Mobile Networks and Applications (2021) 26:13–26
https://doi.org/10.1007/s11036-020-01720-2

ASDN: A Deep Convolutional Network for Arbitrary Scale Image Super-Resolution

Jialiang Shen[1] · **Yucheng Wang**[1,2] · **Jian Zhang**[1]

Accepted: 30 November 2020 / Published online: 22 February 2021

Abstract

Deep convolutional neural networks have significantly improved the peak signal-to-noise ratio of Super-Resolution (SR). However, image viewer applications commonly allow users to zoom the images to arbitrary magnification scales, thus far imposing a large number of required training scales at a tremendous computational cost. To obtain a more computationally efficient model for arbitrary-scale SR, this paper employs a Laplacian pyramid method to reconstruct any-scale high-resolution (HR) images using the high-frequency image details in a Laplacian Frequency Representation. For SR of small-scales (between 1 and 2), images are constructed by interpolation from a sparse set of precalculated Laplacian pyramid levels. SR of larger scales is computed by recursion from small scales, which significantly reduces the computational cost. For a full comparison, fixed- and any-scale experiments are conducted using various benchmarks. At fixed scales, ASDN outperforms predefined upsampling methods (e.g., SRCNN, VDSR, DRRN) by about 1 dB in PSNR. At any-scale, ASDN generally exceeds Meta-SR on many scales.

Keywords Image super-resolution · Any-scale SR · Convolutional neural network

1 Introduction

Deep neural networks have made good progress in Single-image Super-Resolution (SISR), adeptly extracting image priors from data sets and efficiently learning mapping functions from LR to HR patches. However, for applications that allow users to zoom to arbitrary scales (e.g., face image SR [5] and satellite image SR [18]), multi-scale methods which learn the LR to HR mapping functions independently at each of several scales [12, 15, 29] become inefficient. Meta-SR [8] shows that SR of arbitrary decimal scales can be achieved by training one single model with the dynamic meta-upscaling module. But meta-SR can only generate HR images on scales for which it has trained, making it computationally impractical to train for all scales of interest for any-scale SR.

To alleviate the need for so many training scales, we find image patches have the same similarity at different scales. The self-similarity-based SR method [17] enhances the textural content with similar patches across different scales. Furthermore, image edges are scalable, and different-scale images have similar edge information, represented by high-frequency image information. In order to seek the missing high-frequency information of SR images, a Laplacian pyramid based-method is proposed to interpolate between a sparse set of trained scales. Indeed, the Laplacian filter is an edge detector, and the Laplacian noise term can be used to detect the outliners for robust tracking [24]. Therefore, similar high-frequency image information across different scales can be highlighted through the Laplacian pyramid structure. Moreover, the Laplacian pyramid structure has been proved to reduce the training data requirements for multi-scale SR in MS-LapSRN [14], generating the 3× HR images with the 4× SR results and predicting 8× HR images by progressively deploying through the network for 2× SR. Therefore, it is feasible to reduce the training costs with a Laplacian Pyramid [13] network structure.

✉ Jialiang Shen
Jialiang.Shen@student.uts.edu.au

Yucheng Wang
Yucheng.Wang@alumni.uts.edu.au

Jian Zhang
Jian.Zhang@uts.edu.au

[1] University of Technology Sydney, Sydney, Australia

[2] Present address: ByteDance, Beijing, China

Unlike previous Laplacian Pyramid networks for multi-scale SR, we seek to train a model to predict any-scale SR images. Obviously, a large upsampling ratio can be expressed as an integer power of ratios in a small range. Therefore, given a network for super-resolution at scales in a small range (such as the real-number interval $(1, 2])$, arbitrary larger scales (real numbers greater than 2) can be implemented by recursion. Inspired by the classical Laplacian pyramid method [3], which reconstructs HR images by restoring the residual images between two Laplacian pyramid levels, we introduce a Laplacian Frequency Representation to learn the mapping function for SR of scales in the small range $(1, 2]$. Our algorithm represents the HR images of any continuous decimal scale in the range by the two neighboring Laplacian pyramid levels. For SR of the large decimal ratios, we progressively upscale the coarse HR images, and recursively deploy them through the network multiple times with a small decimal ratio in the range to gradually refine the HR images.

In this paper, we propose our network as Any-Scale Deep Super-Resolution Network (ASDN) based on the multi-scale parallel reconstruction architecture. Each reconstruction branch shares the Feature Mapping Branch (FMB) and predicts the Laplacian pyramid levels through the Image Reconstruction Branch (IRB). Our network requires a minimal amount of training data and computational resources but effectively generates any-scale SR results.

We present extensive comparisons on both fixed integer scales and any decimal scale on commonly used benchmarks, and provide the results of the ASDN and the fine-tuned ASDN (FSDN), for the reference in comparison with the existing multi-scale SR methods. ASDN outperforms all of the other predefined upsampling methods and even some single upsampling models, without training on the specific-scale data samples. FSDN has state-of-the-art performance for fixed scale SR, comparing favorably to all existing methods. For any-scale SR factor, we retrain many previous network structures [12, 15, 29] with our any-scale SR method into any-scale SR categories for comparison. Our ASDN is effective for SR of any desired scale and specifically achieves the state-of-the-art performance on scales within the small range $(1, 2]$.

In summary, our work provides the following contributions:

(1) **Laplacian Frequency Representation:** We propose a Laplacian frequency representation mechanism to reconstruct image SR at small scales, those continuously varying between 1 and 2. The HR images are the weighted interpolation of their two neighboring Laplacian pyramid levels, which efficiently reduces the training scale demands for learning the SR at continuous scales.

(2) **Recursive Deployment:** We introduce Recursive Deployment for generating the HR images of the larger upsampling ratios, as we find that the HR images of the larger scales can be gradually upsampled and recursively deployed with small ratios. This extends any-scale SR from small scales to larger ones without requiring additional training scales.

(3) **Any-scale Deep SR Network:** We propose an Any-Scale Deep Super-Resolution Network (ASDN) to generate HR images of any random scale with one unified network, providing enormous computational savings over directly applying existing CNN-based multi-scale methods for any-scale applications.

2 Related works

2.1 Image super-resolution using CNN

Image super-resolution has evolved greatly over the past decades, and numerous image SR methods [12, 15, 29] have been proposed to improve image reconstruction performance. With the fast development of the computation processor, CNN-based SR methods have demonstrated state-of-the-art results by optimizing an end-to-end network to learn the LR-HR mapping function. Dong et al. [4] initially introduced convolutional layers into image SR, which have been proved effective for the task. However, the network consists of only three layers, unable to observe superior results with the deeper model. He et al. [10] solved this problem by residually skip connecting layers inside the network to help the gradient flow across the deeper models. Later on, more skip connection structures, dense connection [23] were proved to accelerate network convergence by feature reusing across the layers. RDN [29] and DBDN [25], embed the dense convolutional neural network into image SR to further improve image reconstruction accuracy. Then, the attention module was adopted into the SR to help the network focus on the high-frequency feature learning. Liu et al. [16] introduced the spatial attention to mask out the high-frequency component locations in the HR images,and RCAN [28] replaced normal feature layers with residual channel layers to adaptively rescale channel-wise features to reduce the unnecessary computations for abundant low-frequency features. However, these methods mainly focus on multi-scale SR (e.g., $2\times$, $3\times$, and $4\times$). In this paper, we propose to reconstruct any-scale SR with a few numbers of training scales, which can significantly reduce the computational cost.

