Receptive Field Size Versus Model Depth for Single Image Super-Resolution

Ruxin Wang, Mingming Gong, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— The performance of single image super-resolution (SISR) has been largely improved by innovative designs of deep architectures. An important claim raised by these designs is that the deep models have large receptive field size and strong nonlinearity. However, we are concerned about the question that which factor, receptive field size or model depth, is more critical for SISR. Towards revealing the answers, in this paper, we propose a strategy based on dilated convolution to investigate how the two factors affect the performance of SISR. Our findings from exhaustive investigations suggest that SISR is more sensitive to the changes of receptive field size than to the model depth variations, and that the model depth must be congruent with the receptive field size to produce improved performance. These findings inspire us to design a shallower architecture which can save computational and memory cost while preserving comparable effectiveness with respect to a much deeper architecture.**

***Index Terms*— Receptive field size, model depth, dilated convolution, single image super-resolution.**

I. Introduction

A HIGH-RESOLUTION (HR) image with delicate details is highly expected for either aesthetics or applications such as computer vision, medical imaging, video surveillance, and entertainment. Unfortunately, many undesirable imaging conditions hinder the acquisition of such images, but result in low-resolution (LR) substitutes. This poses the necessity of single image super-resolution (SISR), which is a classical image processing problem that aims at recovering a high-resolution image from its corresponding single low-resolution image. As any other inverse problems, SISR is intrinsically ill-posed due to the information loss caused by image degradation. The solution of SISR is not unique since a LR image can be generated by different HR images, thus making the determination of the correct solution non-trivial.

Manuscript received August 3, 2017; revised March 5, 2018 and September 27, 2018; accepted August 30, 2019. Date of publication September 25, 2019; date of current version November 27, 2019. This work was supported by the Australian Research Council Project FL-170100117. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Denis Kouame. *(Corresponding author: Mingming Gong.)*

R. Wang is with Union Visual Innovation Technology Co., Ltd., Shenzhen 518000, China (e-mail: rosinwang@gmail.com).

M. Gong is with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: mingming.gong@unimelb.edu.au).

D. Tao is with the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Sydney, NSW 2008, Australia, and also with the School of Computer Science, Faculty of Engineering, The University of Sydney, Sydney, NSW 2008, Australia (e-mail: dacheng.tao@sydney.edu.au).

Digital Object Identifier 10.1109/TIP.2019.2941327

Targeting on this task, the literature typically constrains the solution space based on a series of assumptions, and can be grouped into three categories: interpolation-based, reconstruction-based, and example-based. A recent and very active trend of SISR is the deep convolutional neural network (CNN)-based methods, which have shown excellent performances. Dong *et al.* [1], [2] pioneered the utilisation of CNN for SISR which achieved promising results by building a three to four-layer architecture. Subsequently, the state-of-the-art SISR performance was being refreshed by a series of improvements on the CNN architecture [3]–[7]. Most of these works were motivated by the successful application of deep CNNs on image classification or recognition [8]–[10], which tells us that deepening the architecture can significantly boost the performance of various vision tasks. The reason is that a deeper model has higher nonlinearity and larger receptive field, both of which are important for the semantic understanding of the input image. Likewise, the SISR community began to investigate how to design and train a very deep CNN model [3], [4], and how to exploit the statistics in the semantic feature space characterised by CNN feature maps [7], [11].

While the current SISR achievements are fascinating, we still have questions about the CNN architecture, which is extensively studied in the deep learning community [12]. Does the CNN architecture for SISR really need to be deep (or need so high nonlinearity)? How large receptive field does SISR require? Which is more dominant among model depth (or model nonlinearity) and receptive field size? We interpret these questions from the perspective of SISR as: Is it beneficial to apply a complex function on a very local region to estimate the HR pixel? Or is it good to use a relatively simpler function on a wide region? Or are both a complex function and a wide region needed? These questions inspire us to deeply research and understand which aspects are critical for SISR.

To pursue the answers, we are firstly aware that a larger receptive field of a CNN model captures more contextual information. We use the term "the receptive field of a CNN model" to denote the receptive field of the output layer of the model. Most image classification networks [8]–[10], [13] employ successive pooling and subsampling operations to gradually reduce the resolution of feature maps and extend the size of the receptive field, such that the output layer can respond to the whole input image while maintaining a manageable memory cost. But for the SISR task, a dense prediction of all pixels in the HR image is needed, which prevents the reduction of the feature map resolution, because

the pooling and subsampling operations will cause information lost. To mimic a similar behaviour in SISR, we propose to expand the size of the receptive field by using dilated convolution, without sacrificing the resolution and staking the CNN layers to an unnecessary depth.

Dilated filters have been widely used in the wavelet theory [14]–[16]. They are the key concept for realising the multi-resolution analysis, where the mother wavelets are scaled (or dilated) and translated according to a dilation parameter and a translation parameter. According to the lemma proposed by Holschneider *et al.* [15], the convolution with a dilated filer can be factorised as simpler convolutions. The proposed *algorithme à trous* is similar to the convolution and subsampling operations widely used in CNN. In our work, to prevent resolution reduction, we can perform an inverse factorisation, which means that the subsampling and convolution operations are replaced by a dilated convolution. Instead of explicitly representing the dilated filter based on the learned convolutional filters, the filter parameters of the dilated convolutional layer are trained in conjunction with other layers in the CNN model.

In this paper, we propose to integrate the dilated convolutional layer into CNN models to investigate the questions raised previously. The usage of dilated convolution layers can avoid a large variation in the number of model parameters. To the best of our knowledge, this is the first attempt to apply the dilated convolution in the SISR task. In specific, we investigate the effects of different receptive field sizes on SISR by keeping constant model nonlinearity. Dilated convolutional layers are optionally used to enlarge the receptive field of the CNN models. It is also studied that how an increased receptive field size affects the SISR performance of different kinds of images, given a fixed model depth. In this task, we are towards finding the effective contextual region size required by SISR. For another task, we investigate the effects of the model depth on SISR by fixing the receptive field size. The convolutional layers with 1×1 filters can be stacked to increase the nonlinearity while keeping receptive field size unchanged. We try to analyse from the results whether the nonlinearity is an important aspect for SISR. The findings reveal the dominant relationship and the interdependent relationship between the two factors, which can guide the designs of effective and efficient CNN models for SISR.

The rest of the paper is organised as follows. Section II provides a review on the recent methods of SISR and the detailed techniques of dilated convolution. The basic settings used for experimental comparisons are introduced in Section III. In Section IV, we conduct comprehensive analysis on the effects of different receptive field sizes on the SISR performance. Section V presents the results and remarks on how model depth influences the SISR performance. A further discussion on our empirical findings is provided in Section VI. Finally, Section VII draws the conclusions.

II. Related Work

A. Single Image Super-Resolution

The recent decades have witnessed the rapid development of SISR from conventional interpolation-based methods to reconstruction-based methods, and further to modern example-based methods. Comprehensive review articles include [17] and [18].

The interpolation-based techniques, which employ a parametric interpolation function to estimate the pixels in the HR grids, include bilinear interpolation, bicubic interpolation, nearest neighbour interpolation, and structure-adaptive interpolation [19], [20]. While being very fast, these methods generally result in blurry effects and zigzag artifacts in the HR estimations, due to limited receptive field size and low complexity of the functions.

The reconstruction-based methods typically rely on regularisation techniques or priors which reveal the expected properties of HR images, such as sparse gradients [21], correlated gradient profile between HR and LR images [22], [23], distinctive edge dependency [24], and redundant self-similarity in nature images [25]–[27]. By integrating into the maximum a posterior (MAP) framework [28] or the variational Bayes framework [29], these priors can promote sharp edges and suppressed aliasing artifacts in the results. However, the complex inference processes induce high computational cost, limiting the feasibility in real-time applications.

The example-based methods benefit a lot from modern supervised learning techniques that rely on external training data. These studies typically concern two problems: representation learning and regression-based mapping learning. To solve the first problem, a key assumption is that the LR/HR image patches have a shared representation on the corresponding LR/HR dictionaries [30]–[32] or manifolds [33]. Regarding the second one, a mapping function from the LR to the HR patch spaces is learned through designing regression formulations, such as simple functions [34], kernel regression [35], support vector regression [36], and anchored neighbourhood regression [37], [38]. While all these methods have shown pleasing performances, we should note that they are intrinsically shallow models that have limited representation capacity, and thus it is difficult to further improve the performance.

One way to substantially increase the model complexity is to stack shallow models into a deep one, as in CNN. The very recent works on CNN produced surprising improvements on SISR performance, particularly in terms of the peak signal-to-noise ratio (PSNR). Following the initial CNN work of Dong *et al.* [1], [2], an investigation on using very deep convolutional networks for SISR was conducted by Kim *et al.* [3], who proposed to train a 20-layer CNN model via residual learning and showed the state-of-the-art PSNR values. The same group also presented a deeply-recursive convolutional network [4] that allows for long-range pixel dependencies by using small number of model parameters. Wang *et al.* [39] proposed to employ self examples of multiple scales to fine-tune a pre-trained convolutional auto-encoder in the SR task. Similar to the architecture of [3], Wang *et al.* [5] developed a combined deep and shallow network, where the shallow network can be regarded as learning a pre-super-resolved image from the LR input, while the deep network learns the residuals as in [3]. To accelerate the SR computation, Shi *et al.* [6] and Dong *et al.* [40] designed networks in which

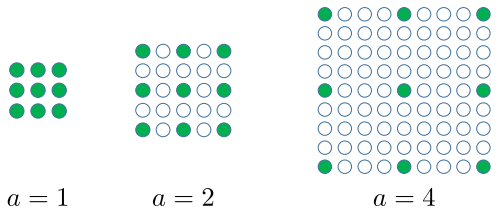


Fig. 1. 2D dilated filters. The solid circles indicate the filter parameters, while the hollow circles indicate zeros that are inserted during dilation.

