Single Image Super-Resolution with Hierarchical Receptive Field

Din Qin
Department of Electronic Engineering
Fudan University
Shanghai, 200433, China
dqin18@fudan.edu.cn

Xiaodong Gu
Department of Electronic Engineering
Fudan University
Shanghai, 200433, China
xdgu@fudan.edu.cn

***Abstract*—As a pixel-level prediction task, it's crucial for single image super-resolution (SISR) to capture contextual information over the multi-scale regions in low-resolution (LR) space, which is used to predict the image details in high-resolution (HR) space. So researchers proposed multiple methods to enhance the size of receptive field and take the contextual information of images into account. But most of them tend to increase the depth of networks or the size of kernels simply, which is inefficient and consumes a large amount of computational resources and memory. In this paper, we combine dilated convolutions with standard convolutions and propose hierarchical receptive field network (HRFN) to enlarge receptive field without additional computational complexity. Specially, in each hierarchical receptive field block (HRFB), we apply standard convolutions with different kernel sizes and dilated convolutions with different dilation factors to adaptively obtain multi-scale features. Meanwhile, to ease the training process and make the model focus on the prediction of image details (high-frequency features), the residual learning is adopted locally and globally to explicitly learn and predict the difference between HR images and LR images. Finally, experimental results on five extensively used datasets show that our model outperforms those state-of-the-art methods for both quantitative and qualitative comparisons.**

***Index Terms*—single image super-resolution, dilated convolution, receptive field, residual learning**

I. Introduction

Single image super-resolution (SISR) aims to reconstruct a high-resolution image I^{SR} with better visual performance and refined details from a given low-resolution image I^{LR} with coarse details. As a low-level computer vision task, SISR has attracted a lot of researchers' attention because of its extensive applications, such as object detection in scenes [1], medical image [2] and remote sensing [3]. However, single image super-resolution (SISR) is an ill-posed problem and there are multiple solutions for the same low-resolution (LR) image, which means the same LR image may correspond to diverse high-resolution (HR) images. So super-resolution (SR) is still considered a challenge in computer vision. Among plenty of algorithms, learning-based methods [4]–[6] are widely adopted to learn the mapping between LR and HR images.

With the successful applications of convolutional neural network (CNN) in computer vision, a variety of CNN-based algorithms have been proposed to solve this problem. Dong et al. [4] firstly proposed three-layer convolutional neural network named SRCNN to learn the nonlinear mapping between LR and HR images and achieved significant improvements. Inspired by SRCNN, Kim et al. [5] adopted 20 convolutional layers and residual learning in their model (VDSR) which used large receptive field to capture large context of an image. Meanwhile, they also explored a deeply-recursive convolutional network (DRCN) [8] to enlarge the receptive field, which demonstrated larger receptive fields lead to better performance. With the encouragement of the ResNet [9], the structures of SR models become deeper and deeper. Ledig et al. [11] stacked 16 residual blocks and used global residual learning to alleviate gradient vanishing/exploding. Tong et al. [6] introduced dense connections into SISR and proposed a 68-layer network called SRDenseNet, which achieved more efficient use of features between layers.

The main concern of SISR is how to reconstruct as much high-frequency details as possible. It is necessary for us to dig for more information from the input LR image because it's the only clue we have. On the one hand, contextual information in an image highly influences reconstruction performance for pixel-level predictions. When our network can extract richer contextual details by a large receptive field, it's sure that the network will infer missing high-frequency information more accurately and achieve better performance as demonstrated in VDSR [5] and DRCN [8]. However, if simply increasing the layer depth or the kernel size, additional parameters and computational complexity will also increase in response. On the other hand, different regions contain different textures, shapes and sizes. It is obviously unfair if we treat various regions with the receptive fields which have the same size. However, most existing methods like [5], [6], [11] ignore the difference between regions and adopt fixed-size receptive field with a single kernel size such as 3×3 kernel.

Based on above analysis, we propose a novel convolutional neural network named hierarchical receptive field network (HRFN) and design the hierarchical receptive field block (HRFB) using multi-branch architecture with standard and dilated convolutions. Dilated convolutions [12] are proposed to solve dense prediction problems, which support expansion of the receptive field without parameters and computational

This work was supported in part by National Natural Science Foundation of China under grants 61771145 and 61371148.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