Any-scale SR model is seldom investigated in image SR. Recently, Meta-SR [8] proposed a meta-upscale module for arbitrary scale SR, which dynamic magnifies image with decimal scales, by training and testing with 40 different

scales at the stride of 0.1. However, Meta-SR [8] did not provide a systemic approach or experimental results for any scale that not included in the 40 trained scales. In other words, only training with 40 different scales, Meta-SR can not solve the SR of undetermined decimal scales. Nevertheless, if we use enormous scales of data to train the Meta-SR model for the full any-scale SR approximation, it might take a very long time to optimize the network for its convergence, which is not practical. Different from these methods trained with all the scales of interest, we propose a novel network ASDN for SR of any potential scale, which adopts our any-scale SR method, including Laplacian Frequency Representation and Recursive Deployment.

2.2 Laplacian pyramid structure

The Laplacian Pyramid [3] is used for restoring HR images by preserving residual image information. As shown in Fig. 1a, the decomposition step firstly preserves the residual information R_1, R_0, as image downscaled. Then the kept residual information R_1, R_0 will be stored back by adding with the low-resolution image I_1, I_0, to reconstruct the initial HR image H_1, H_0.

With the development of deep learning, many models adopt the Laplacian Pyramid structure as the main mapping frameworks, which construct progressive upsampling networks for image SR. Such as LapSRN [13] in Fig. 1b, a multi-phase network and each phase learn the residual information with convolutional layers. LapSRN progressively reconstructs each pyramid levels at the interval of 2 times, for $2\times$, $4\times$, and $8\times$ SR, respectively. MS-LapSRN [14] is the parameter sharing version of LapSRN, which shares the network parameters across pyramid levels and exhibits the efficiency of recursive deployment. However, these models are designed to effectively predict SR of large scale factors.

In this paper, we present Laplacian Frequency Representation to reconstruct SR results of continuous scales. In our design, each pyramid level is at the interval of 0.1 in scale and parallelly allocated at the end of the network. According to the Laplacian pyramid [3] that the lost high-frequency information can be presented by the two neighboring pyramid levels, the high-frequency information of HR image is predicted based on the weighted interpolation of the two Laplacian pyramid levels neighboring the testing scale. The Laplacian Frequency Representation entails fewer training SR samples to generate HR images of scales in the continuous ratio range, which reduces the undesired training data storage space and shrinks the optimization period to accelerate the network convergence.

3 Any-scale image super-resolution

In this section, we provide the mathematical background of the any-scale SR method including Laplacian Frequency Representation and Recursive Deployment and introduce the structure of our proposed ASDN.

3.1 Any-scale SR method

There are two steps in the any-scale SR method: Laplacian Frequency Representation and Recursive Deployment. The proposed Laplacian Frequency Representation method is to generate HR images of decimal scales in a continuous scale range, and Recursive Deployment is to define the recursion times N and the small ratio r at each recursion for any-scale

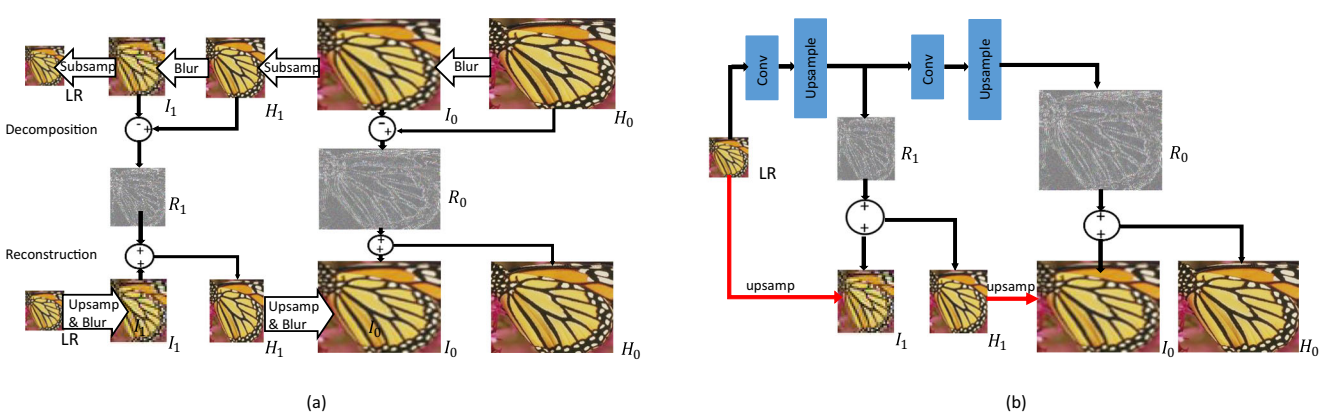


Fig. 1 Comparison of two-level Laplacian Pyramids. a Laplacian Pyramid [3]. The decomposition step produces two residual images R_1, R_0 by subtracting H_1 with I_1, H_0 with I_0 to preserve the high-frequency information, which then added with the interpolated LR I_1, I_0 respectively to reconstruct the $2\times$, $4\times$ frequency levels H_1, H_0. **b** LapSRN [13]. The residual images R_1, R_0 are progressively learned by the networks and upsampled at each level, then added with I_1, I_0 for HR images H_1, H_0

SR prediction. To use the minimum training samples, we define the small decimal ratios $r \in (1, 2]$. For SR of each upscaling ratio R, the HR image of upscale ratio R can be achieved by recursively upscaled with a small ratio r and deployed N times.

3.1.1 Laplacian frequency representation

To generate SR of decimal scales in a continuous range, the intuitive method is to train the network with random dense scales in the range. However, we find this method is difficult due to a large amount of training scale samples and computation power for optimizing the network. To deal with this problem, we introduce Laplacian Frequency Representation as the intermediate representation of the high-frequency image information of SR results.

As shown in Fig. 2, our proposed Laplacian Frequency Representation has L Laplacian pyramid levels, and each pyramid level l is tasked with learning the high-frequency image information of HR images O_{r_l} for the scale r_l with training samples of corresponding scales.

$$r_l = \frac{l}{L-1} + 1, l = 0, ..., L-1 \tag{1}$$

According to the scalability of the image edges and the comprehensive coverage of high-frequency information of images in edges, we can interpolate the high-frequency image details of SR results of any small decimal scales r based on their two neighboring Laplacian pyramid levels.

For a given scale factor r in this continuous range, the Laplacian frequency represented HR images O_r can be defined as

$$O_r = O_{r_i} + w_r * P_i \tag{2}$$

where

$$P_i = O_{r_{i-1}} - O_{r_i}, i = 1, ..., L-1 \tag{3}$$

Here the phase number $i = \lceil (L-1) * (r-1) \rceil$ and w_r is the weight parameter of the edge information for the r scale SR. We define the weight parameter according to distance proportion of the scale r to the r_i in the phase P_i.

$$w_r = (L-1) * (r_i - r) \tag{4}$$

The interpolated representation can be regarded as calculating the missing high-frequency image details of HR images of certain scales, so we name the mechanism as Laplacian Frequency Representation. The further evaluation of the accuracy of Laplacian Frequency Representation and the density of Laplacian pyramid levels in the experiment section proves that the represented SR results highly coordinate with the directly learned results, and the performance is stable when the Laplacian pyramid levels are at a certain density. As a result, we propose to train the Laplacian pyramid levels using deep neural networks with several scales and reconstruct the HR images of continuous decimal scales in the range with Laplacian Frequency Representation.

3.1.2 Recursive deployment

For SR of any upsampling ratio R in the larger range, it is impossible to train SR samples of all the scales to learn the mapping function of these scales. To minimize the training sample demands, we reuse the learned mapping network for SR of decimal scales in the range of $(1, 2]$. We are based on the idea that any upscale decimal ratio R can be expressed as an integer N power of decimal ratios r in a small range. Therefore, the HR images of R can be generated by gradually upscaling and recursively deploying through the mapping network N times with small decimal ratios $r \in (1, 2]$.