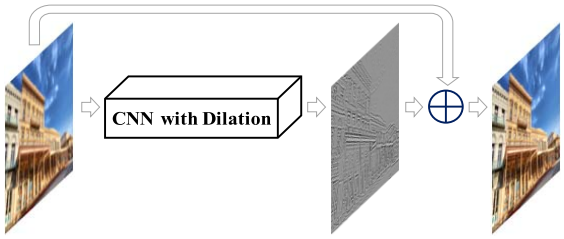


Fig. 2. Basic CNN architecture. The CNN model with dilation will be specified in respective tasks.

most of the computation occurs in the LR space and the up-scaling operation is only performed in the last layer of the networks. Besides the above CNN-related works, the studies on other feed-forward neural networks were conducted, such auto-encoders [41]–[43] and sparse coding-based networks [44], [45]. More recent studies focus on how to generate pleasing textures in SR when the scale factor gets large [7], [11], [46], [47].

B. Dilated Convolution

In the field of wavelet decomposition, a signal is decomposed through a series of convolutions with filters that have different scales. These filters form a filter bank, which can be generated by introducing a dilation factor into the so-called mother wavelet [14], [15], [48]. In mathematics, let $f(t)$ be a function in the space $L_2(\mathbb{R})$, and $\psi(t) \in L_2(\mathbb{R})$ be a filter which is dilated via $\psi_a(t) = \psi(\frac{t}{a})$. The convolutions with a non-dilated filter and with a dilated filter are, respectively, expressed as,

$$(f * \psi)(t) = \int_\tau f(\tau)\psi(t-\tau)d\tau, \tag{1}$$

$$(f * \psi_a)(t) = \int_\tau f(\tau)\psi(\frac{t-\tau}{a})d\tau, \tag{2}$$

where $*$ is the convolution operation. In discrete case, Eq. (2) is rewritten as

$$(f * \psi_a)[k] = \sum_\upsilon f[\upsilon]\psi[\frac{k-\upsilon}{a}]. \tag{3}$$

The *algorithme à trous* implements the convolution with a dilated filter through the convolution with a small filter followed by subsampling. Such an implementation indicates that, in discrete case, ψ is dilated by zeros to generate ψ_a, which means $\psi_a[k] = 0$ when k is divisible by a. Then, Eq. 3 is equal to

$$(f * \psi_a)[k] = \sum_\upsilon f[k - a\upsilon]\psi[\upsilon], \tag{4}$$

which is referred as the dilated convolution operation by Yu and Koltun [49]. A dilated convolutional layer can be integrated into a deep CNN model, where the filter ψ is learned via back-propagation. This is different from the dilated filters in wavelet decomposition, which must be constructed under rigorous conditions, such as orthonormality.

Suppose that ψ has a finite support of r in discrete case, then the effective size of ψ_a is calculated as $a(r-1)+1$. The value of the dilation factor a is chosen to be exponentially increased, such that the subsampling operation either in *algorithme à trous* or in the 2-stride pooling layer of deep models can be compensated, *i.e.* $a = 2^i$ for $i = 0, 1, ..., n$. The generalisation of the above formulations to 2D case is straightforward and omitted here. Fig. 1 gives an illustration of the 2D dilated filters.

Dilated convolutional layers have recently been employed in the image segmentation task [49], [50]. The motivations for using dilation are, on one hand, to avoid reducing the resolution of CNN feature maps and increasing the number of model parameters, and on the other hand, to maximally expand the receptive field to grasp global information of the input image.

III. Basic Settings for Comparison

We consider the up-scaling factors commonly used in most SISR methods, including $\times 2$, $\times 3$, and $\times 4$. The questions presented in Section I will be investigated in details for each of these factors. Even though it has been proven that a single model is competent for multiple scales [3], we empirically found that such a model requires high complexity and its training takes long time to converge. In our studies, we train the CNN models for these factors separately.

The training data for all studies is the 291 images as in [3], [51]. Data augmentation, such as flip and rotation, is used. During training, the samples of size 80×80 are randomly cropped from the images on-the-fly. The test data includes Set5 [52], Set14 [53], the Berkeley Segmentation 100 test images (BSD100) [54], and Urban100 [27]. We use bicubic interpolation to generate the LR images.

Only the luminance channel of images are used to train and test the models, while the chromatic channels are up-scaled via bicubic interpolation. As pointed out by Kim *et al.* [3], training becomes easier when the CNN model predicts the residual between the ground truth and the bicubic interpolated images, rather than directly predict the HR image. We follow such a residual learning strategy to save training time. Thus, our CNN models take as input an 80×80-sized bicubic interpolated image, and output a residual image with the same size. The HR result is then estimated by a summation operation. The general architecture is depicted in Fig. 2, where the CNN model is constructed according to the specific settings in Sections IV and V.

Following most SISR methods [2], [3], we employ the Euclidean distance (equivalent to MSE) as the training objective:

$$\ell(x_{re}, \hat{x}) = \|x_{re} - \hat{x}\|^2,$$
$$x_{re} = x - x_{bi}, \tag{5}$$

TABLE I
TRAINING PARAMETERS

Parameter	Value
batch size	64
optimization algorithm	SGD
initial learning rate	0.1
learning rate update policy	step
decreasing factor of learning rate	0.1
step size for decreasing learning rate	70000
momentum	0.9
weight decay	0.0001
clip gradients	0.1
max number of iterations	100000
learning rate multiplier for filter parameters	1
learning rate multiplier for bias terms	0.1

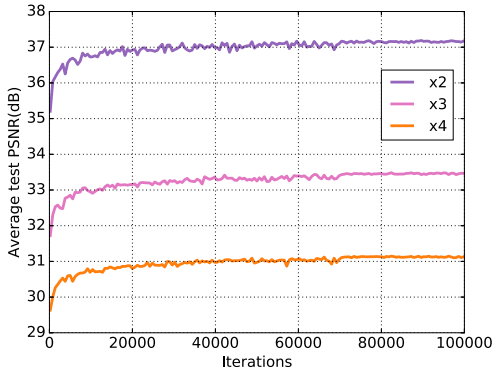


Fig. 3. The curves of test performances on Set5 for $\times 2$, $\times 3$, and $\times 4$ tasks.

where x_{re}, $\hat{x}$, x, and x_{bi} are the ground truth residual image, the estimated residual image, the ground truth HR image, and the bicubic interpolated HR image, respectively. We use the Caffe package [55] to implement the training and test phases. The necessary training parameters are summarised in Table I. The gradient clipping technique is employed, which was shown to be beneficial for training CNN networks in SISR task [3]. To avoid numerical instability in float-type computation, we normalise the pixel range of the images from $[0, 255]$ to $[0, 1]$. The maximal iterations for training and the step size for decreasing learning rate are set according to experience. The curves of test performances in Fig. 3 verify that training can converge under such settings and no overfitting occurs.

We note that in SISR, the HR image pixels have close values to the LR image pixels, which means that the output values of the network are close to the input values. Thus, the bias terms in convolutional layers are not expected to have large values. Regarding this, we set the learning rate of all bias terms as 0.1 of the learning rate of filter parameters, as indicated by the last row of Table I.

IV. EFFECTS OF RECEPTIVE FIELD SIZE ON SISR

In this section, we investigate the effects of different receptive field sizes on SISR. For comparison, the model nonlinearity is fixed, which is achieved by keeping the model depth unchanged. In this condition, we employ the dilated convolutional layers which can exponentially increase the receptive field of a CNN model, as introduced in Section II-B.

TABLE II
RECEPTIVE FIELD SIZES OF DIFFERENT DILATED FILTERS (3×3-BASIC SIZE)

a	1	2	4	8	16
Receptive field size	3	5	9	17	33

In the following, we will use DC to denote the dilated convolution for concise presentation.

We begin by describing the notations that facilitate us to specify the CNN architectures designed in the following. We use LmDn to denote that a CNN model contains $m + 1$ convolutional layers among which n DC layers are inserted. The one additional convolutional layer (*i.e.* "+1") is used for reconstructing the single-channel output, which is topped on all CNN models.

In all convolutional layers including the dilated ones but except the last layer, we use 64 filters with the basic size of 3×3 (dilation enlarges the filter size according to the dilation factor). While a larger filter can increase the receptive field, high computational cost would be induced. 3×3 has been shown in different CNN-based applications [3], [8], [10] to be effective to encode local structures of the input while preserving small number of parameters.