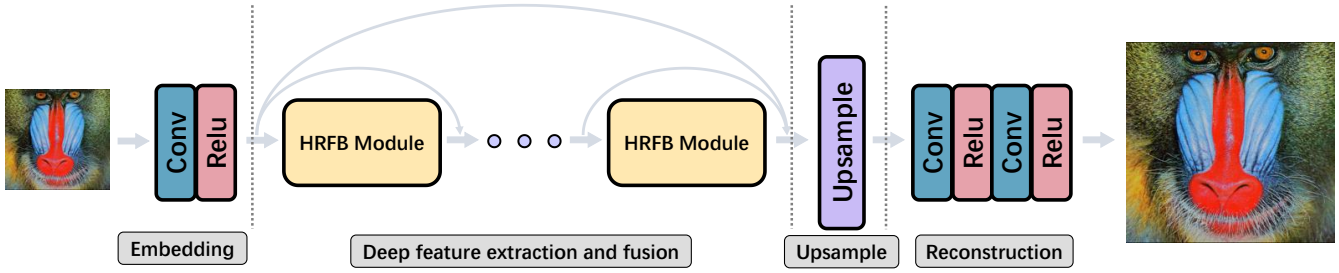


Fig. 1. The architecture of hierarchical receptive field network (HRFN). HRFN consists of four parts: embedding module, deep feature extraction and fusion module, upsampling module and reconstruction module.

complexity increase. Because of blind spots in dilated convolutions, we combine standard and dilated convolutions together to avoid the loss of information. Then, three-branch structure is applied to capture multi-scale features. These features contain small, median and large three-level information. Through HRFB, our network can extract richer contextual information in varying scales and finally achieve notable reconstruction performance. To ease the training process and alleviate gradients vanishing/exploding, we use global and local residual learning simultaneously. This strategy enables our model to concentrate on the high-frequency information. In summary, our main contributions are as follow:

(1) We propose a hierarchical receptive field network (HRFN) and our HRFN outperforms those state-of-the-art methods on both quantitative results and visual performance, which demonstrates it's vital to extract and fuse multi-scale features for SISR.

(2) We combine standard convolutions and dilated convolutions together to expand receptive field efficiently without additional parameters and computational complexity. The mixed layer enables our model to utilize more contextual information for inferring lost high-frequency details.

(3) We design a multi-branch block to capture and fuse multi-scale features, which is suitable for regions of different sizes in an image.

II. Related Work

Numerous SISR methods have been proposed including interpolation-based [13], reconstruction-based [14], learning-based [4]–[6] methods and so on. Here, we only focus on the CNN-based methods.

Dong et al. [4] proposed SRCNN to learn the nonlinear mapping from the LR image to the HR image, which is the ground breaking work in SISR with deep learning. From then on, CNN has been the mainstream choice and these methods have achieved notable improvement. Then to enlarge the receptive field and ease the training, Kim et al. [5], [8] introduced residual learning and recursive structure into their VDSR and DRCN respectively, which verified the effectiveness of large receptive fields. Inspired by Kim et al. [5], [8], Tai et al. [15] constructed a deeper recursive residual network (DRRN) with 52 convolutional layers to acquire performance improvement. However, these networks need to firstly upsample the input LR image to the target size via bicubic interpolation, which inevitably increases the computational complexity and produces visible artifacts [16].

To avoid this problem, Dong et al. [7] introduced a deconvolution layer at the end of the model and the model can directly learn the mapping from LR image to HR image without interpolation. Shi et al. [16] proposed a novel subpixel layer in their ESPCN and the interpolation function is implicitly contained in previous convolutional layers, which can be learned automatically. From then on, post-upsampling becomes the main choice. More recently, inspired by DenseNet [10], Tong et al. [6] applied dense connections in their SRDenseNet and the features in different levels are fused through skip connections to enhance the reconstruction performance. Different from those methods optimized by L1 or L2 loss, Ledig et al. [11] employed perceptual loss and adversarial loss to generate photo-realistic images in the framework of generative adversarial network. Though these produced images have lower PSNR (peak signal-to-noise ratio) values, they are more in line with human visual perception. Zhang et al. [17] proposed a novel residual dense network which uses densely connected convolutional layers to sufficiently extract local features. Then these local dense features are fused globally to adaptively learn global hierarchical features.

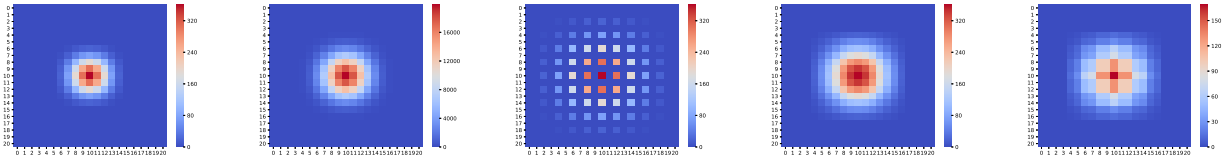
These methods achieve significant performance, but most of them ignore the features at different scales or expand receptive field in an inefficient manner. The features in different scales are supposed to be disposed with the receptive fields in different sizes as done in inception module [34]. Therefore, we propose a hierarchical receptive field network (HRFN).

