4. Visualization of super-resolution image of AdderSR Network and CNN on ×3 scale.

details information than those of the model without Eq. 7. Moreover, it is obvious that the high-frequency information (*e.g.*, edge, corner) is emphasized while a portion of low-frequency information is eliminated with the help of Eq. 11. That is a very important functionality for proceeding high-resolution images. In summary, by exploiting the proposed method, we can generate features with abundant texture and establish effective SISR models using only additions.

Overall Comparison. After investigating the impact of different components of the proposed method, we then conduct the experiments using the benchmark VDSR and EDSR architectures with different scaling factors, *i.e.*, ×2, ×3 and ×4 to illustrate the superiority of the proposed AdderSR networks for the SISR problem. Table 2 shows the average PSNR and SSIM results of different models using both convolutional layers and adder units on Set5, Set14, B100, Urban100, respectively. It can be found in Table 2, our models using only additions in intermediate layers can also achieve comparable PSNR and SSIM values to those of their baselines with massive multiplications. The average

PSNR gap between AdderSR models and baselines using convolutions over the four datasets is about 0.17 *dB*.

Besides the quantitative comparison, we also provide the comparison on the visual qualities of the same architecture using both additions and multiplications as shown in Figure 4. Actually, since the PSNR values of AdderSR networks and their baselines as shown in Table 2 are very closed, the output high-resolution images are of the similar visual quality. Particularly, the network using adder units can also effectively reconstruct the texture and color information. These results demonstrate that, besides the visual recognition task, we can also utilize additions to establish excellent neural networks for solving the SISR image processing problem. More visualization results of these models can be found in the supplementary materials.

Energy Consumptions. As discussed in previous works [2, 13, 31, 40], additions require much lower energy consumptions compared with multiplications. We further calculate the energy consumptions of different networks used in Table 2. In practice, most of values in recent

Table 2. Quantitative results of baseline convolutional networks and our AdderSR models. Wherein, ×2, ×3 and ×4 are the output scaling factors for the SISR task. ANN and CNN denote the networks using adder units and traditional convolution layers, respectively. "G" represents 10^9.

Scale	Model	Type	#Mul. (G)	#Add. (G)	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM
×2	VDSR	ANN	1.1	1224.1	37.37/0.9583	32.93/0.9115	31.81/0.8945	30.48/0.9104
		CNN	612.6	612.6	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
	EDSR	ANN	7.9	18489.8	37.92/0.9589	33.82/0.9183	32.23/0.9000	32.63/0.9309
		CNN	9248.9	9248.9	38.11/0.9601	33.92/0.9195	32.32/0.9013	32.93/0.9351
×3	VDSR	ANN	1.1	1224.1	33.54/0.9204	29.75/0.8306	28.76/0.7958	26.95/0.8220
		CNN	612.6	612.6	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
	EDSR	ANN	7.1	8825.2	34.35/0.9212	30.33/0.8420	29.13/0.8068	28.54/0.8555
		CNN	4416.1	4416.1	34.65/0.9282	30.52/0.8462	29.25/0.8093	28.80/0.8653
×4	VDSR	ANN	1.1	1224.1	31.19/0.8804	27.93/0.7646	27.22/0.7225	25.02/0.7447
		CNN	612.6	612.6	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
	EDSR	ANN	6.8	5442.6	32.13/0.8864	28.57/0.7800	27.58/0.7368	26.33/0.7874
		CNN	2724.7	2724.7	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033

Table 3. Quantitative results of CARN and pruned architectures using convolutional and adder layers, respectively. PSNR values are reported on Urban 100 database.

Model	CARN	CARN-$\frac{2}{3}$	CARN-$\frac{1}{2}$	CARN-$\frac{1}{4}$	A-CARN	A-CARN-$\frac{2}{3}$	A-CARN-$\frac{1}{2}$	A-CARN-$\frac{1}{4}$
PSNR(dB)	31.92	31.35	31.15	30.40	31.74	31.27	30.94	30.21
Energy(pJ)	1027G	404G	260G	66G	402G	160G	102G	26G

Table 4. The energy consumptions of different networks for ×2 scale. The energy cost is computed using 720p (*i.e.* 1280×720) high-resolution image.

Model	VDSR		EDSR	
	CNN	ANN	CNN	ANN
Energy cost (pJ)	2817.9G	1105.6G	42544.7G	16670.3G

SISR models are 32-bit floating numbers, and the energy consumptions for a 32-bit addition and multiplication are 0.9 *pJ* and 3.7 *pJ*, respectively. The amount of remaining multiplication operations in AdderSR network is extremely small compared with the FLOPs of the entire network. The detailed energy costs of the these networks are reported in Table 4, which is computed according to literature [13, 31]. Obviously, the proposed AdderSR method can reduce the energy cost for reconstructing a 1280×720 image by a fact of about 2.5×. If we further quantize the weights and activations of these models shown in Table 2 to int8 values, we can obtain an about 3.8× reduction on the energy consumption using the proposed AdderSR with comparable performance. These results can be found in supplementary materials. We also conduct experiments using our methods on recent lightweight SR models (*i.e.*, CARN [1]) and report the energy consumption v.s. performance comparison in Table 3, where CARN-$\frac{2}{3}$, CARN-$\frac{1}{2}$ and CARN-$\frac{1}{4}$ are the models with $\frac{2}{3}$, $\frac{1}{2}$ and $\frac{1}{4}$ channels after applying filter pruning, respectively. A-CARN denotes the model using AdderNets. We can achieve the comparable PSNR

values on compact SISR architectures. The performance of A-CARN is about 0.4dB higher than that of CARN-$\frac{2}{3}$ with similar energy consumption (404G*pJ*) after pruning. This significant reduction on energy consumption will make these deep learning models portable on mobile devices.