We express the R as an integer N power of small decimal ratios $r \in (1, 2]$. The integer N denotes the recursion times for the deployment, and the small ratio r is the upsampling ratio at each recursion. To determine the best solution of N and r for any-scale SR, several comparison experiments are performed in the experiment section. As we observed, SR with the larger upscale ratio r at the early recursions

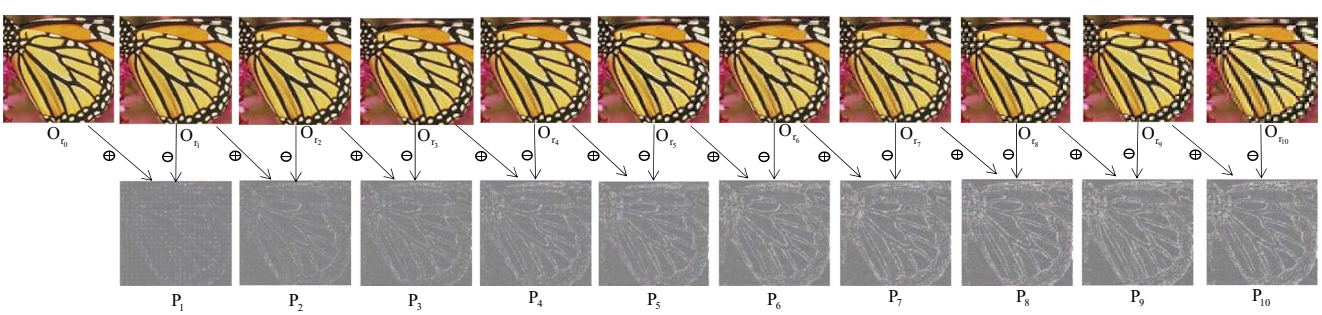


Fig. 2 The Laplacian Frequency Representation has 11 Laplacian pyramid levels ($O_{r_0}, ..., O_{r_{10}}$), with 10 phases in the scale range of $(1, 2]$ ($P_1, ..., P_{10}$). Each phase represents the difference between the successive Laplacian pyramid levels

and the smaller recursive deployment times N has better performance than other N and r solutions.

Therefore, for any scale factor R, the recursive times N

$$N = \lceil \log_2 R \rceil \tag{5}$$

The upscale ratio r_n at each recursion n can be defined as

$$r_n = \begin{cases} 2 & \text{if } n \leq N-1 \\ \frac{R}{2^{N-1}} & \text{if } n = N \end{cases} \tag{6}$$

Based on the defined N and r solution for R, if the recursion time N is 1, the HR images of $R = r$ are directly deployed by the network. In other situations, the coarse HR images from the previous recursion are bicubic upscaled with the small ratio r as the input LR images at the current recursion. For better SR performance, at the early $N-1$ recursions, the small ratio $r_n = 2$, and at the N_{th} recursion, $r_n = \frac{R}{2^{N-1}}$.

3.2 Any-scale SR deep network

In this section, we build a deep neural network to predict the Laplacian Frequency Representation from the input images.

3.2.1 Network architecture

The Laplacian Frequency Representation should consist of $L = 11$ Laplacian pyramid levels for SR in the scale range $(1, 2]$. Each Laplacian pyramid level is the reconstructed HR image containing high-frequency details. Due to the mutual relationship among different scales in the SR networks [12], our network for Laplacian Frequency Representation are based on the multi-scale parallel [28] framework by sharing the Feature Mapping Branch (FMB) across different scales and restoring HR images with separate Image Reconstruction Branches (IRBs). Sharing the FMB can largely reduce the computation capacity, and separating IRB reduces the complexity of the original learning problem and leads to an accurate result.

The feature mapping branch (FMB) of the Laplacian Frequency Representation is constructed by a deep convolutional neural network $H = f_{fmb}(I)$. As shown in Fig. 3, FMB consists of Bi-Dense structure [25] for efficient feature learning and channel attention modules [28] for highlighting high-frequency context information. In the Dense Attention Block (DAB), channel attention module (see Fig. 3c) connects right after the concatenated feature channels. Therefore, the high-frequency information of the concatenated channel features are highlighted before preceding into the next block and thus allow the network to focus on more useful channels to improve reconstruction performance.

The other part of the network f_{irb} is the image reconstruction branch (IRB), which represents the Laplacian pyramid levels. For each Laplacian pyramid level $O_{r_i} = f_{irb}(H), i = 0, ..., 10$, the locations of the tiny textures

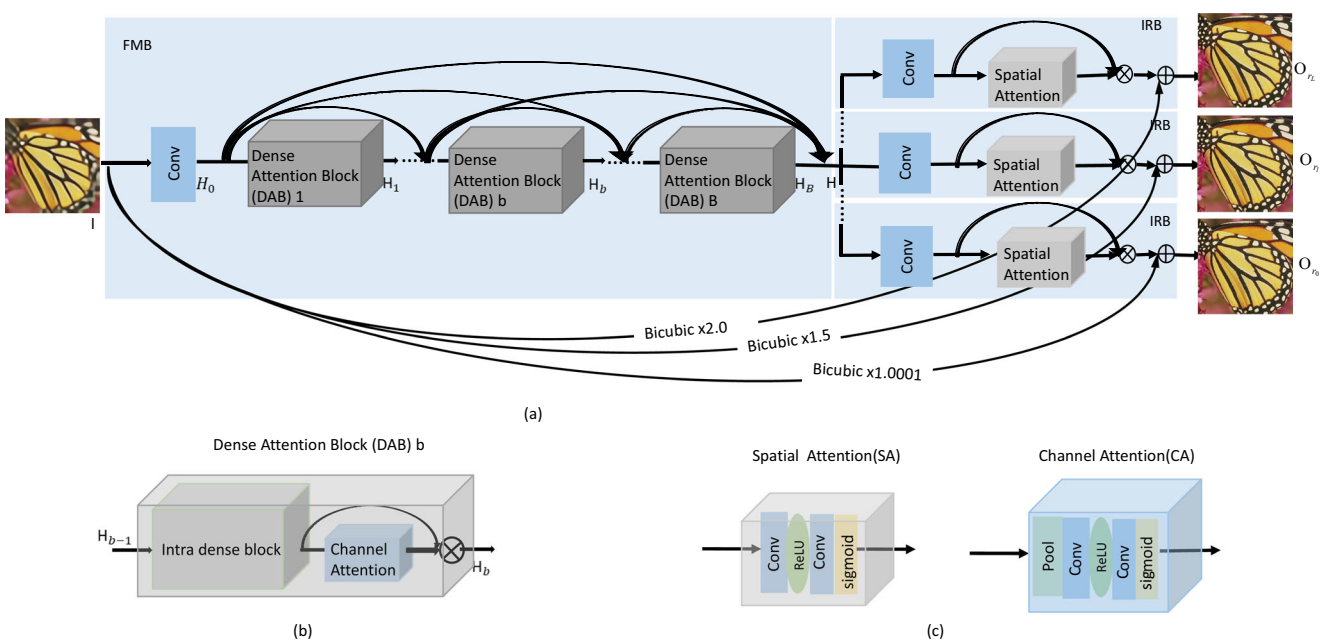


Fig. 3 **a** The overall architecture of the proposed ASDN network, multiple Image Reconstruction Branches (IRBs) parallelly allocate after Feature Mapping Branch (FMB). The FMB adopts the bi-dense structure from DBDN [25], and the spatial attention in IRB is the same spatial attention module from CSFM [9]. **b** The dense attention block (DAB) in (a), which combines the Intra dense block from DBDN [25] and channel attention module from RCAN [28]. **c** The illustration of the adopted spatial attention (SA) and channel attention (CA) modules from CSFM and RCAN [9, 28]

are different, and these textures usually contain high-frequency information, while the smooth areas have more low-frequency information. Therefore, to recover high-frequency details for image SR of different scales, it is helpful to mask out the discriminative high-frequency locations with spatial attention mechanism [16]. As shown in Fig. 3a, the learned high-level features are firstly restored into image space by a three-channel convolutional layer at each Laplacian pyramid Level. Then the restored image goes into the spatial attention (SA) [9] unit in Fig. 3c, to mask out the adaptive high-frequency information in the HR images of different scales. To preserve the smooth areas information and concentrate on training high-frequency information, the input interpolated LR images are added with the network output by identity skip connection (SC) to generate HR images.