III. Proposed Method

In this section, we will detail our model, including the design of hierarchical receptive field block (HRFB), the architecture of our hierarchical receptive field network (HRFN) and the loss function.

A. Hierarchical Receptive Field Block

The ultimate aim of SISR is to reconstruct as many high-frequency details as possible. When reconstructing high-frequency details in an image, different regions contain dif-



(a) k3d1_×4 (b) k3d1_×6 (c) k3d2_×4 (d) k5d1_×3 (e) k3d1k3d2_×2

Fig. 2. Comparisons of receptive field intensity maps. Receptive field intensity maps denote the times that the information have been captured in each position. The more times, the color gets closer to red. (c) k3d2_×4 indicates blind spots will incur information loss if we only adopt dilated convolutions. (e) k3d1k3d2_×2 could achieve the receptive filed with the same size as (d) k5d1_×3 but use less parameters.

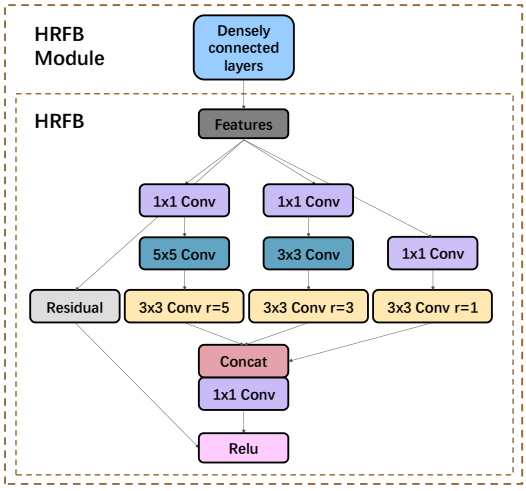


Fig. 3. The architecture of HRFB module. Every HRFB module contains densely connected layers and a hierarchical receptive field block (HRFB). 1×1, 3×3 and 5×5 denote the kernel size of convolutional layer. r represents the dilation rate of dilated convolutions.

ferent information like shapes and textures and these regions also vary in sizes. So it's crucial to treat different regions differently. However, many state-of-the-art methods only extract all features with fixed-size receptive fields and ignore the fact that different features are in different scales. To avoid this drawback, we design a hierarchical receptive field block (HRFB) to achieve multi-scale receptive fields and extract multi-scale features from various receptive fields for learning different scale mappings. At the same time, Kim et al. [5], [8] found the size of receptive field is of great importance because a large receptive field could provide the network with more contextual information to reconstruct high-frequency details in an image. Though some methods are proposed to explore receptive fields for image SR, most of them only focus on the model depth or the kernel size, which is certain to greatly increase computational complexity and parameters. To achieve the balance between large receptive field and computational cost, we combine standard convolutions and dilated convolutions together within HRFB, which efficiently expands the receptive field.

As shown in Fig. 3, the HRFB module contains densely connected convolutional layers for preliminary feature extraction, a skip connection for local residual learning and three different branches with different receptive fields to suit for details and textures in different scales. Each branch can be divided in four parts: 1×1 convolutional layer for dimension reduction, standard convolutional layers with various kernels, 3×3 dilated convolutional layers with corresponding dilation factors and concatenation.

Next, we take the second branch for example to detail the HRFB. A 1×1 convolutional layer is first applied to reduce the channel number of feature maps. Then these features are forwarded into a 3×3 standard convolutional layer to capture the information in a specific scale. To enlarge receptive field and capture more information at a large area without the addition of parameters, a 3×3 dilated convolution layer with a corresponding dilation factor $r = 3$ is followed. The operations can be formulated as,

$$F_{n-1}^2 = \mathcal{C}_{3\times3}^{d,r=3}(\mathcal{C}_{3\times3}^{s}(\mathcal{C}_{1\times1}^{s}(F_{n-1}))) \tag{1}$$

where $\mathcal{C}_{i\times i}^{s}$ denotes standard convolutional operation with $i \times i$ kernel and $\mathcal{C}_{3\times3}^{d,r=k}$ is dilated convolutional operation where kernel size is 3×3 and dilation factor is k. F_{n-1} represents the input of the n-th HRFB and F_{n-1}^2 is the output of the second branch.

Next, the features of this branch are concatenated with other branch features for subsequent operations.

$$F_{concat} = \mathcal{C}_{1\times1}^{s}([F_{n-1}^1, F_{n-1}^2, F_{n-1}^3]) \tag{2}$$

where F_{n-1}^i denotes the output of the i-th branch and $[\cdot]$ is concatenation operation. To ensure that the dimension of F_{n-1}^i is the same as F_{n-1} in each branch, we apply a 1×1 convolutional layer again after concatenation, which increases the channel number back to the start.