To insert DC layers, we have two concerns: how to change the dilation factor and where to insert. Regarding the first one, we follow the dilation strategy which is consistent with the multi-scale wavelet decomposition, *i.e.* $a = 2^i$ for the i-th DC layer where $i = 1, ..., n$. For example, if $n = 3$, a will be set to 2, 4, and 8 for the first, second, and third DC layers, respectively. Even though a can be set arbitrarily to increase the receptive field, such as in image segmentation [50], their motivation is to exploit global contexts as much as possible for semantically understanding the category of each pixel. However, we intuitively realise that it is more important for SISR to analyse local structures rather than global contexts. Too large receptive field is not helpful for improving the SISR performance, as demonstrated in the following experiments. In our work, the maximal number of DC layers that we will insert is $n = 4$. The receptive field sizes of different dilated filters are listed in Table II.

For the second concern, our experiences indicate that when the DC layers are inserted into the very beginning position (close to the input) or the very end position (close to the output) of the network, the performance is limited. It is possibly because 1) the nonlinear function expressed by the first few layers is not effective for analysing the local structures in a large region of the LR image, and 2) the reconstruction of a HR pixel should be performed on local structures rather than distant points on feature maps. Considering this, here, we insert the DC layers around the middle position of the CNN model. In specific, when LmDn is used, the DC layers are successively placed after the $\lfloor \frac{m+1-n}{2} \rfloor$-th layer.

Given the above strategy of designing architectures, one may argue that the number of model parameters varies which may lead to an unfair comparison between different models. However, we insist that the performance of a vision task (including SISR) is highly dependent on architecture settings

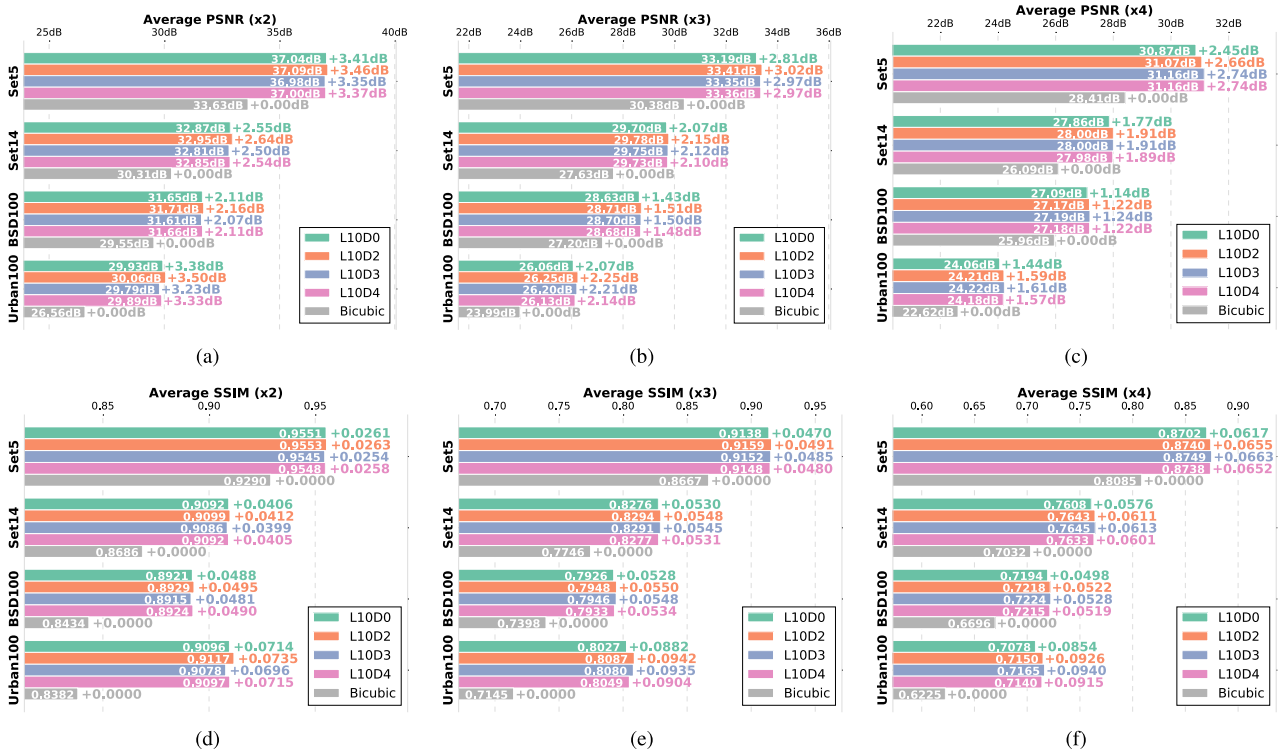


Fig. 4. The SISR performances of the L10 models with different number of dilated convolutional layers. The up-scaling factor used in each comparison is marked in the top of the subfigures. In each dataset, the positive value on the right of each bar indicates the improved quality by the corresponding model *w.r.t.* bicubic interpolation. The minimum at x-axis is adjusted for better visualisation of the differences between the performances.

and training strategies. While a large variation in the number of model parameters matters, a small variation is negligible and is less important than a reasonable architecture setting. This has been indicated by the similar classification performances of ResNet-101 [8] and Inception-V3 [9], which, however, have different architectures and different numbers of model parameters. In our work, to minimise the influence cased by a large variation of the number of parameters, we thus use the dilated convolution instead of the regular convolution.

Regarding the image quality assessment metric, we denote the assessment metric by $\mathcal{P}^s_{\text{model}}$, where the subscript indicates the used model and the superscript denotes the up-scaling factor. For example, $\mathcal{P}^2_{\text{bi}}$ means the quality produced by bicubic interpolation in the $\times 2$ SISR task. The *improved quality* by method A with respect to (*w.r.t.*) B is defined as

$$\Delta_B \mathcal{P}^s_A = \mathcal{P}^s_A - \mathcal{P}^s_B. \tag{6}$$

In our work, we consider two metrics including peak-to-signal noise ratio (PSNR) and structural similarity (SSIM) [56]. In the following analyses, we use $\mathcal{P}^s_{\text{model}}$ to denote both cases of the metrics. A detailed explanation will be provided to avoid ambiguous understandings when necessary. We first evaluate the performances of the L10 models with different receptive field sizes.

A. L10 CNN Models

For the L10 models, we choose four schemes to insert the DC layers, including L10D0, L10D2, L10D3, and L10D4. The inserted positions and the respective field sizes are summarised in Table III, which indicates that diverse sizes of receptive field are considered.

TABLE III
ARCHITECTURE SETTINGS OF THE L10 CNN MODELS

Model	DC positions	Receptive Field Size
L10D0	-	23
L10D2	6, 7	31
L10D3	5, 6, 7	45
L10D4	4, 5, 6, 7	75

We first illustrate the average qualities on each of the test sets, as shown in Fig. 4. Since the models take as input the bicubic interpolated images, we calculate the improved quality of each model *w.r.t.* bicubic interpolation, *i.e.* $\Delta_{bi}\mathcal{P}^s_{\text{L*D*}}$, to show how much the models can boost the SISR performance, as marked in Fig. 4.

Fig. 4a shows that the model L10D2 achieves the best PSNR value. But only a marginal improvement can be observed compared with the others (in most cases less than 0.1dB). The second best model is L10D0 which performs almost the same as L10D2. An interesting case is that L10D3 is beat by both L10D2 and L10D4, which empirically indicates that the receptive field size of 45 may be a bad choice under the setting of L10. Similar phenomenon can also be observed from the SSIM metric, as shown in Fig. 4d. These results suggest that when using a L10 model, 1) increasing the receptive field size does not necessarily benefit for the $\times 2$ SISR task, and

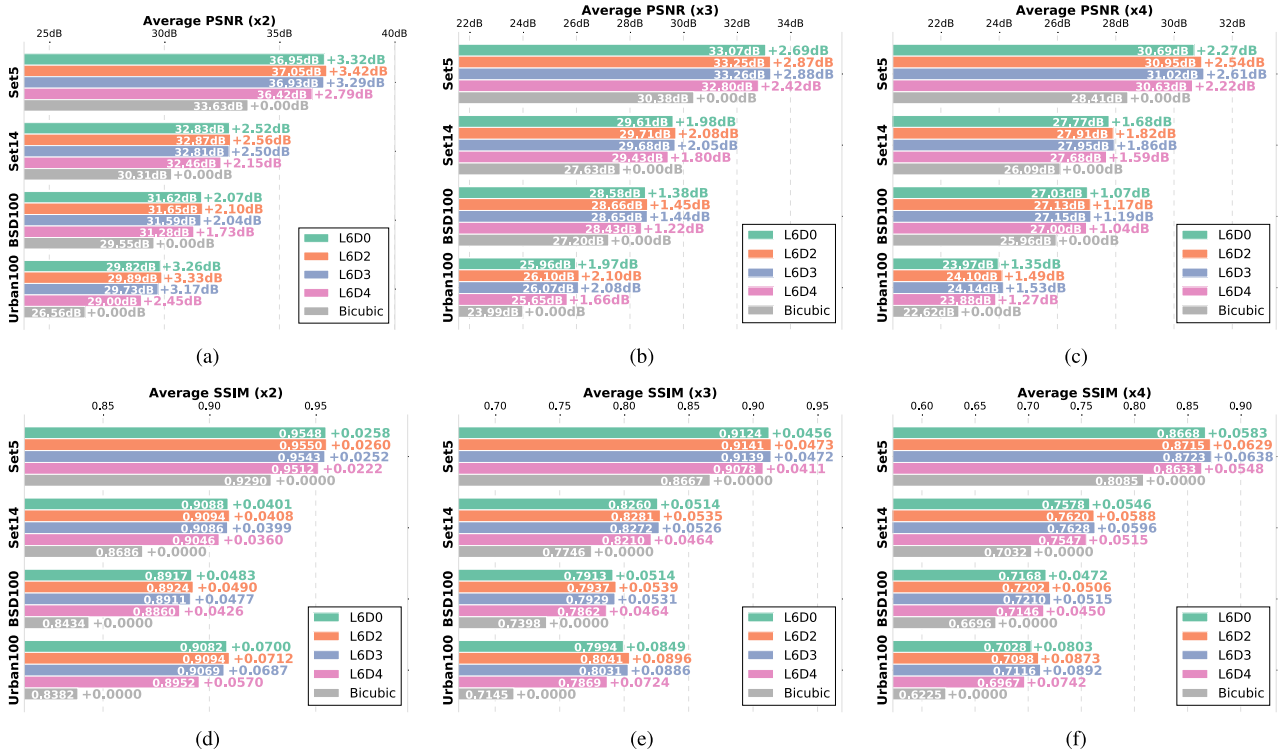


Fig. 5. The SISR performances of the L6 models with different number of dilated convolutional layers.