To train the Any-scale SR Deep Network (ASDN) and generate Laplacian Frequency Representation, each IRB is randomly selected and combined after FMB at each update. For some practical applications where only require SR of specific scales, our ASDN can be fine-tuned to a fixed-scale network (FSDN) to further improve the reconstruction accuracy for the scales of interest by training image samples of specific scales. FSDN shares the same network structure as ASDN, except the deconvolutional layer of a specific scale, is inserted at the front of each IRB, which follows the common multi-scale single upscaling SR networks [15, 28, 29].

4 Experiments

In this section, we describe the implementation details of our models, including model hyper-parameters, training and testing details. Then we compare the proposed any-scale network and the fine-tuned fixed-scale model with several state-of-the-art SR methods on both fixed and any scale benchmark datasets including the quantitative, qualitative comparisons and any-scale comparisons. The effectiveness evaluation of the proposed any-scale method and the contribution study of different components in the proposed any-scale deep network are also provided in the paper.

4.1 Implementation details

Network settings In the proposed ASDN, all convolutional layers have 64 filters and 3×3 kernel size except the layers in IRB for restoring images and the convolutional layers in CA and SA units. The layers for image restoration have 3 filters and all the convolutional layers in CA and SA units are 1×1 kernel size, which adopt the same setting as CSFM [9]. Meanwhile, the 3×3 kernel size convolutional layer zero-pads the boundaries before applying convolution to keep the size of all feature maps the same as the input of each level. ASDN and FSDN share the same FMB structure, where 16 DAB are densely connected and each DAB has 8 dense layers. But in FSDN, the deconvolutional layer settings follow single upsampling networks [15, 29] to upscale feature mappings with the corresponding scales.

Training details The original training images are from DIV2K dataset [1] and Flicker dataset [1]. The input LR images for ASDN are bicubic interpolated from the training images with 11 decimal ratios r, which are evenly-distributed in the range of $(1, 2]$. In each training batch, 16 augmented RGB patches with the size of 48×48 are extracted from LR images as the input, and the LR images are randomly selected from one scale training samples among the total 11 scales training data. Here the data augmentation includes horizontal flips and 90-degree rotations are randomly adopted on each patch. To fine-tune the FSDN, the input LR images are downscaled by the scale factor among $2\times, 3\times, 4\times$, and $8\times$. In the training batch, a batch of 96×96 size patches is used as the targets and the corresponding scale LR RGB patches to optimize the specific scale modules. In general, ASDN and FSDN are all built with the platform Torch and optimized by Adam with L1 loss by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is initially set to 10^{-4} and halved at every 2×10^5 minibatch updates for 10^6 total minibatch updates.

Testing details Our proposed networks are tested on five widely-used benchmark datasets for image SR: Set5 [2], Set14 [26], BSD100 [22], Urban100 [11] and Manga109 [19]. To test any-scale network (ASDN) for SR of a random scale s, the testing images are first downscaled with the scale factor s as the LR images. If the scale s is not larger than 2, the LR images with scale s are upsampled and forwarded into the ASDN with the two enabled neighboring Laplacian pyramid levels of the scale s. HR images are predicted by interpolating these two levels based on Eq. 2. While if the scale s is larger than 2, the testing recursion times are based on $N = \lceil \log_2 R \rceil$. At each recursion n, the outputs of previous recursion are upscaled as input and deployed through ASDN with r_n according to the Eq. 6, except the initial recursion, which uses the LR images as input. To test fixed-scale network (FSDN), the testing input images are downscaled by the fixed scales s and deployed into the FSDN with the scale corresponding modules are enabled to yield the testing output.

4.2 Comparison with state-of-arts

To confirm the ability of the proposed methods, We first compare with state-of-the-art SR algorithms for qualitative

and quantitative analysis on the normal fixed scales 2×, 3×, 4×, 8×, which includes predefined upsampling methods (SRCNN [4], VDSR [12], DRRN [7], MemNet [21] and SRMDNF [27]), and single upsampling methods (RDN [29], LapSRN [13], EDSR [15], RCAN [28]).

4.2.1 Quantitative comparison

We compare the performance of our any-scale SR networks with state-of-the-art methods on the five challenging dataset benchmarks. Table 1 shows quantitative comparisons for

Table 1 Quantitative evaluation of state-of-the-art SR algorithms

scale	Algorithms	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2×	Bicubic	33.64	0.929	30.22	0.868	29.55	0.842	26.66	0.841	30.84	0.935
	SRCNN [4]	36.65	0.954	32.29	0.903	31.36	0.888	29.52	0.895	35.72	0.968
	VDSR [12]	37.53	0.958	32.97	0.913	31.90	0.896	30.77	0.914	37.16	0.974
	DRRN [7]	37.74	0.959	33.23	0.914	32.05	0.897	31.23	0.919	37.52	0.976
	LapSRN [13]	37.52	0.959	33.08	0.913	31.80	0.895	30.41	0.910	37.27	0.974
	MemNet [21]	37.78	0.959	33.28	0.914	32.08	0.898	31.33	0.919	37.72	0.974
	SRMDNF [27]	37.79	0.960	33.32	0.916	32.05	0.899	31.33	0.920	38.07	0.976
	ASDN(ours)	*38.12*	*0.961*	*33.82*	*0.919*	*32.30*	*0.901*	*32.47*	*0.931*	*39.16*	*0.978*
	EDSR [15]	38.11	0.960	33.92	0.920	32.32	0.901	32.93	0.935	39.10	0.976
	RDN [29]	38.24	0.961	34.01	0.921	32.34	0.902	32.96	0.936	39.19	0.978
	DBPN [6]	38.09	0.961	33.85	0.920	32.27	0.900	32.55	0.932	38.89	0.978
	RCAN [28]	38.27	0.961	34.12	0.922	32.41	0.903	**33.34**	**0.938**	39.44	0.979
	FSDN(ours)	**38.27**	**0.961**	**34.18**	**0.923**	**32.41**	**0.903**	33.13	0.937	**39.49**	**0.979**
3×	Bicubic	30.39	0.867	27.53	0.774	27.20	0.738	24.47	0.737	26.99	0.859
	SRCNN [4]	32.75	0.909	29.30	0.822	28.41	0.786	26.25	0.801	30.59	0.914
	VDSR [12]	33.66	0.921	29.77	0.831	28.82	0.798	27.41	0.830	32.01	0.934
	DRRN [7]	34.03	0.924	29.96	0.835	28.95	0.800	27.53	0.764	32.42	0.939
	LapSRN [13]	33.82	0.922	29.87	0.832	28.82	0.798	27.07	0.828	32.21	0.935
	MemNet [21]	34.09	0.925	30.00	0.835	28.96	0.800	27.57	0.839	32.51	0.937
	SRMDNF [27]	34.12	0.925	30.04	0.838	28.97	0.802	27.57	0.839	33.00	0.940
	ASDN(ours)	*34.48*	*0.928*	*30.35*	*0.843*	*29.18*	*0.808*	*28.45*	*0.858*	*33.87*	*0.947*
	EDSR [15]	34.65	0.928	30.52	0.846	29.25	0.809	28.80	0.865	34.17	0.948
	RDN [29]	34.71	0.929	30.57	0.847	29.26	0.809	28.80	0.865	34.13	0.948
	RCAN [28]	34.74	0.930	**30.65**	0.848	29.32	0.811	**29.09**	**0.870**	34.44	0.949
	FSDN(ours)	**34.75**	**0.930**	30.63	**0.848**	**29.33**	**0.811**	28.98	0.868	**34.53**	**0.950**
4×	Bicubic	28.42	0.810	26.10	0.704	25.96	0.669	23.15	0.660	24.92	0.789
	SRCNN [4]	30.49	0.862	27.61	0.754	26.91	0.712	24.53	0.724	27.66	0.858
	VDSR [12]	31.35	0.882	28.03	0.770	27.32	0.730	25.18	0.750	28.82	0.886
	DRRN [7]	31.68	0.889	28.21	0.772	27.38	0.728	25.44	0.764	29.18	0.891
	MemNet [21]	31.74	0.889	28.26	0.772	27.40	0.728	25.50	0.763	29.42	0.894
	SRMDNF [27]	31.96	0.892	28.35	0.778	27.49	0.734	25.68	0.773	30.09	0.902
	ASDN(ours)	*32.27*	*0.896*	*28.66*	*0.784*	*27.65*	*0.740*	*26.27*	*0.792*	*30.91*	*0.913*
	LapSRN [13]	31.54	0.885	28.19	0.772	27.32	0.727	25.21	0.756	29.46	0.890
	EDSR [15]	32.46	0.896	28.80	0.788	27.71	0.742	26.64	0.803	31.02	0.915
	RDN [29]	32.47	0.899	28.81	0.787	27.72	0.742	26.61	0.803	31.00	0.915
	DBPN [6]	32.42	0.898	28.76	0.786	27.68	0.740	26.38	0.796	30.91	0.914
	RCAN [28]	32.63	0.900	28.87	0.789	27.77	0.744	**26.82**	**0.809**	31.22	0.917
	FSDN(ours)	**32.63**	**0.900**	**28.89**	**0.789**	**27.79**	**0.744**	26.79	0.807	**31.44**	**0.919**