Ultimately, we adopt the local residual learning inspired by ResNet [9] to make main branches focus on the residual features and alleviate gradients vanishing. Now we can obtain the final output of the HRFB.

$$F_n = Relu(F_{concat} + F_{n-1}) \tag{3}$$

where F_n is the output of the n-th HRFB module.

B. Network Architecture

As discussed above, the size of receptive field and multi-scale contextual information are crucial for predicting image

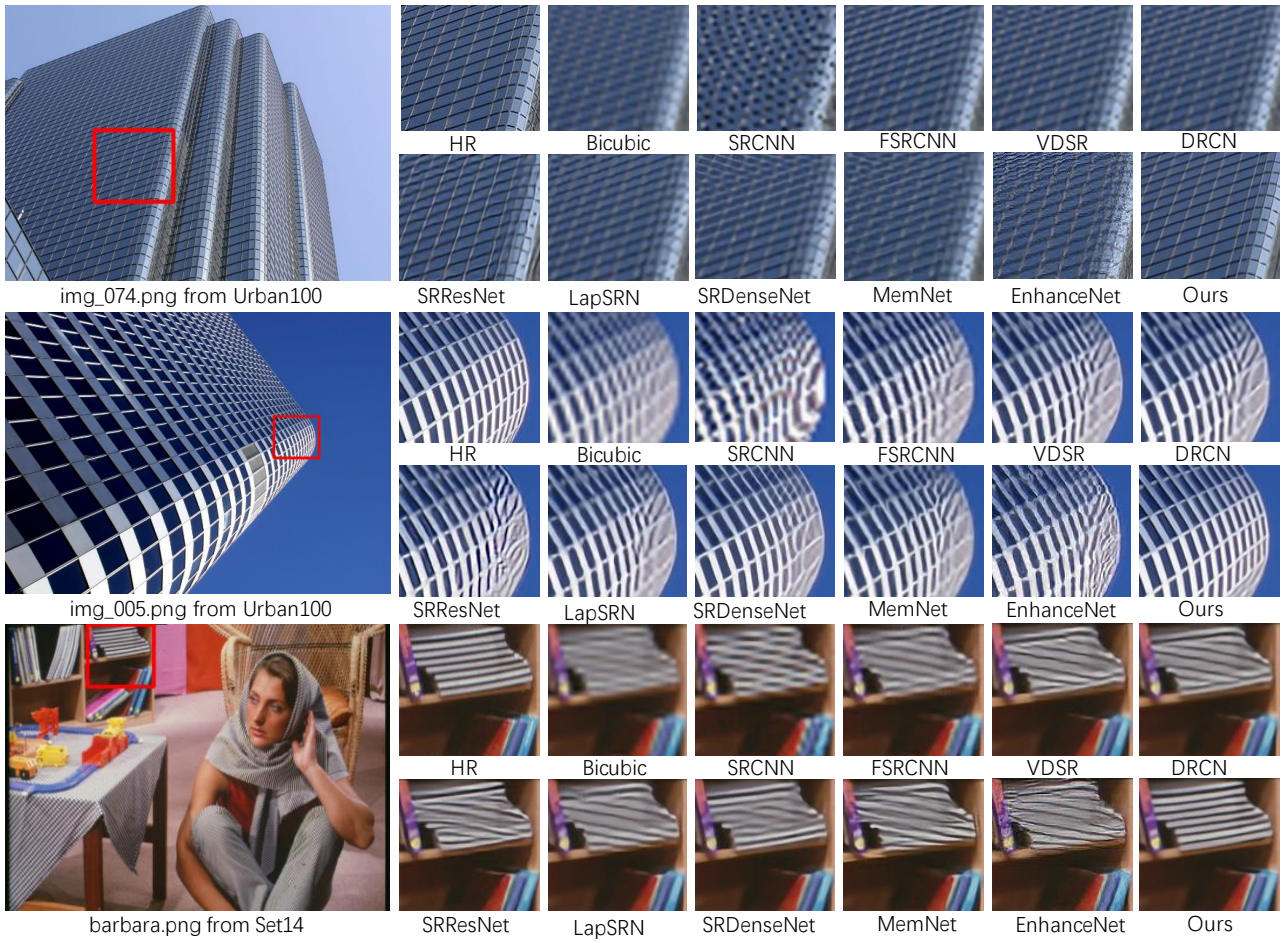


Fig. 4. The visual comparisons for a scale factor of $\times 4$ on the images from Urban100 and Set14. Only our HRFN can generate more accurate high-resolution (HR) images and recover more high-frequency details.

details in HR space. Therefore, we propose a novel method named hierarchical receptive field network (HRFN) to obtain features in various scales efficiently. To reduce computational cost, post-upsampling strategy is employed and upsampling module is placed at the tail of the model. As shown in Fig. 1, our HRFN contains four parts: embedding module for learning low-level features, deep feature extraction and fusion module for learning high-level features of various receptive fields, upsampling module and reconstruction module for generating the HR output.