2) a receptive field size between 23 to 31 is large enough to produce a pleasing performance here. This tells us that for the up-scaling factor of 2, a HR pixel is highly correlated with its surrounding pixels, and it is better to perform the HR estimation locally. Getting more pixels involved in estimation may degrade the final performance, given that the model nonlinearity is specified by L10.

In Fig. 4b, it is observed that L10D2 surpasses L10D0 by 0.1∼0.2dB in all datasets, and is the best one in setting of $\times 3$. Focusing on the dilated models, we see that the performance monotonously decreases when the receptive field size increases. In spite of this, all the dilated models (*i.e.* L10D2, L10D3, L10D4) perform better than L10D0. The results on SSIM metric in Fig. 4e depict similar phenomenons. We analyse from those figures that in the $\times 3$ SISR task, a large receptive field size is preferred, *e.g.* 31 (although it may be not the optimum); and in addition, a larger receptive field size may bring deficiencies in performance, which can be compensated by increasing the model depth.

In the SISR task of $\times 4$, Figs. 4c and 4f illustrate that the receptive field size of 45 (specified by L10D3) is the best choice among the others. When smaller than 45, the size is positively correlated with the performance; see L10D0, L10D2 and L10D3. But the fourth DC layer does not help to improve the performance further, as in the case of L10D4. These results advise that the $\times 4$ SISR task requires a large receptive field size to cover more image contexts for the reconstruction of a single HR pixel. This is reasonable because much valuable information is lost in the down-sampled images for $\times 4$, especially in the texture regions. The restoration thus needs more detailed information in distant regions, not only in a local region surrounding the HR pixel.

TABLE IV

COMPARISON OF IMPROVED QUALITY $\Delta_{\text{L10D0}}\mathcal{P}^s_{\text{BEST}}$ FOR THE L10 MODELS

	$\Delta_{\text{L10D0}}\mathcal{P}^2_{\text{L10D2}}$	$\Delta_{\text{L10D0}}\mathcal{P}^3_{\text{L10D2}}$	$\Delta_{\text{L10D0}}\mathcal{P}^4_{\text{L10D4}}$
Set5	0.05/0.0002	0.21/0.0021	**0.29/0.0046**
Set14	0.09/0.0006	0.08/0.0018	**0.14/0.0037**
BSD100	0.05/0.0007	0.08/0.0022	**0.10/0.0030**
Urban100	0.12/0.0021	**0.18**/0.0060	0.17/**0.0086**

By comparing the results of $\times 4$ and $\times 3$, we see that $\Delta_{\text{L10D0}}\mathcal{P}^4_{\text{L10D3}} > \Delta_{\text{L10D0}}\mathcal{P}^3_{\text{L10D2}}$ for both PSNR and SSIM in each dataset, where L10D3 and L10D2 are the best models in the respective tasks. Similar effects can also be observed between $\times 3$ and $\times 2$ according to $\Delta_{\text{L10D0}}\mathcal{P}^3_{\text{L10D2}} > \Delta_{\text{L10D0}}\mathcal{P}^2_{\text{L10D2}}$. The quantitative comparison can be found in Table IV, from which we know that enlarging the receptive field size can help more for the $\times 4$ SISR task than for both the $\times 2$ and $\times 3$ SISR tasks.

After analysing Fig. 4 and Table IV, we obtain the following remarks. On one hand, given a L10 model, a HR pixel can be effectively estimated based on a region of size 23∼31 for $\times 2$ task, of size 31 for $\times 3$ task, and of size 45 for $\times 4$ task. Thus, it is beneficial to increase the receptive field size on the basis of a L10D0 model. On the other hand, when the degradation of an image gets severer, a large receptive field (within a certain range) is more helpful than a small one for producing the top SISR performance.

B. L6 and L20 CNN Models

In this subsection, we conduct more experiments to verify the remarks obtained by using L10 models in last subsection. In specific, shallower models L6D*n* and deeper models L20D*n*

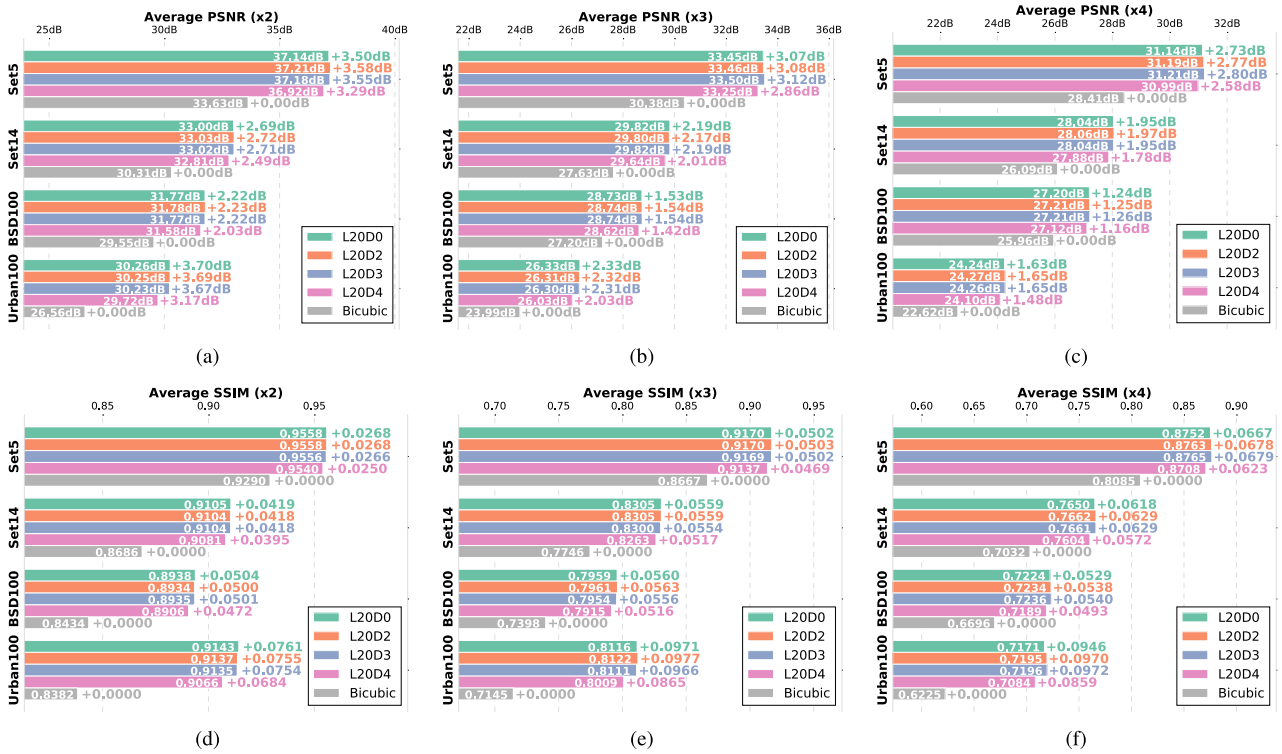


Fig. 6. The SISR performances of the L20 models with different number of dilated convolutional layers.

TABLE V
ARCHITECTURE SETTINGS OF THE L6 AND L20 CNN MODELS

Model	DC positions	Receptive Field Size
L6D0	-	15
L6D2	3, 4	23
L6D3	3, 4, 5	37
L6D4	2, 3, 4, 5	67
L20D0	-	43
L20D2	10, 11	51
L20D3	10, 11, 12	65
L20D4	9, 10, 11, 12	95

TABLE VI
COMPARISON OF IMPROVED QUALITY $\Delta_{\text{L6D0}}\mathcal{P}^s_{\text{BEST}}$ FOR THE L6 MODELS

	$\Delta_{\text{L6D0}}\mathcal{P}^2_{\text{L6D2}}$	$\Delta_{\text{L6D0}}\mathcal{P}^3_{\text{L6D2}}$	$\Delta_{\text{L6D0}}\mathcal{P}^4_{\text{L6D4}}$
Set5	0.10/0.0002	0.18/0.0017	**0.34/0.0055**
Set14	0.04/0.0007	0.10/0.0021	**0.18/0.0050**
BSD100	0.03/0.0007	0.07/0.0025	**0.12/0.0043**
Urban100	0.07/0.0012	0.13/0.0047	**0.18/0.0089**

are selected. The detailed architecture settings are summarised in Table V. Note that L20D4 has an overlarge receptive field, and we accordingly change the size of training samples from 80×80 to 95×95 when training this model. The resultant performances of each model on the four test sets are illustrated in Figs. 5 and 6.