5. Conclusions and Discussion

This paper investigates the single image super-resolution problem using AdderNets. Without changing the original neural architectures, we develop a new adder unit and a novel learnable power activation for addressing the defects in existing adder neural networks. The new AdderSR models can learn the functionalities of conventional identity mapping and high-pass filter, which are essential for providing images with high visual quality. Experimental results conducted on several benchmarks illustrate that, the proposed AdderSR networks can achieve the similar visual quality to that of their baselines using traditional convolution filters. Meanwhile, since additions are much cheaper than multiplications, we can reduce the energy consumptions of these models by about 2.5×. Besides the image super-resolution task, the techniques in this paper can be well transferred to other image processing problems including denoising, deblurring, *etc*. Future works will focus on more applications and low-bit quantization versions of these networks to achieve higher reduction on the energy consumption.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pages 252–268, 2018. 2, 8

[2] Hanting Chen, Yunhe Wang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. Addernet: Do we really need multiplications in deep learning? In *CVPR*, 2020. 2, 3, 7

[3] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*, pages 3123–3131, 2015. 1, 2

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199, 2014. 1, 3, 5

[5] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407. Springer, 2016. 2

[6] Yonggan Fu, Wuyang Chen, Haotao Wang, Haoran Li, Yingyan Lin, and Zhangyang Wang. Autogan-distiller: Searching to compress generative adversarial networks. In *ICML*, 2020. 1, 2

[7] Qinquan Gao, Yan Zhao, Gen Li, and Tong Tong. Image super-resolution using knowledge distillation. In *ACCV*, pages 527–541. Springer, 2018. 1, 2

[8] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *ICCV*, pages 3224–3234, 2019. 2

[9] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *NeurIPS*, pages 1379–1387, 2016. 1, 2

[10] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *CVPR*, pages 1580–1589, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NeurIPS workshops*, pages 1–9, 2015. 1, 2

[13] Mark Horowitz. Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14, 2014. 2, 7, 8

[14] Zejiang Hou and Sun-Yuan Kung. Efficient image super resolution via channel discriminative deep neural network pruning. In *ICASSP*, pages 3647–3651. IEEE, 2020. 1, 2

[15] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *CVPR*, pages 723–731, 2018. 2

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 3, 5, 6

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645, 2016. 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[19] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 624–632, 2017. 2

[20] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Yuchao Li, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams: Quantized super-resolution via parameterized max scale. In *ECCV*, 2020. 1, 2

[21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017. 5

[22] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *CVPR*, pages 1529–1538, 2020. 1, 2

[23] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In *CVPRW*, pages 0–0, 2019. 1, 2

[24] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, 2001. 5

[25] Abdul Muqeet, Jiwon Hwang, Subin Yang, Jung Heum Kang, Yongwoo Kim, and Sung-Ho Bae. Ultra lightweight image super-resolution with multi-attention layers. In *ECCV Workshop*, 2020. 2

[26] Nícholas André Pinho de Oliveira et al. Single image super-resolution method based on linear regression and box-cox transformation. 2018. 5

[27] Giovanni Ramponi, Norbert K Strobel, Sanjit K Mitra, and Tian-Hu Yu. Nonlinear unsharp masking methods for image contrast enhancement. *Journal of electronic imaging*, 5(3):353–367, 1996. 5

[28] Remi M Sakia. The box-cox transformation technique: a review. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(2):169–178, 1992. 5

[29] Walid K Sharabati and Bowei Xi. Fast local polynomial regression approach for speckle noise removal. In *ICPR*, pages 3198–3203, 2016. 5

[30] Dehua Song, Chang Xu, Xu Jia, Chunjing Xu, and Yunhe Wang. Efficient residual dense block search for image super-resolution. In *AAAI*, 2020. 2

[31] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017. 2, 7, 8

[32] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017. 2

[33] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, pages 4539–4547, 2017. 2

[34] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, pages 114–125, 2017. 5

[35] Haotao Wang, Shupeng Gui, Haichuan Yang, Ji Liu, and Zhangyang Wang. Gan slimming: All-in-one gan compression by a unified optimization framework. In *ECCV*, 2020. 1, 2

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 5

[37] Jingwei Xin, Nannan Wang, Xinrui Jiang, Jie Li, Heng Huang, and Xinbo Gao. Binarized neural network for single image super resolution. In *ECCV*, 2020. 1, 2

[38] Yixing Xu, Chang Xu, Xinghao Chen, Wei Zhang, Chunjing Xu, and Yunhe Wang. Kernel based progressive distillation for adder neural networks. In *NeurIPS*, 2020. 1, 2

[39] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873, 2010. 5

[40] Haoran You, Xiaohan Chen, Yongan Zhang, Chaojian Li, Sicheng Li, Zihao Liu, Zhangyang Wang, and Yingyan Lin. Shiftaddnet: A hardware-inspired deep network. *NeurIPS*, 33, 2020. 2, 7

[41] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018. 1