First, a low-resolution image I^{LR} is taken as the input and fed into the embedding module to generate low-level features $F_L \in \mathbb{R}^{C\times H\times W}$. The operation can be defined as,

$$F_L = \mathcal{L}(I^{LR}) \tag{4}$$

where $\mathcal{L}(\cdot)$ represents the function of embedding module for learning low-level features.

Then these low-level features are used as the input of deep extraction and fusion module for obtaining more accurate high-level features. To efficiently extract high-level features, we adopt the block layout and stack HRFB modules. Meanwhile, local and global residual learning is used, which not only allows gradients to be directly back-propagated to bottom layers but also enables this module can make full use of intermediate features. This module can be formulated as,

$$F_H = \mathcal{H}(F_L) \tag{5}$$

$$= \mathcal{M}_N(\mathcal{M}_{N-1}(\cdots \mathcal{M}_1(F_L))) + F_L) \tag{6}$$

where $F_H \in \mathbb{R}^{C\times H\times W}$ is the high-level features. $\mathcal{H}(\cdot)$ and $\mathcal{M}_n(\cdot)$ denote the functions of this module and n-th HRFB module respectively.

Then these high-level features, which contain multi-scale information, are upsampled via the upsampling module. In our method, nearest-neighbor upsampling and convolutional layer are used to upsample the features to the target size.

$$F_{UP} = \mathcal{U}(F_H) = \mathcal{C}(\mathcal{N}(F_N)) \tag{7}$$

TABLE I
COMPARISONS OF PARAMETERS AND RECEPTIVE FIELD

Structures	Parameters	Receptive Field
k3d1_×4	36	9×9
k3d1_×6	54	13×13
k5d1_×3	75	13×13
k3d2_×4	36	17×17
k3d1-k3d2_×2*	36	13×13

TABLE II
PERFORMANCE OF DIFFERENT ARCHITECTURES WITH DIFFERENT RECEPTIVE FIELDS ON SET5

Architectures	w/o multi-scale	1-3	1-3-5	1-3-5-7*
PSNR(dB)	30.50	30.53	30.62	30.75

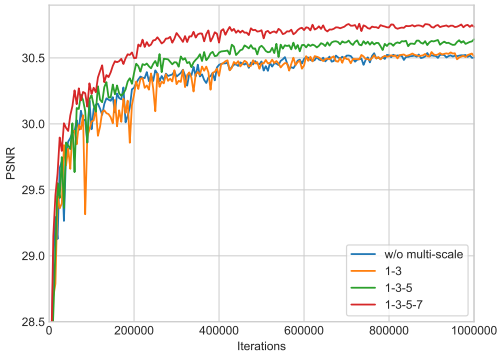


Fig. 5. The PSNR(dB) vs. Iterations curves on Set5 by four different structures as defined in Table II.

where $F_{UP} \in \mathbb{R}^{C \times sH \times sW}$ represents the upsampled features and s is the scale factor. $\mathcal{U}(\cdot)$ denotes the function of upsampling module. $\mathcal{C}(\cdot)$ and $\mathcal{N}(\cdot)$ are the convolutional operation and interpolation method respectively.

Finally, to avoid artificial priors induced by interpolation method and fine tune the final feature maps in HR space, two convolutional layers are applied at the end of the model and the last layer generates three feature maps, namely three channels in an image, to finish the reconstruction of the SR image.

$$I^{SR} = \mathcal{F}_{HRFN}(L^{LR}) \tag{8}$$

$$= \mathcal{R}(\mathcal{U}(\mathcal{H}(\mathcal{L}(I^{LR})))) \tag{9}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{R}(\cdot)$ are the functions of the whole HRFN model and the reconstruction module respectively.

C. Loss function

We have implemented an end-to-end model to learn the mapping function $\mathcal{F}_{HRFN}$ between I^{LR} and I^{HR}. So the optimization goal of HRFN is to prompt I^{SR} as close to I^{HR} as possible. Generally, L2 loss function is the default choice because it favors higher PSNR (peak signal-to-noise ratio). However, when adopting L2 loss function as the objective optimization function, the model often generates excessively smooth textures which does not accord with the human visual perception. In EDSR [18] and IDN [19], they found L1 loss function can lead to improved results. So in HRFN, we choose L1 loss as the loss function to train the model. There are multiple patch pairs $\{P^i_{LR}, P^i_{HR}\}^N$ in the training set, where N is the total number of patch pairs. P^i_{LR} and P^i_{HR} are the i-th LR and HR patches. Thus, the objective function can be formulated as,

$$\mathcal{L}^{SR}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|P^i_{HR} - \mathcal{F}_{HRFN}(P^i_{LR})\|_1 \tag{10}$$

where θ is the parameters of HRFN to be optimized.