In the case of L6 models (Fig. 5), we observe similar effects to the case of L10 models, which state that L6D2 is the best model for both $\times 2$ and $\times 3$ tasks, while L6D3 is the best for $\times 4$. The overall performances of L6 models are slightly lower than that of L10 models due to insufficient nonlinear layers. A notable phenomenon is that the performance of L6D4 is worse, by a noticeable margin, than that of L6D0 in all datasets and all SR tasks, which is different from the case of L10. We foretell that increasing the receptive field size can involve much contextual information for SR, but a model of high nonlinearity is required. Apparently, the nonlinearity of L6 models are not strong enough to handle such a case.

Regarding the L20 models, Fig. 6 shows that the performance differences between L20D0, L20D2, and L20D3 are marginal. We notice that the results of PSNR and SSIM are not always consistent. For example, in $\times 2$ task, L20D2 achieves the best average PSNR while L20D0 has the best average SSIM; in $\times 3$ task, L20D3 and L20D2 are the best for PSNR and SSIM, respectively. This inconsistency may be caused by the training loss, *i.e.* MSE, which prefers high PSNR instead of high SSIM. L20D4 with an overlarge receptive field fails to produce a higher performance compared with L20D0.

We compute the values of the improved quality $\Delta_{\text{L}m\text{D0}}\mathcal{P}_{\text{best}}$ for both L6 and L20, which are listed in Tables VI and VII, respectively. The best models for L20 are selected according to the PSNR metric (since the improvement is not significant, we do not mark the best values). By comparing the improved qualities for L6, L10, and L20, we find that the receptive field size is an important factor for SISR especially when the model depth is not sufficient. While the model is very deep, the receptive field size characterised by the stack of 3×3 filters may be saturated, in which case the performance cannot be substantially improved by increasing the size further.

TABLE VII
COMPARISON OF IMPROVED QUALITY $\Delta_{\text{L20D0}}\mathcal{P}^s_{\text{BEST}}$ FOR THE L20 MODELS

	$\Delta_{\text{L20D0}}\mathcal{P}^2_{\text{L20D2}}$	$\Delta_{\text{L20D0}}\mathcal{P}^3_{\text{L20D3}}$	$\Delta_{\text{L20D0}}\mathcal{P}^4_{\text{L20D4}}$
Set5	0.08/0.0000	0.05/0.0000	0.07/0.0012
Set14	0.03/-0.0001	0.00/-0.0005	0.00/0.0011
BSD100	0.01/-0.0004	0.01/-0.0004	0.02/0.0011
Urban100	-0.01/-0.0006	-0.02/-0.0005	0.02/0.0026

C. An Instance-Level Investigation

While the above analyses concern about the effects of receptive field size on different up-scaling factors, we next investigate the effects on each instance of BSD100 and Urban100. This investigation is to explore what kind of images benefits more from increasing the receptive field size. In specific, we employ the *local entropy* to characterise the type of images. As introduced by Kadir and Brady [57], given a local region R of a grey-level image and the pixel descriptor $D \in \{d_1, ..., d_n\}$, the local entropy is calculated as

$$H_{D,R} = -\sum_i P_{D,R}(d_i) \log_2 P_{D,R}(d_i), \tag{7}$$

where $P_{D,R}(d_i)$ is the probability of the descriptor D taking the value d_i in the local region R. This quantity has nice properties to distinguish different kinds of regions in an image; for example, texture, edge, and flat regions have high, moderate, and low values of the local entropy, respectively, which can be observed from the examples in Fig. 8. We regard the average local entropy over the whole image, denoted by $\bar{H}_D$, as an indicator of whether the image has more textures, more edges, or more flat areas. Then, the goal is to find the correlation between the average local entropy and the improved quality by increasing the receptive field size. In each SISR task, the value of the pair $(\Delta_{\text{L10D0}}\mathcal{P}^s_{\text{best}}, \bar{H}_D)$ is calculated for each image in BSD100 and Urban100 (specifically, PSNR is calculated). The results in Fig. 7 illustrate that for all tasks, $\Delta_{\text{L10D0}}\mathcal{P}^s_{\text{best}}$ has scattered large values when $\bar{H}_D$ is low. With the increase of $\bar{H}_D$, $\Delta_{\text{L10D0}}\mathcal{P}^s_{\text{best}}$ tends to shrink to small values (even zero). This implies that increasing the receptive field size works well in the flat and edge regions that possess low local entropy values, but it is limited to tackle highly textured image regions.

We select a set of images that achieve the highest and the lowest values of $\Delta_{\text{L10D0}}\mathcal{P}^s_{\text{best}}$, as shown in Fig. 8. The bad examples contain many textures in the residual images that are not recovered, and it is worth noting that the residual images are highly correlated with the local entropy images. One reason for the failure is that it is difficult to model the complex structures in texture areas by using limited information in LR images. Another reason is the training loss, *i.e.* MSE. As indicated in [46], the solution induced by MSE loss is a pixel-wise average of possible HR patches on a natural image manifold, leading to blurry effects in the restored images.

Remarks on Receptive Field Size:

- Within a certain range, increasing the receptive field size can always help improve the SISR performance.
- Receptive field size is an important factor which can compensate the deficiency caused by insufficient model nonlinearity.
- A larger receptive field size is required when the image degradation gets severer.

V. EFFECTS OF MODEL DEPTH ON SISR

This section focuses on detailed analysis of how model depth affects the SISR performance. To make a fair comparison, the models used in this section have the same receptive field size. For this purpose, we propose to employ the convolutional layer with 1×1 filter size, which is an effective way to arbitrarily increase the model nonlinearity without changing the receptive field size. In the following, the convolutional layer with $k \times k$ filter size is denoted by $k \times k$ convolutional layer for concise presentation.

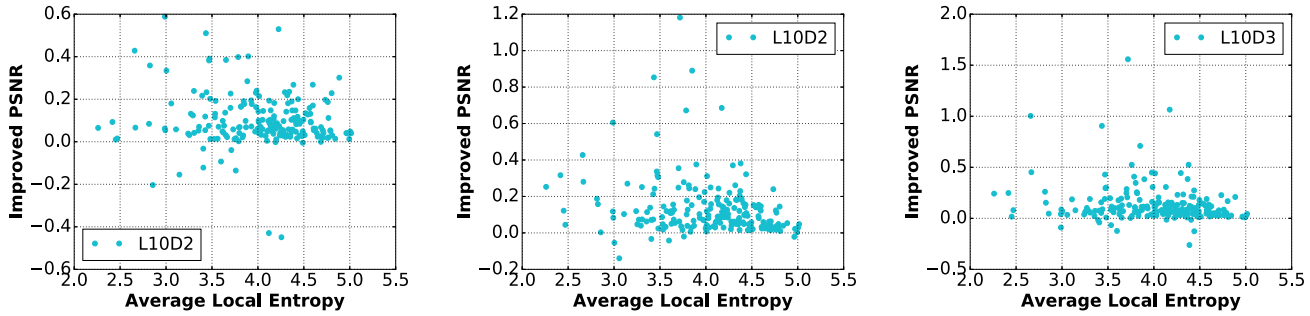
To begin with, we modify the model notations to incorporate with the newly introduced 1×1 convolutional layers. Specifically, the notation LmDnPo is used to denote a CNN model with $m+1$ 3×3 convolutional layers and o 1×1 convolutional layers, among which there are n DC layers. The total number of convolutional layers in this model is $m+o+1$. Similar to the settings in previous section, all convolutional layers except the reconstruction layer employ 64 filters. The setting of dilation factor follows Table II.

One concern about designing a LmDnPo model is where to insert the 1×1 convolutional layers. Existing studies [6], [30], [40] inform us that a SISR process is a combination of LR feature encoding and HR feature decoding. Intuitively, in the architecture depicted by Fig. 2, we regard that the convolutional layers close to the input focus on LR feature encoding, while the convolutional layers close to the output are responsible for HR feature decoding. Since HR feature space is more complex than LR feature space, more nonlinear activation functions (*i.e.* ReLU) are required for HR feature analysis. In this sense, the CNN models are designed by inserting the 1×1 convolutional layers after the L 3×3 convolutional layers. Such a design allows that the contextual information in the receptive field can be completely extracted before the 1×1 convolutions.

To analyse the effects of model depth, we are particularly interested in two cases, including the models with and without sufficient receptive field size, which are discussed separately.