Table 1 (continued)

scale	Algorithms	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
8×	Bicubic	24.40	0.658	23.10	0.566	23.97	0.548	20.74	0.516	21.47	0.650
	SRCNN [4]	25.33	0.690	23.76	0.591	24.13	0.566	21.29	0.544	22.46	0.695
	VDSR [12]	25.93	0.724	24.26	0.614	24.49	0.583	21.70	0.571	23.16	0.725
	LapSRN [13]	26.15	0.738	24.35	0.620	24.54	0.586	21.81	0.581	23.39	0.735
	MemNet [21]	26.16	0.741	24.38	0.619	24.58	0.584	21.89	0.583	23.56	0.738
	ASDN(ours)	*27.02*	*0.776*	*24.99*	*0.641*	*24.82*	*0.600*	*22.57*	*0.620*	*24.73*	*0.748*
	EDSR [15]	26.96	0.776	24.91	0.642	24.81	0.599	22.51	0.622	24.69	0.784
	DBPN [6]	27.21	0.784	25.13	0.648	24.88	0.601	22.73	0.631	25.14	0.799
	RCAN [28]	27.31	0.788	25.23	0.651	24.98	**0.606**	**23.00**	**0.645**	25.24	0.803
	FSDN(ours)	**27.33**	**0.789**	**25.24**	**0.651**	**24.98**	0.604	22.90	0.638	**25.24**	**0.803**

We report the average PSNR/SSIM for 2×, 3×, 4× and 8× SR. Bold indicates the best performance, and italic indicates the best performance among predefined upsampling methods

×2, ×3, ×4, ×8 SR. For fair comparisons with the recent single upsampling networks, we fine-tune the ASDN with the fixed ×2, ×3, ×4, ×8 scale SR samples as FSDN for reference. It is obvious that FSDN has better performance than state-of-the-art methods, except RCAN on some datasets. Although on Urban100, which is consisted of straight-line building structure images, RCAN has better performance than FSDN due to the more channel attentions across the network, which is sensitive to the sharp edges in the image reconstruction. On other datasets, FSDN reconstruction accuracy is comparable to RCAN. This indicates the network, which is the same main framework as ASDN is effective to learn mapping functions for SR tasks.

Due to the strong ability of the framework, our ASDN performs favorably against the existing methods, especially compared to the predefined upsampling methods. Noted that ASDN does not use any 3×, 4×, 8× SR samples for training but still generates comparable results as EDSR. There are mainly two reasons for ASDN drops behind some upsampling models. First, these upsampling models are trained with fixed-scale SR samples, and customized for the 2×, 3×, 4×, and 8× scales deployments, but ASDN is trained with scales in $(1, 2]$. Second, the upsampling layers [20] can improve the reconstruction performance, as shown in our experiment, FSDN (the upsampling version of ASDN) has more than $0.1dB$ PSNR compared to ASDN on scale 2×. However, some of the upsampling layers can only apply for SR of the integer scales [20], such as transposed layers. Although, Meta-upsampling [8] layer can upscale images with decimal scales, these scale factors need to be trained before deployment. Therefore, we compromise some reconstruction accuracy for the continuous scale SR using the predefined upsampling structure, which only requires to be trained with several representative scales. Our ASDN is still very profound on the normal fixed scales compared with the existing predefined upsampling deep methods. Regarding the speed, our ASDN takes 0.5 seconds to process a 288×288 image for 2× SR on a Titan X GPU, and FSDN takes about 0.04 seconds to generate a 288×288 image for 2× SR.

4.2.2 Any-scale comparison

In this section, in order to evaluate the efficiency of our ASDN for any upscale ratio SR, we firstly compare ASDN with other methods. The Bicubic interpolation method is adopted as the reference, and some deep learning network frameworks (EDSR, RDN, VDSR) are retrained with the proposed any-scale SR method and the same training data as our ASDN for any-scale SR comparison denoted as EDSR-Conv, RDN-Conv, and VDSR-Conv. Meta-EDSR and Meta-RDN [8] are dynamic meta-upsampling models which are trained with scale factors from ×1 to ×4 at the stride of 0.1.

The experimental results are shown in Table 2, which uses the PSNR value for comparison. It shows the PSNR value on SR of 9 trained scales from ×1.1 to ×1.9 and it is obvious that our ASDN reaches the state-of-the-art performance. Then it also illustrates ASDN efficiency on the scales not trained before and evaluates the effective scale range of our proposed any-scale SR network. For SR of scales out of the range, ASDN is comparable to Meta-EDSR, but slightly drops behind Meta-RDN. This is due to ASDN is the recursively deployed results, and Meta-RDN is customized with these scales. Although the recursively

Table 2 Results of any-scale SR on different methods tested on BSD100

Method \ Scale	X1.1	X1.2	X1.3	X1.4	X1.5	X1.6	X1.7	X1.8	X1.9	X2.0	X2.8	X4.0	X5.7	X8.0	X11.3	X16.0	X22.6	X32.0
Bicubic	36.56	35.01	33.84	32.93	32.14	31.49	30.90	30.38	29.97	29.55	27.53	25.96	24.96	23.67	22.65	21.73	20.73	19.90
VDSR-Conv	42.13	39.52	37.88	36.53	35.42	34.50	33.72	33.03	32.41	31.89	29.23	27.25	25.56	24.58	23.49	22.47	21.39	20.38
EDSR-Conv	*42.92*	40.11	*38.33*	36.93	35.79	34.85	34.06	33.38	32.75	32.23	29.54	27.58	*26.01*	*24.78*	*23.65*	*22.63*	*21.55*	*20.53*
RDN-Conv	42.86	40.04	38.25	36.86	35.72	34.78	33.99	33.29	32.67	32.07	29.47	27.51	25.94	24.72	23.60	22.58	21.50	20.51
Meta-EDSR [8]	42.72	39.92	38.16	36.84	35.78	34.83	34.06	33.36	*32.78*	32.26	29.61	*27.67*	–	–	–	–	–	–
Meta-RDN [8]	42.82	**40.40**	38.28	*36.95*	*35.86*	*34.90*	*34.13*	*33.45*	**32.86**	**32.35**	**29.67**	**27.75**	–	–	–	–	–	–
ASDN(ours)	**43.05**	*40.24*	**38.42**	**37.02**	**35.87**	**34.92**	**34.14**	**33.46**	**32.86**	*32.30*	*29.63*	27.65	**26.07**	**24.85**	**23.70**	**22.66**	**21.59**	**20.55**

PSNR values on SR of trained scales from 1.1 to 2 shows the results of Laplacian pyramid levels and PSNR values on SR of scales greater than 2 demonstrates SR performance on randomly selected scales and boundary condition. Bold indicates the best performance, and italic indicates the second best performance

deployed SR results have slight drop back as the directly deployed results, recursive deployment can still effectively generate SR of scales not trained before. Through this way, ASDN only needs 11 training scales for any-scale SR.