IV. EXPERIMENTS AND ANALYSIS

In this section, we first introduce the implement details, including datasets, metrics for evaluation and training settings. Then we compare several structures which have different receptive fields and analyze the contribution of our HRFB. Finally, we evaluate the performance of our HRFN with some state-of-the-art methods on five extensively used benchmark datasets.

A. Implement Details

1) Datasets: Following [17], [18], we use DIV2k [21] as our training set which has 800 2k-resolution images. We first crop these 2k-resolution with 480×480 size as HR patches. Then these HR patches are downsampled with MATLAB bicubic kernel function to produce corresponding LR patches. For testing, five commonly used benchmark datasets are used to evaluate our HRFN, including Set5 [22], Set14 [23], BSD100 [24], Urban100 [25] and Manga109 [26]. Set5, Set14 and BSD100 are widely used in SISR. Urban100 contains 100 urban scene images with abundant buildings. Moreover, there are 109 Japanese comic images in Manga109.

2) Evaluation metrics: For SISR, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [27] are commonly adopted to evaluate the SR results. PSNR is a widely used objective function for image quality evaluation, which is determined by mean square error (MSE).

$$PNSR = 10 \times log_{10}(\frac{MAX_I^2}{MSE}) \tag{11}$$

where MAX_I denotes the maximum value of a pixel. SSIM aims to measure the structural similarity of two images and can be formulated as,

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2, +c_1)(\sigma_x^2\sigma_y^2 + c_2)} \tag{12}$$

where μ_x, μ_y and σ_x^2, σ_y^2 denote the mean and variance of image X and image Y respectively. σ_{xy} is the co-variance of image X and image Y. c_1 and c_2 are used to stabilize the division with weak denominator. As done in previous methods, our SR results are all evaluated on Y channel of YCbCr space with PSNR and SSIM.

TABLE III
QUANTITATIVE COMPARISONS WITH STATE-OF-THE-ART METHODS. THE BEST PERFORMANCE IS **HIGHLIGHTED** AND THE SECOND IS UNDERLINED.

Methods	Scale	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	33.68	0.9304	30.24	0.8691	29.56	0.8435	26.88	0.8405	30.80	0.9339
SRCNN [4]		36.66	0.9542	32.45	0.9067	31.36	0.8879	29.51	0.8946	35.72	0.9677
FRSRCNN [7]		36.98	0.9556	32.62	0.9087	31.50	0.8904	29.85	0.9009	36.56	0.9703
VDSR [5]		37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.16	0.9741
DRCN [8]		37.63	0.9588	33.04	0.9118	31.85	0.8942	30.75	0.9133	37.57	0.9731
LapSRN [29]		37.52	0.9591	33.08	0.9124	31.80	0.8949	30.41	0.9101	37.27	0.9740
MemNet [30]		37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
IDN [19]		37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196	38.02	0.9749
SRMDNF [32]		37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
D-DBPN [31]		38.09	0.9600	**33.85**	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
SRFBN [33]		38.11	0.9609	33.82	0.9196	32.29	0.9010	32.62	0.9328	39.08	0.9779
Ours		**38.17**	**0.9612**	**33.85**	**0.9206**	**32.31**	**0.9014**	**32.70**	**0.9336**	**39.21**	**0.9784**
Bicubic	×4	28.43	0.8109	26.00	0.7027	25.96	0.6678	23.14	0.6574	24.89	0.7866
SRCNN [4]		30.48	0.8628	27.50	0.7513	26.90	0.7103	24.52	0.7226	27.58	0.8555
FRSRCNN [7]		30.70	0.8657	27.59	0.7535	26.96	0.7128	24.60	0.7258	27.90	0.8610
VDSR [5]		31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8870
DRCN [8]		31.53	0.8854	28.02	0.7670	27.23	0.7233	25.14	0.7510	28.98	0.8860
LapSRN [29]		31.54	0.8866	28.19	0.7694	27.32	0.7270	25.21	0.7553	29.09	0.8900
MemNet [30]		31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
SRDenseNet [6]		32.02	0.8934	28.50	0.7782	27.53	0.7337	26.05	0.7819	-	-
IDN [19]		31.82	0.8903	28.25	0.7730	27.41	0.7297	25.41	0.7632	29.40	0.8936
SRMDNF [32]		31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
D-DBPN [31]		32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
SRFBN [33]		32.47	0.8983	28.81	0.7868	27.72	0.7409	26.60	0.8015	31.15	0.9160
Ours		**32.68**	**0.9010**	**28.91**	**0.7899**	**27.82**	**0.7446**	**27.01**	**0.8136**	**31.58**	**0.9202**