A. Insufficient Receptive Field Size

The experiments in this subsection is to examine whether it is useful to apply more nonlinear computations on limited contextual information. We select different models in different SISR tasks to ensure that the receptive field size in each task is insufficient. Specifically, according to the results in Section IV, we select L6D0 (or L6D0P0) as the basic model in $\times2$ task, which is compared with L6D0P4 and L6D0P14. L10D0 (or L10D0P0) is used as the basic model in both $\times3$ and $\times4$ tasks, which is compared with L10D0P5 and L10D0P10. We use L6* instead of L10* for $\times2$ such that the condition of insufficient receptive field size can be fulfilled. The receptive field sizes and model depths are summarised in Table VIII.



(a) $\times 2$ (b) $\times 3$ (c) $\times 4$

Fig. 7. Improved PSNR $\Delta_{\text{L10D0}}\mathcal{P}^{s}_{\text{best}}$ *v.s.* average local entropy for each image in BSD100 and Urban100. In each case, the best model (L10D2 for $\times 2$, L10D2 for $\times 3$, and L10D3 for $\times 4$) is used for evaluation.

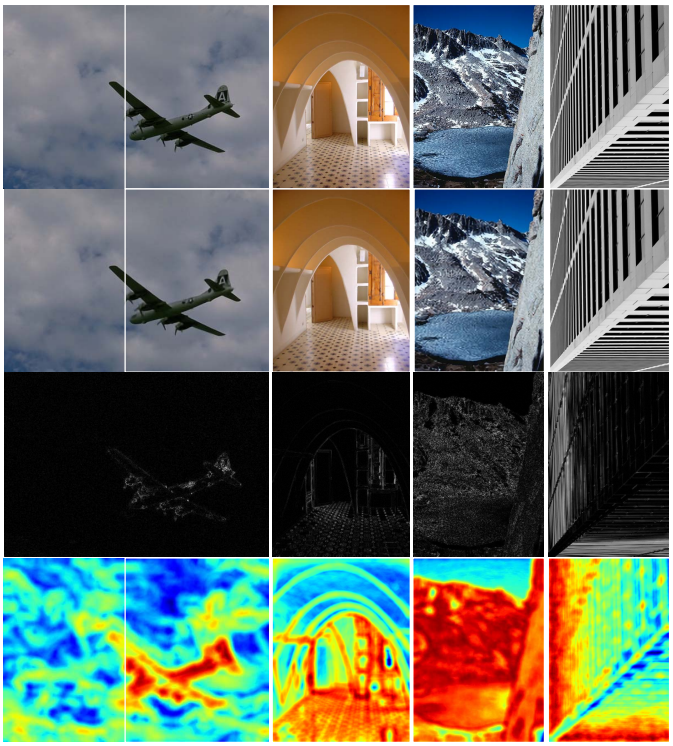


Fig. 8. SR examples of L10D2 for $\times 3$. From top to bottom: original HR image, restored image, residual image, and local entropy image. The left two columns are the examples that achieve the highest $\Delta_{\text{L10D0}}\mathcal{P}^{3}_{\text{L10D2}}$ in BSD100 and Urban100, respectively. The right two columns achieve the lowest $\Delta_{\text{L10D0}}\mathcal{P}^{3}_{\text{L10D2}}$ in the two datasets.

The comparison of the performances produced by different models is exhibited in Fig. 9. In the case of $\times 2$, it is noticed that L6D0P4 performs slightly better than L6D0P0 and L6D0P14. Similarly in the $\times 3$ task, L10D0P5 achieves the best performance among the others. In the case of $\times 4$, L10D0P5 and L10D0P10 perform almost the same, and are both better than L10D0P0. Overall, the largest improvements *w.r.t.* L*m*D0P0 in $\times 3$ and $\times 4$ exceed the largest improvements in $\times 2$. But the overall improvements in all cases are marginal and not very attractive.

These results suggest that given a model with insufficient receptive field size, increasing the model depth only slightly improves the SISR performance. An over-deep model without viewing adequate contextual information can even produce degraded performance (see L6D0P14 in $\times 2$ and L10D0P10 in $\times 3$). It is possibly due to that a long stack of 1×1 convolutional layers hinders the training process to produce a good model. In addition, when the information is lost significantly as in $\times 4$, a deep architecture is preferred. This is reasonable because the LR space structure is simple, whereas the HR space structure is complex. The significant differences between LR and HR spaces result in a highly nonlinear point-to-point mapping function from LR to HR spaces.

TABLE VIII
ARCHITECTURE SETTINGS OF THE L*m*D0P*o* CNN MODELS

Model	Receptive Field Size	Depth
L6D0P0	15	7
L6D0P4	15	11
L6D0P14	15	21
L10D0P0	23	11
L10D0P5	23	16
L10D0P10	23	21

TABLE IX
ARCHITECTURE SETTINGS OF THE L6D*n*P*o* CNN MODELS

Model	DC positions	Receptive Field Size	Depth
L6D2P0	3, 4	23	7
L6D2P4	3, 4	23	11
L6D2P14	3, 4	23	21
L6D3P0	3, 4, 5	37	7
L6D3P4	3, 4, 5	37	11
L6D3P14	3, 4, 5	37	21

B. Sufficient Receptive Field Size

In this subsection, we investigate how the model depth affects the SISR performance given the models with sufficient views of contextual information. We first set up an experiment by using the shallow basic models, including L6D2P0 for $\times 2$ task and L6D3P0 for $\times 3$ and $\times 4$ tasks. Deeper models are constructed by stacking 4 and 14 1×1 convolutional layers. The model details can be found in Table IX.

The second experiment is to investigate whether the best L10 model in each SR task can be further improved

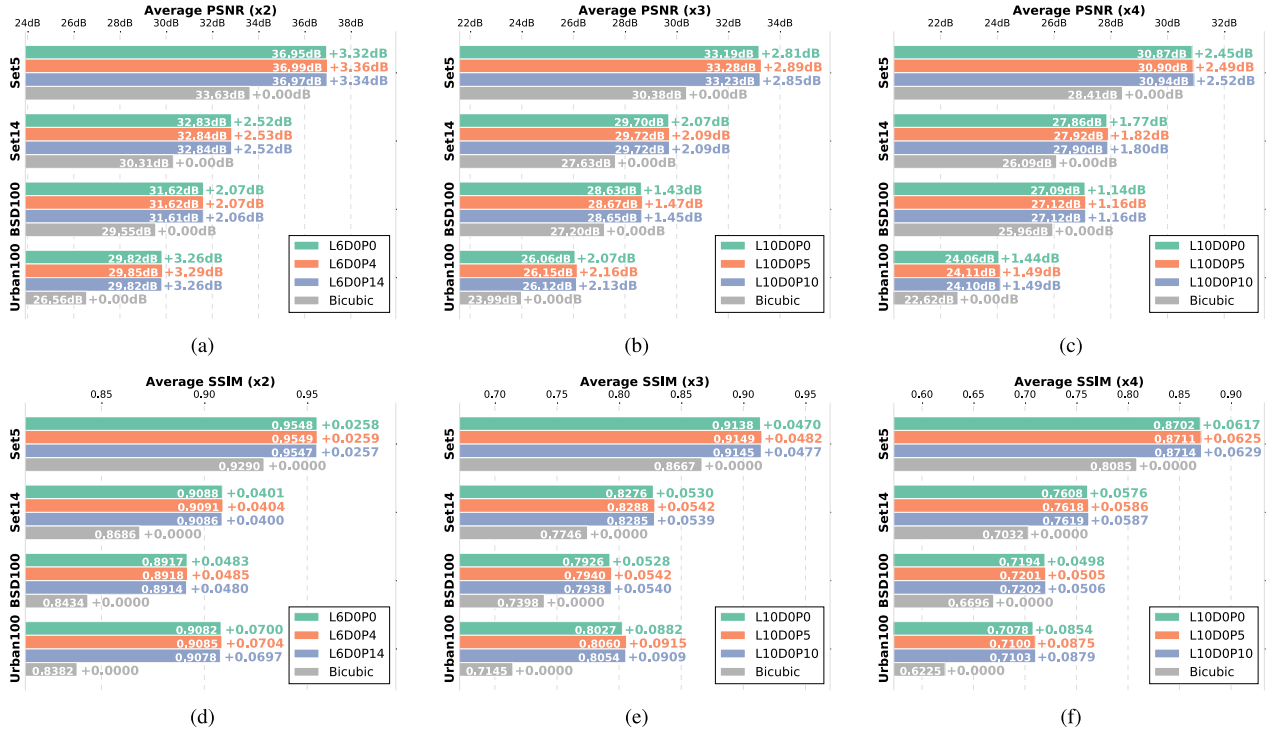


Fig. 9. The SISR performances of the LmD0Po models for the investigation of model depth under the setting of insufficient receptive field size.

TABLE X
ARCHITECTURE SETTINGS OF THE L10DnPo CNN MODELS

Model	DC positions	Receptive Field Size	Depth
L10D0P0	-	23	11
L10D0P5	-	23	16
L10D0P10	-	23	21
L10D2P0	6, 7	31	11
L10D2P5	6, 7	31	16
L10D2P10	6, 7	31	21
L10D3P0	6, 7, 8	45	11
L10D3P5	6, 7, 8	45	16
L10D3P10	6, 7, 8	45	21

by increasing model depth. The basic models used here are L10D0P0 for $\times 2$ task, L10D2P0 for $\times 3$ task, and L10D3P0 for $\times 4$ task. We do not use the L20 models because our experiences indicate that an over-deep model (*e.g.* with more than 30 layers) takes very long time for training. As previously, we choose two policies to increase the depth, which are by adding 5 and 10 1×1 convolutional layers. The corresponding architecture settings are detailed in Table X.