Figure 4 shows the any-scale SR results on a continuous scale range. We test our any-scale network performance with random decimal scales distributed in the commonly used range of $\times 2$ to $\times 8$ on Set5 and plot out the results into the line. It is proved that ASDN and the models trained with our any-scale SR method can effectively reconstruct HR images of continuous upscale ratios. Our ASDN outperforms all the other methods, which is generally 0.15 dB better than EDSR-Conv, outperforms VDSR-Conv by 0.6 dB and robustly keeps the deference of more than 3 dB PSNR from Bicubic method in the continuous scale range. The result demonstrates our ASDN can effectively reconstruct HR images of continuous upscale ratios and our any-scale training method is flexible to many deep CNN networks.

4.2.3 Qualitative comparison

We show visual comparisons on the testing datasets for $2\times$, $4\times$ and $8\times$ SR. For $2\times$ enlargement of Set14 in Fig. 5, FSDN suppresses the bias artifacts and recovers the cloth pattern and text closer to the ground truths than all the other methods. Meanwhile, ASDN tends to construct less biased images than other methods. For $4\times$ enlargement of the parallel straight lines in Fig. 6. Our methods generate a clearer building line, while other methods suffer the blurring artifacts. RCAN tends to generate misleading strong edges due to the more channel attention structure, but our ASDN and FSDN generates soft patterns closer to the ground truth. The reconstruction performance on $8\times$ SR is further analyzed in Fig. 7. FSDN restores the sharper characters than the compared networks and ASDN is able to recover more accurate textures from the distorted LR image than many other fixed-scale methods.

4.3 Study of any-scale methods

We study the effects of Laplacian Frequency Representation and Recursive Deployment of the any-scale SR methods.

4.3.1 Laplacian frequency representation

To evaluate the accuracy of the Laplacian Frequency Representation for continuous scale SR. We compare the reconstruction results of the Laplacian Frequency Representation with the directly deployed HR images of 100 scales in the range (1, 2].

We first modify EDSR, RDN, and ASDN frameworks into the single predefined upsampling networks and train them with these 100 scales SR samples as EDSR-100, RDN-100

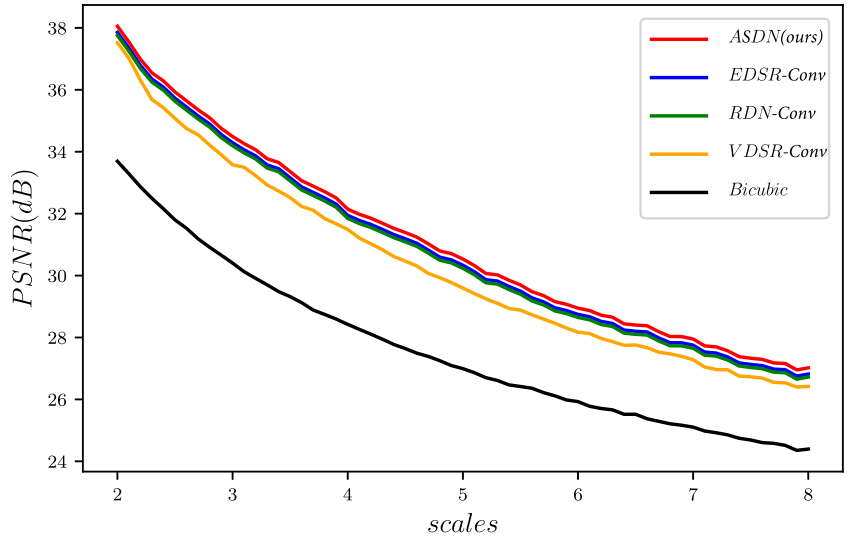


Fig. 4 PSNR comparison of ASDN with other works within the continuous scale range $(\times 2, \times 8]$ on Set5

and ASDN-100 to generate HR images of Set5 on the 100 scales. Then we reconstruct the single redefined upsampling EDSR-100 and RDN-100 with 11 parallel IRBs as EDSR-Conv and RDN-Conv, as suggested in Section 4.2.2, trained with the same method and data as ASDN. As shown in Fig. 8a, It is obvious that the Laplacian frequency represented HR images have a similar quality to the direct deployed HR images.

To analyze the influence of the Laplacian pyramid level density on the SR performance, we train ASDN on $5, 9, 17$ evenly distributed upscale decimal ratios in $(1, 2]$ with DIV2K, which separates the Laplacian Frequency Representation into $4, 8$ and 16 phases and names ASDN-4, ASDN-8, and ASDN-16 separately. Figure 8b demonstrates the performance of the three versions of ASDN with scales in $(1, 2]$. In order to make the difference more obvious, we choose some scale ranges in $(1, 2]$. It illustrates that ASDN-4 drops behind ASDN-8 and ASDN-16 commonly, and ASDN-8 and ASDN-16 almost overlap. The results show the Laplacian pyramid level density influences SR performance. To some extent, the model trained with more dense scales achieves better performance, but it saturates beyond a certain point, such as 10 phases. Due to this reason, we can

Fig. 5 Qualitative comparisons of our models with other works on $\times 2$ super-resolution. Red indicates the best performance, and blue indicates the second best

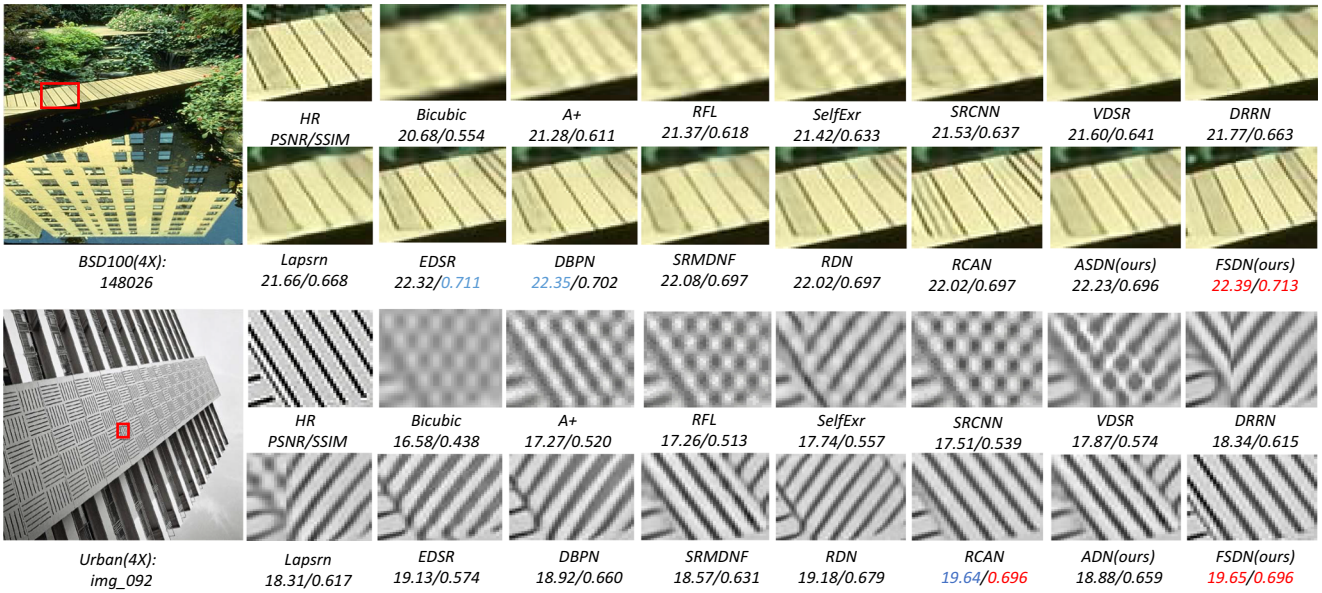


Fig. 6 Qualitative comparisons of our models with other works on ×4 super-resolution. Red indicates the best performance, and blue indicates the second best

generate HR images of any decimal scale in the range of $(1, 2]$ by the several Laplacian pyramid levels in $(1, 2]$.