3) Training settings: When training, the HR patches are randomly cropped to 192×192 and the corresponding LR patches are also cropped to 96×96 and 48×48 under the down-sampling factors $\times 2$ and $\times 4$. Adam [28] optimizer is adopted to train our HRFN with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The initial learning is set as 2×10^{-4} and then halved after every 2×10^5 iterations.

B. Comparisons of Receptive Fields

To validate the effectiveness of our HRFB, we compare several different structures with ours in parameters and receptive field as shown in Fig. 2 and Table I. k and d mean the kernel size and dilation factor. When $d = 1$, the convolution is standard convolution. Otherwise, the convolution becomes dilated convolution with corresponding dilation factor d. As k3d1_$\times 4$ shown in Table I, four stacked standard 3×3 convolutional layers have 36 parameters to be optimized and can capture 9×9 receptive field. However, if we only increase the depth like k3d1_$\times 6$ or the kernel size like k5d1_$\times 3$ in Table I and Fig. 2, the parameters will be with huge growth. It suggests that the simply widening or deepening the structure is not efficient at all. When using dilated convolutions alone as k3d2_$\times 4$, blind spots will occur in receptive field as shown in Fig. 2(c) though dilated convolutions expand receptive field without additional parameters. These blind spots are unacceptable for SISR because it will cause the loss of information about adjacent pixels. But if we combine standard convolutions and dilated convolutions together as k3d1-k3d2_$\times 2$, this structure can remove the blind spots as shown in Fig. 2(e) and achieve a better trade-off between receptive field and parameters.

C. Contributions of Hierarchical Receptive Field Block

Furthermore, we explore the contribution of HRFB and conduct a series of ablation studies. There are four different architectures including the block without multi-scale branch (w/o multi-scale) which only has densely connected layers and the block with branches in different scales (1/3/5/7). As shown in Table II, the branch with different scales is added in turn and other settings remain the same for fair comparisons. We use the kernel sizes (1/3/5/7) of standard convolutions in each branch denote that branch, e.g. 1-3 represents 1×1 and 3×3 convolutional branches are adopted. It can be found in Table II that with the addition of large scale branches, the performance of SR results is improved correspondingly and PSNR finally gains 0.03dB, 0.12dB and 0.25dB. At the same time, we also illustrate the convergence curve against PNSR with these four architectures in Fig. 5. The block without multi-scale branches and the block 1-3 have similar convergence curves and final PSNR values. This is because 3×3 kernel is the main choice in our model and the block 1-3 does not provide any other scales for feature capture so there are no gains for PNSR. It is noted that the block with multiple branches (scales) has quicker convergence speed than those block with few branches (scales) and the block 1-3-5-7 which has four scales finally achieves best performance. The above results demonstrate our previous analysis and indicate it's necessary for SISR to extract multi-scale features for better reconstruction performance.

D. Comparisons with State-of-the-art Methods

To demonstrate the performance of our HRFN, we compare our model with several state-of-the-art methods including SR-

CNN [4], FSRCNN [7], VDSR [5], DRCN [8], LapSRN [29], MemNet [30], SRDenseNet [6], IDN [19], SRMDNF [32], D-DBPN [31], SRFBN [33]. Table III presents the summaries of the quantitative evaluations under $\times 2$ and $\times 4$ scale factors on five benchmark datasets: Set5, Set14, BSD100, Urban100 and Manga109. The best performance is **highlighted** and the second is underlined. Obviously our model outperforms other state-of-the-art methods in terms of PSNR and SSIM under $\times 2$ and $\times 4$ scale factors. Especially for Urban100 which contains a lot of modern architectures and is hard to be recovered, our HRFN achieves a large margin 0.41dB under $\times 4$ scale factor than others. These comparisons indicate the superiority of our HRFN.

At the same time, we also illustrate some visual comparisons with several state-of-the-art methods as shown in Fig. 4 and our HRFN could accurately recover more accurate textures and more faithful results. In img_074.png and img_005.png, the compared methods obviously suffer from blurring artifacts but our HRFN can predict more accurate details and alleviate the artifacts. Specially in barbara.png from Set14, those methods all generate the wrong high-frequency details of the books. The edges of books are mixed and fuzzy even there are severe distortions. By contrast, our HRFN generates the more faithful result which contains more accurate edges and details. The visual comparisons further demonstrate the necessity of multi-scale features and large receptive field.