Figs. 10 and 11 illustrates the obtained performances, showing that only the $\times 2$ task can benefit slightly from increasing the depth of L10D0P0. All other cases fail to show noticeable promotion of performance when adding the 1×1 convolutional layers.

An explanation for the above phenomenon is that the performances of the selected basic models may have reached the limit, and the induced nonlinearity is competent to analyse the contextual information viewed by the models. By comparing the values in Figs. 11 and 4, it is easily seen that a L10Dn model can be improved more by enlarging receptive field than by increasing depth.

Remarks on Model Depth:

- It is not helpful for SISR to simply increase the model depth without changing the receptive field size.

VI. DISCUSSION

While the analyses in previous sections perhaps illustrate that the receptive field size is more important than the model depth, we need to emphasise that the two factors are mutualistic within a certain range of values. That is, the larger the context region is, the stronger nonlinearity the model requires, and *vice versa*.

To jointly evaluate the influences of different receptive field sizes and model depths, we test the previously trained models on BSD100 and Urban100. The acquired average PNSR results are listed in Tables XI, XII, and XIII. Since there would be an extremely large amount of work to exhaustively evaluate all combinations of the pairs (receptive field size, model depth), we only list the performances of the previously selected models and leave the rest as "–". As seen in the tasks, the differences between the four values along each row where applicable are no larger than 0.11dB, and mostly around 0.02dB. But when comparing the values along the columns, a large variation can be noticed. The best values in all tasks are produced by the L20 models with appropriate receptive field sizes, and the $\times 4$ task requires larger receptive field than the $\times 2$ and $\times 3$ tasks. We also note that the performance of a L10 model can approach to that of the best L20 model. For example, in $\times 2$ the best L10 is less than the corresponding best L20 by 0.13dB; in $\times 3$ the best L10 is less than the best

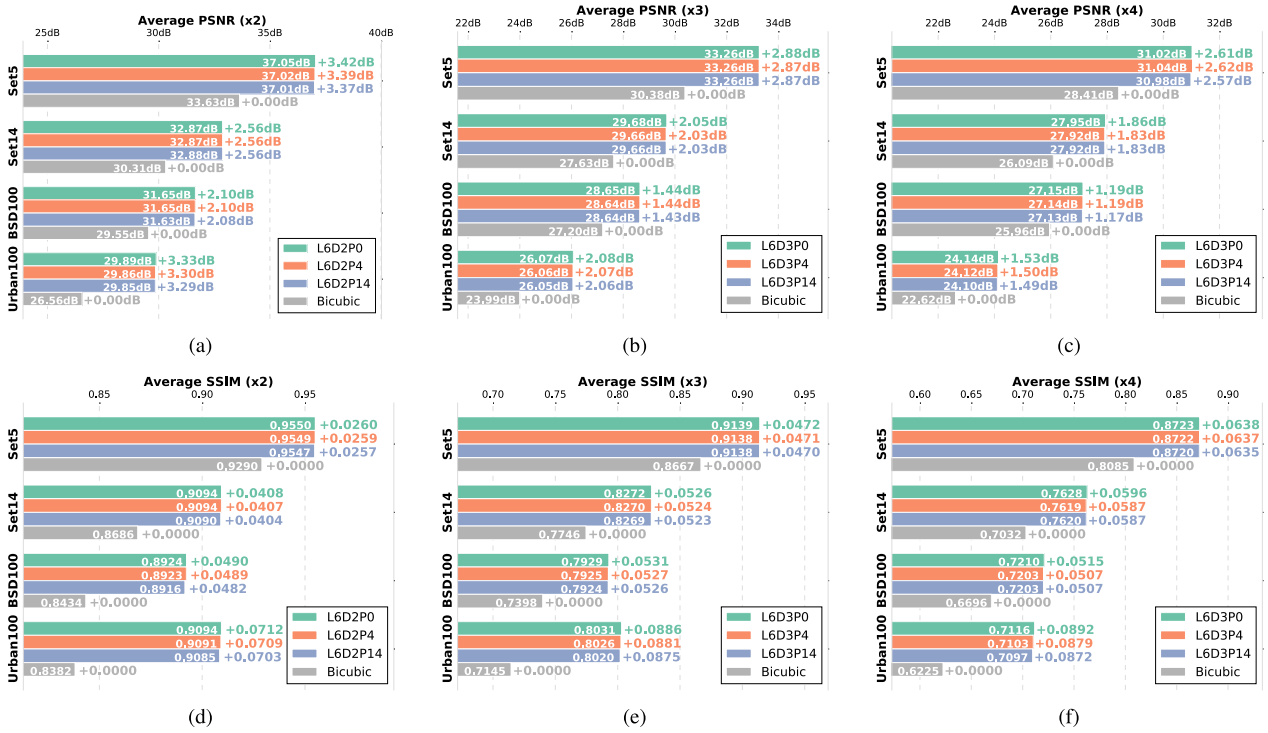


Fig. 10. The SISR performances of the L6D*n*P*o* models for the investigation of model depth under the setting of sufficient receptive field size.

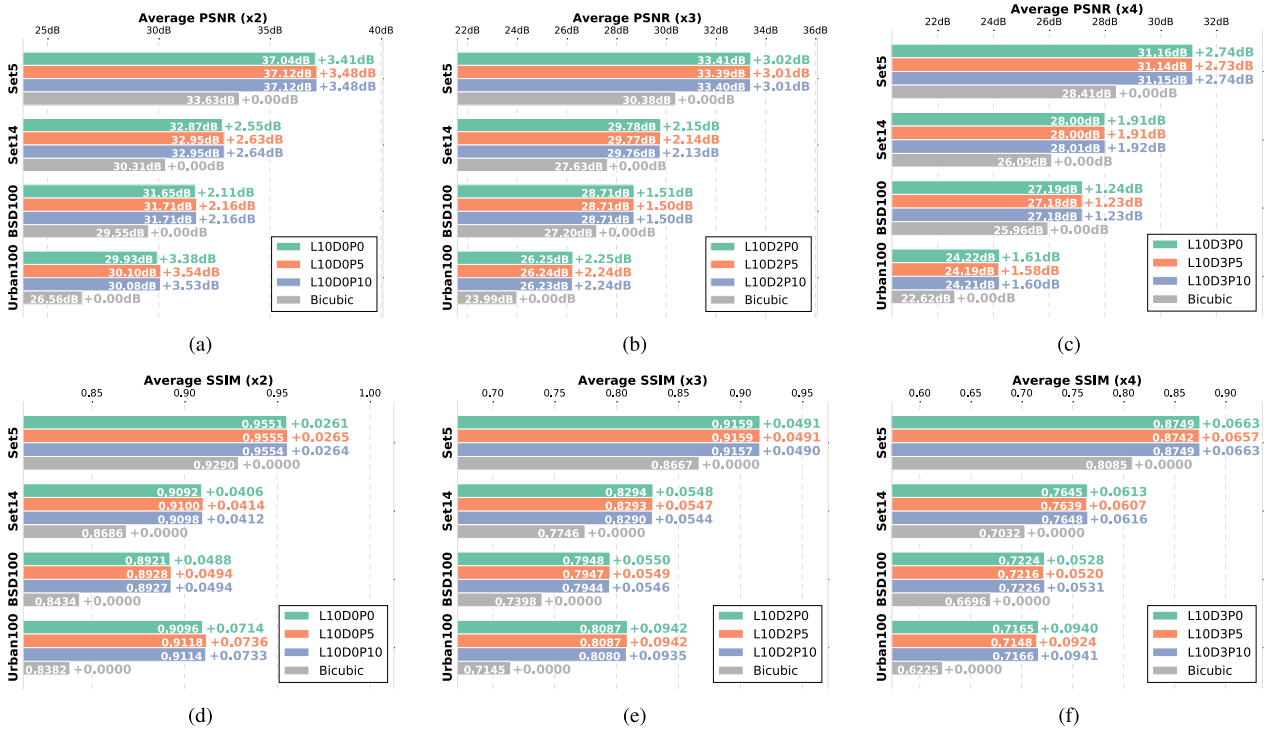


Fig. 11. The SISR performances of the L10D*n*P*o* models for the investigation of model depth under the setting of sufficient receptive field size.

L20 by 0.05dB; and in $\times 4$ the best L10 is less than the best L20 by 0.03dB. We have also conducted significance tests using the paired t-test to statistically evaluate the significance of the performance difference between the best performers and the others in each depth setting and in each RFS setting. The results indicate that in almost all settings, the best performer is significantly better than the others at level $p < 0.05$.