4.3.2 Recursive deployment

In order to investigate the effects of recursive deployment for HR images of larger decimal scales. We mainly demonstrate the comparison of recursive deployment and direct deployment on scales ×2, ×3, ×4 We trained VDSR-Conv, EDSR-Conv, RDN-Conv, and ASDN with 11 evenly distributed upscale decimal ratios in $(1, 2]$ as the recursive models and the HR images are twice upscaled with the upscale ratios $\times\sqrt{2}$, $\times\sqrt{3}$, $\times\sqrt{4}$. To form the fair comparisons, we trained VDSR-Conv, EDSR-Conv, RDN-Conv, and ASDN with ×2, ×3, ×4 SR images as the direct deployment models. Table 3 illustrates the PSNR of recursive deployment and direct deployment. It is

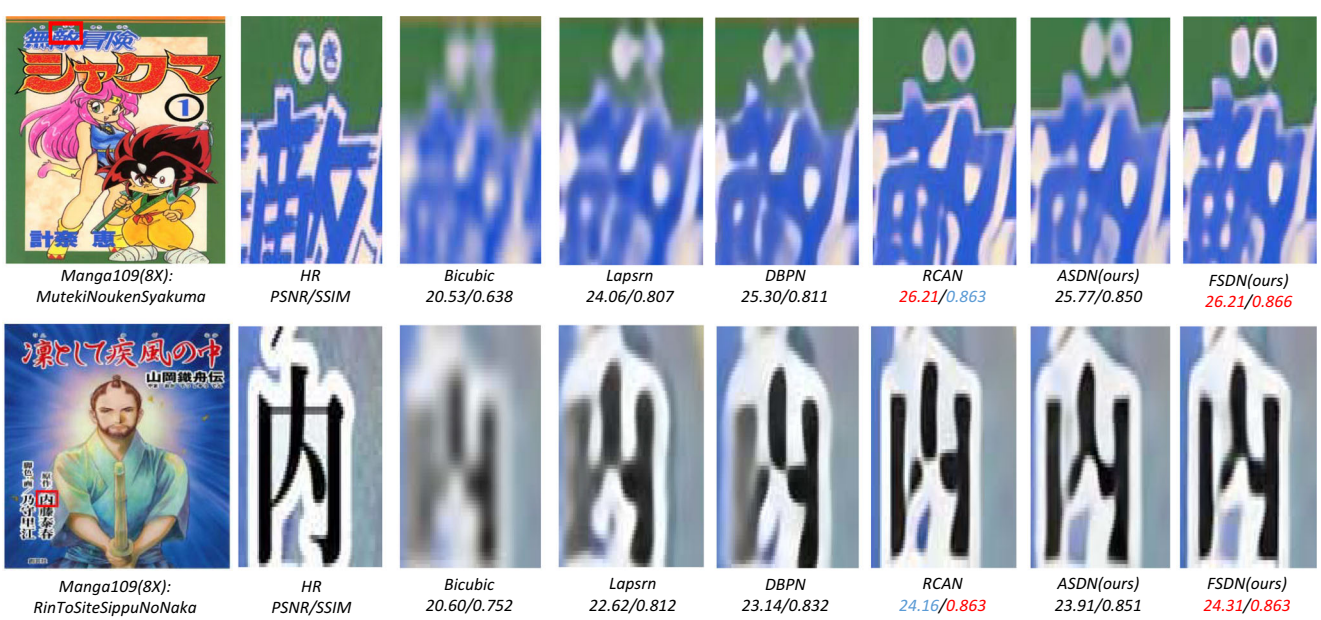
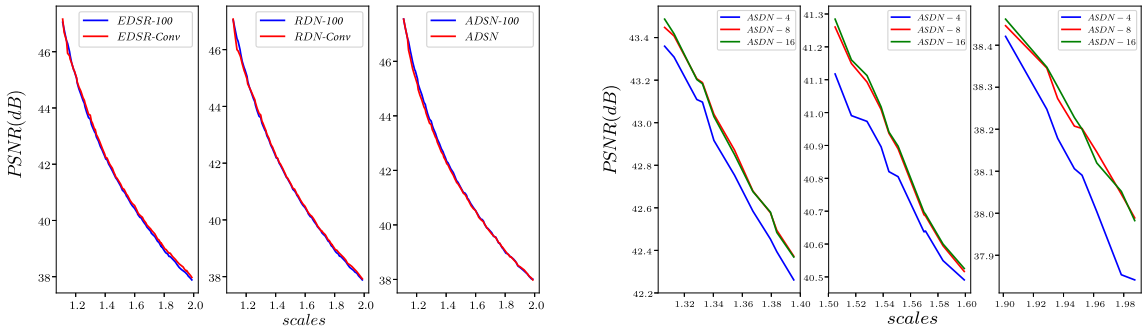


Fig. 7 Qualitative comparisons of our models with other works on ×8 super-resolution. Red indicates the best performance, and blue indicates the second best



(a) (b)

Fig. 8 Study of Laplacian Frequency Representation

Table 3 PSNR of the recursive deployment and direct deployment on SR for ×2, ×3, ×4

Methods	Direct deployment			Recursive deployment		
	×2	×3	×4	×2	×3	×4
VDSR-Conv	37.57	33.77	31.56	36.86	33.70	31.50
EDSR-Conv	38.04	34.45	32.29	37.18	34.32	32.26
RDN-Conv	38.05	34.46	32.31	37.27	34.38	32.23
Ours	38.12	34.52	32.28	37.35	34.43	32.27

Table 4 PSNR of recursive deployment and direct deployment on SR for ×2, ×3, ×4. **Black** indicates the best performance

Scale(R)	Recursion(N)	UpscaleRatio(r)	PSNR
3×	2	1.732, 1.732	34.43
		1.500, 2.000	34.19
		2.000, 1.500	**34.48**
	3	1.442, 1.442, 1.442	33.18
4×	2	2.000, 2.000	**32.27**
		1.587, 1.587, 1.587	31.96
	3	1.800, 1.800, 1.234	32.16
		2.000, 1.800, 1.100	32.24

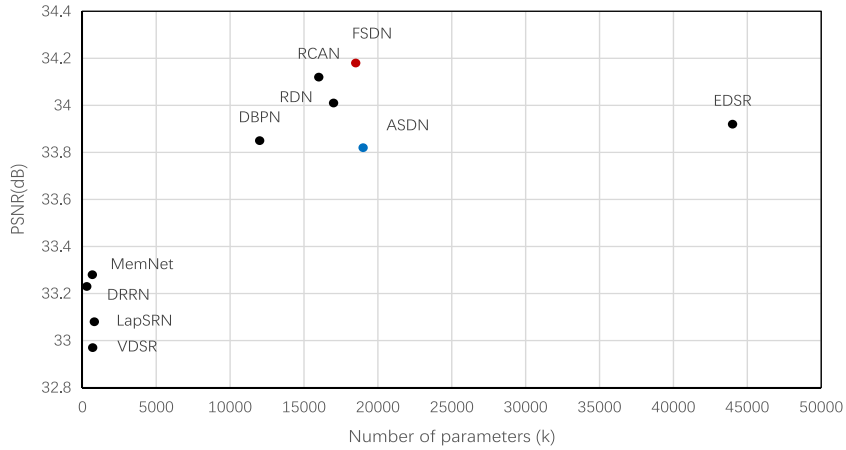


Fig. 9 Performance vs number of parameters. The results are evaluated with Set14 for 2× enlargement. Red indicates the best performance, and blue indicates the best performance among predefined upsampling methods