V. CONCLUSION

In this paper, a hierarchical receptive field network is proposed to generate high-resolution images by capturing multi-scale features. Specially, we combine the standard convolutions and the dilated convolutions to expand receptive fields without additional parameters and construct a multi-branch block named hierarchical receptive field block to extract and fuse multi-scale features for better reconstruction performance. Meanwhile, residual learning is adopted locally and globally, which can ease the training process and alleviate gradients vanishing/exploding. The extensive experimental results on benchmark datasets demonstrate our HRFN achieves better performance than those state-of-the-art methods on both quantitative and visual comparisons.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[2] D. Kouame and M. Ploquin, "Super-resolution in medical imaging : An illustrative approach through ultrasound," in 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Boston, MA, USA, 2009, pp. 249–252, doi: 10.1109/ISBI.2009.5193030.

[3] W. Ma, Z. Pan, J. Guo, and B. Lei, "Achieving Super-Resolution Remote Sensing Images via the Wavelet Transform Combined With the Recursive Res-Net," IEEE Trans. Geosci. Remote Sensing, vol. 57, no. 6, pp. 3512–3527, Jun. 2019, doi: 10.1109/TGRS.2018.2885506.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2016, doi:10.1109/TPAMI.2015.2439281.

[5] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1646–1654, doi: 10.1109/CVPR.2016.182.

[6] T. Tong, G. Li, X. Liu, and Q. Gao, "Image Super-Resolution Using Dense Skip Connections," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 4809–4817, doi: 10.1109/ICCV.2017.514.

[7] C. Dong, C. C. Loy, and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network," in Computer Vision – ECCV 2016, vol. 9906, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 391–407.

[8] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1637–1645, doi: 10.1109/CVPR.2016.181.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[11] C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 105–114, doi: 10.1109/CVPR.2017.19.

[12] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," arXiv:1511.07122 [cs], Apr. 2016.

[13] Lei Zhang and Xiaolin Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," IEEE Trans. on Image Process., vol. 15, no. 8, pp. 2226–2238, Aug. 2006, doi: 10.1109/TIP.2006.877407.

[14] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li, "Single Image Super-Resolution With Non-Local Means and Steering Kernel Regression," IEEE Trans. on Image Process., vol. 21, no. 11, pp. 4544–4556, Nov. 2012, doi: 10.1109/TIP.2012.2208977.

[15] Y. Tai, J. Yang, and X. Liu, "Image Super-Resolution via Deep Recursive Residual Network," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2790–2798, doi: 10.1109/CVPR.2017.298.

[16] W. Shi et al., "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1874–1883, doi: 10.1109/CVPR.2016.207.

[17] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Super-Resolution," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2472–2481, doi: 10.1109/CVPR.2018.00262.

[18] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," arXiv:1707.02921 [cs], Jul. 2017.

[19] Z. Hui, X. Wang, and X. Gao, "Fast and Accurate Single Image Super-Resolution via Information Distillation Network," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 723–731, doi: 10.1109/CVPR.2018.00082.

[20] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 4501–4510, doi: 10.1109/ICCV.2017.481.

[21] R. Timofte et al., "NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results," in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, pp. 1110–1121, doi: 10.1109/CVPRW.2017.149.

[22] M. Bevilacqua, A. Roumy, C. Guillemot, and M. A. Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in Procedings of the British Machine Vision Conference 2012, Surrey, 2012, pp. 135.1-135.10, doi: 10.5244/C.26.135.

[23] R. Zeyde, M. Elad, and M. Protter, "On Single Image Scale-Up Using Sparse-Representations," in Curves and Surfaces, vol. 6920, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schumaker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730.

[24] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

[25] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 5197–5206, doi: 10.1109/CVPR.2015.7299156.

[26] Y. Matsui et al., "Sketch-based manga retrieval using manga109 dataset," Multimed Tools Appl, vol. 76, no. 20, pp. 21811–21838, Oct. 2017, doi: 10.1007/s11042-016-4020-z.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. on Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs], Jan. 2017.

[29] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5835–5843, doi: 10.1109/CVPR.2017.618.

[30] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A Persistent Memory Network for Image Restoration," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 4549–4557, doi: 10.1109/ICCV.2017.486.

[31] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep Back-Projection Networks for Super-Resolution," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 1664–1673, doi: 10.1109/CVPR.2018.00179.

[32] K. Zhang, W. Zuo, and L. Zhang, "Learning a Single Convolutional Super-Resolution Network for Multiple Degradations," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 3262–3271, doi: 10.1109/CVPR.2018.00344.

[33] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback Network for Image Super-Resolution," in 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 3867-3876.

[34] C. Szegedy et al., "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.