Table 5 Investigation of channel attention (CA), spatial attention (SA), and skip connection (SC). **Black** indicates the best performance

Module	Different combination of CA, SA and SC							
CA	×	×	×	√	√	√	×	√
SA	×	×	√	×	√	×	√	√
SC	×	√	×	×	×	√	√	√
PSNR	37.92	37.96	37.93	37.95	37.97	37.99	37.97	**38.01**

obvious that recursive deployment generally leads to the SR performance decline compared to the direct deployment. But the difference between recursive deployment and direct deployment goes down as the scale goes up. Since the decline is still in an acceptable range and goes gentle as the upscale ratios up, we adopt recursive deployment for SR in higher upscale ratio ranges.

To determine the best solution of recursive times N and upscale ratios r for recursive deployment. We also explore various combinations of N and r to deploy any-scale HR images with different strategies. Table 4 illustrates the performance of the HR images deployed by different strategies with ASDN on Set5. It is obvious that the larger upscale ratio r combined with, the smaller recursive time N will contribute to better performance. Furthermore, choosing the larger upscale ratios in the early recursions can produce better results than using the smaller scales. For these reasons, we recommend choosing $N = \lceil \log_2 R \rceil$ with the largest upscale ratios $r = 2$ at the early $N - 1_{th}$ recursions for large scale SR.

4.4 Model analysis

4.4.1 Number of parameters

To demonstrate the compactness of our model, we compare the model performance and network parameters of our model with the existing deep networks for image SR in Fig. 9. Our model shows the trade-off between the parameter demands and performance. Since VDSR, DRRN, LapSRN, and MemNet are all light version networks, they all visibly concede the performance for the model parameter numbers. Therefore ASDN outperforms all the other predefined upsampling methods over 0.5 dB on Set14 for 2× enlargement. Furthermore, FSDN achieves the best results with a moderate number of parameters compared to all the other upsampling methods.

4.4.2 Abviation study

In this section, we evaluate the influence of different network modules, such as channel attention (CA) in FMB, spatial attention (SA) in IRB, and skip connection (SC) between input and output. To demonstrate the effect of CA in the proposed network structure, we remove the CA from the FMB. In Table 5, we can see when CA is removed, the PSNR value on Set5 (×2) is relatively low compared to the model having CA. To investigate the effect of SA, we remove the SA from the ASDN to compare with the network with SA. SA can improve performance by 0.02 dB or 0.01 dB with or without CA in the models. We further investigate the contribution of SC to the network by comparing the models with or without SC. Adding global skip connections between the network input and output generally improves 0.04 dB on Set5. Generally combining attention modules into the network design, helps the residual high-frequency information reconstruction.

5 Conclusion

In this paper, we propose an any-scale deep network (ASDN) to generate HR images of any scale with one unified network by adopting our proposed any-scale SR method, including Laplacian Frequency Representation for SR of small continuous scale ranges and Recursive Deployment for larger-scale SR. The any-scale SR method helps to reduce the demands of training scale samples and accelerate the network convergence. The extensive comparisons show our ASDN is superior to the most state-of-the-art methods on both fixed-scale and any-scale benchmarks.

References

1. Agustsson E, Timofte R (2017) Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE conference on computer vision and pattern recognition (CVPR) workshops, vol 3, p 2
2. Bevilacqua M, Roumy A, Guillemot C, Alberi-morel ML (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding
3. Burt PJ, Adelson EH (1987) The laplacian pyramid as a compact image code. In: Readings in computer vision, pp 671–679. Elsevier
4. Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: European conference on computer vision, pp 184–199. Springer

5. Gao G, Zhu D, Yang M, Lu H, Yang W, Gao H (2018) Face image super-resolution with pose via nuclear norm regularized structural orthogonal procrustes regression. Neural Comput & Applic, 1–11
6. Haris M, Shakhnarovich G, Ukita N (2018) Deep backprojection networks for super-resolution. In: Conference on computer vision and pattern recognition
7. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
8. Hu X, Mu H, Zhang X, Wang Z, Tan T, Sun J (2019) Meta-sr: a magnification-arbitrary network for super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1575–1584
9. Hu Y, Li J, Huang Y, Gao X (2019) Channel-wise and spatial feature modulation network for single image super-resolution. IEEE Transactions on Circuits and Systems for Video Technology
10. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: European conference on computer vision, pp 646–661. Springer
11. Huang JB, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 5197–5206
12. Kim J, Kwon Lee J, Mu Lee K (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1646–1654
13. Lai WS, Huang JB, Ahuja N, Yang MH (2017) Deep laplacian pyramid networks for fast and accurate superresolution. In: IEEE Conference on computer vision and pattern recognition, vol 2, p 5
14. Lai WS, Huang JB, Ahuja N, Yang MH (2018) Fast and accurate image super-resolution with deep laplacian pyramid networks IEEE transactions on pattern analysis and machine intelligence
15. Lim B, Son S, Kim H, Nah S, Lee KM (2017) Enhanced deep residual networks for single image super-resolution. In: The IEEE conference on computer vision and pattern recognition (CVPR) workshops, vol 1, p 4
16. Liu Y, Wang Y, Li N, Cheng X, Zhang Y, Huang Y, Lu G (2018) An attention-based approach for single image super resolution. In: 2018 24Th international conference on pattern recognition (ICPR), pp 2777–2784. IEEE
17. Lu H, Li Y, Nakashima S, Kim H, Serikawa S (2017) Underwater image super-resolution by descattering and fusion. IEEE Access 5:670–679
18. Lu T, Wang J, Zhang Y, Wang Z, Jiang J (2019) Satellite image super-resolution via multi-scale residual deep neural network. Remote Sens 11(13):1588
19. Matsui Y, Ito K, Aramaki Y, Fujimoto A, Ogawa T, Yamasaki T, Aizawa K (2017) Sketch-based manga retrieval using manga109 dataset. Multimed Tools Appl 76(20):21811–21838
20. Schulter S, Leistner C, Bischof H (2015) Fast and accurate image upscaling with super-resolution forests. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3791–3799
21. Tai Y, Yang J, Liu X, Xu C (2017) Memnet: a persistent memory network for image restoration. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4539–4547
22. Timofte R, De Smet V, Van Gool L (2014) A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Asian conference on computer vision, pp 111–126. Springer
23. Tong T, Li G, Liu X, Gao Q (2017) Image super-resolution using dense skip connections. In: Computer vision (ICCV), 2017 IEEE international conference on, pp 4809–4817. IEEE
24. Wang D, Lu H, Yang MH (2015) Robust visual tracking via least soft-threshold squares. IEEE Trans Circuits Sys Vid Techn 26(9):1709–1721
25. Wang Y, Shen J, Zhang J (2018) Deep bi-dense networks for image super-resolution. In: 2018 Digital image computing: techniques and applications (DICTA), pp 1–8. IEEE
26. Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. International conference on curves and surfaces. 711–730
27. Zhang K, Zuo W, Zhang L (2018) Learning a single convolutional super-resolution network for multiple degradations. In: IEEE Conference on computer vision and pattern recognition, vol 6
28. Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y (2018) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 286–301
29. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y (2018) Residual dense network for image super-resolution. In: The IEEE conference on computer vision and pattern recognition (CVPR)