According to these results as well as the previous analyses, we can **conclude** the relationship between receptive field size and model depth *w.r.t* SISR, which states:

TABLE XI

AVERAGE PSNRS (dB) (×2) USING DIFFERENT RECEPTIVE FIELD SIZES (RFS) AND MODEL DEPTHS. THE TOP 30% VALUES ARE MARKED BY BOLD. THE TOP THREE VALUES ARE MARKED BY RED, GREEN, AND BLUE, RESPECTIVELY

RFS \ Depth	7	11	16	21
15	30.72	30.73	-	30.71
23	30.77	30.79	**30.90**	**30.89**
31	-	**30.88**	-	-
37	30.66	-	-	-
43	-	-	-	**31.01**
45	-	30.70	-	-
51	-	-	-	**31.01**
65	-	-	-	**31.00**
67	30.14	-	-	-
75	-	30.77	-	-
95	-	-	-	30.65

TABLE XII

AVERAGE PSNRS (dB) (×3) USING DIFFERENT RECEPTIVE FIELD SIZES (RFS) AND MODEL DEPTHS

RFS \ Depth	7	11	16	21
15	27.27	-	-	-
23	27.38	27.35	27.41	27.39
31	-	**27.48**	**27.47**	**27.47**
37	27.36	27.35	-	27.34
43	-	-	-	**27.53**
45	-	**27.45**	-	-
51	-	-	-	**27.53**
65	-	-	-	**27.52**
67	27.04	-	-	-
75	-	27.41	-	-
95	-	-	-	27.32

TABLE XIII

AVERAGE PSNRS (dB) (×4) USING DIFFERENT RECEPTIVE FIELD SIZES (RFS) AND MODEL DEPTHS

RFS \ Depth	7	11	16	21
15	25.50	-	-	-
23	25.61	25.58	25.61	25.61
31	-	25.69	-	-
37	25.65	25.63	-	25.62
43	-	-	-	**25.72**
45	-	**25.71**	25.69	**25.70**
51	-	-	-	**25.74**
65	-	-	-	**25.74**
67	25.44	-	-	-
75	-	25.68	-	-
95	-	-	-	25.61

- Model depth is an essential element for SISR to produce a top performance, which must cooperate with a proper receptive field size.
- Given a fixed receptive field size, an optimal model depth exists which can produce a top performance. Further increasing the depth based on the optimal value is not helpful. But given a fixed model depth, the performance of the model is sensitive to the size of receptive field.
- When the image degradation is severe, a large receptive field is required, and otherwise, a large depth is expected.
- The restoration of flat and edge regions can benefit more from increasing receptive field size, whereas the restoration of textures benefits less, given that the training objective is MSE.

From the perspective of image contexts and structures, we understand that in flat or edge areas, a HR pixel is highly correlated with the pixels in its surrounding region or along a specific direction. If the degradation of the nearby region is not serious, the HR pixel can be easily estimated based on the surrounding LR pixels by using a model with appropriate complexity. But if the degradation is severe, especially when the nearby LR pixels become dissimilar to the HR pixel, a large region of context information is expected, and a highly nonlinear restoration function is potentially required. On the other hand, in texture areas, the local structures are always complex. In such a case, we can expect a model with large receptive field size and high nonlinearity to exploit regular patterns in the texture region, which may produce a good performance. But if the textures are irregular, the model will inevitably fail, especially when the degradation is serious.

The above discussions provide us a guide of improving the performance of existing deep learning-based SISR methods. For example, given an existing SISR model, we could enlarge its receptive field size by stacking the dilated convolutional layers, which could result in improvement of performance. On the other hand, if an efficient model is required, a good option is to reduce the model depth but keeping the receptive field size unchanged by using dilated convolution. This can reduce the parameter number of the model without having a significant performance drop.

The issue of texture restoration is also a bottleneck encountered by modern SISR techniques that are based on the MSE training loss. Even though high PSNR values can be obtained (possibly due to the good estimation of flat and edge regions), the resultant textures generally exhibit blurry effects and are thus visually unpleasing. The very recent SISR researches begin to focus on developing perceptual losses by which the HR estimation exhibits clear textures. Even though the restored textures are sometimes not similar to those in the original HR image, the resultant image becomes more perceptually acceptable. While the studies in this paper are based on MSE, the empirical findings are consistent with our intuitive understanding, and are applicable for designing perceptual loss-based architectures.

We finally compare the dilated L10 and L20 models with SRCNN [2], CSCN [44], VDSR [3], FSRCNN [40], and LapSRN [58]. The dilated model is named as dilated convolutional network for SR (DCNSR). The average performances are summarised in Table XIV. As shown, the DCNSR_{20} model is among the best performers, indicating that enlarging the receptive field size would help improve the performance. The DCNSR_{10} model performs well and yields high efficiency due to the reduction of its parameter number.

TABLE XIV
AVERAGE PERFORMANCES OF DIFFERENT METHODS ON SET5, SET14, BSD100, AND URBAN100

Dataset	Scale	Bicubic	SRCNN	CSCN	VDSR	FSRCNN	LapSRN	$DCNSR_{10}$	$DCNSR_{20}$
Set5	×2	33.66/0.9299	36.66/0.9542	36.93/0.9552	37.53/0.9587	37.00/0.9558	37.53/0.9592	37.09/0.9553	37.56/0.9599
	×3	30.39/0.8682	32.75/0.9090	33.10/0.9144	33.66/0.9213	33.16/0.9140	-	33.41/0.9159	33.70/0.9120
	×4	28.42/0.8104	30.48/0.8628	30.86/0.8732	31.35/0.8838	30.71/0.8657	31.54/0.8854	31.07/0.8740	31.43/0.8846
Set14	×2	30.24/0.8688	32.42/0.9063	32.56/0.9074	33.03/0.9124	32.63/0.9088	32.98/0.9126	32.95/0.9099	33.04/0.9124
	×3	27.55/0.7742	29.28/0.8209	29.41/0.8238	29.77/0.8314	29.43/0.8242	-	29.78/0.8294	29.82/0.8326
	×4	26.00/0.7027	27.49/0.7503	27.64/0.7587	28.01/0.7674	27.59/0.7535	28.09/0.7701	28.00/0.7643	28.04/0.7681
BSD100	×2	29.56/0.8431	31.36/0.8879	31.40/0.8884	31.90/0.8960	31.50/0.8909	31.80/0.8954	31.71/0.8929	31.98/0.8984
	×3	27.21/0.7385	28.41/0.7863	28.50/0.7885	28.82/0.7976	28.52/0.7900	-	28.71/0.7948	28.94/0.8000
	×4	25.96/0.6675	26.90/0.7101	27.03/0.7161	27.29/0.7251	26.96/9,7139	27.32/0.7276	27.17/0.7218	27.31/0.7266
Urban100	×2	26.88/0.8403	29.50/0.8946	29.27/0.9031	30.76/0.9140	29.49/0.9035	30.03/0.9129	30.06/0.9117	30.75/0.9137
	×3	24.46/0.7349	26.24/0.7989	25.77/0.7961	27.14/0.8279	25.84/0.7945	-	26.25/0.8087	27.10/0.8221
	×4	23.14/0.6577	24.52/0.7221	23.96/0.7065	25.18/0.7524	23.91/0.6976	24.39/0.7263	24.21/0.7150	25.26/0.7536

VII. CONCLUSION

In this paper, we are towards trying to answer the question: does SISR require a complex function or a wide region, or both, to produce good estimations of HR pixels? Accordingly we conduct a comprehensive investigation on two key factors of a deep architecture, namely receptive field size and model depth, which may affect the SISR performance. Specifically, it is proposed to employ the dilated convolutional layers which can enlarge the receptive field size without changing the model depth. Our findings state that given the model depth fixed, the SISR performance is sensitive to the selection of the receptive field size, and that the more seriously the image is degraded, the larger receptive field size is required. In the studies on model depth, we propose to use 1×1 convolutional layers to strengthen the model nonlinearity. It is observed that there is no noticeable improvement on performance by simply stacking more convolutional layers without extending the receptive field. But in an investigation of the joint effects between receptive field size and model depth, we find that a deep architecture is necessary to produce the top performance, while the optimal receptive field size should be congruent with the model depth. It is also noticed that the model depth has more influence on performance when the image degradation is mild, and the receptive field size is more important when the degradation is severe. The proposed strategy of dilated convolution can also help reduce model depth while maintaining desirable performances, thus saving the computation cost. Further work will focus on exploiting more combining ways of dilated convolution layers to design more efficient SISR networks.

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*. Springer, 2014, pp. 184–199.

[2] 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. 2015.

[3] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, 2016, pp. 1646–1654.

[4] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. CVPR*, 2016, pp. 1637–1645.

[5] Y. Wang, L. Wang, H. Wang, and P. Li, "End-to-end image super-resolution via deep and shallow convolutional networks," 2016, *arXiv:1607.07680*. [Online]. Available: https://arxiv.org/abs/1607.07680

[6] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. CVPR*, 2016, pp. 1874–1883.

[7] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," in *Proc. ICLR*, 2016, pp. 1–17.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015, *arXiv:1512.00567*. [Online]. Available: https://arxiv.org/abs/1512.00567

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[11] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016, pp. 694–711

[12] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. NIPS*, 2014, pp. 2654–2662.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[14] S. Mallat, *A Wavelet Tour of Signal Processing*. 1999.

[15] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets*. Marseilles, France: CPT, CNRS lUMINY, 1990, pp. 286–297.

[16] M. Shensa, "The discrete wavelet transform: Wedding the a trous and Mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[17] K. Nasrollahi and T. B. Moeslund, "Super-resolution: A comprehensive survey," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1423–1468, 2014.

[18] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *Proc. ECCV*. Springer, 2014, pp. 372–386.

[19] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[20] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[21] A. Marquina and S. J. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, Dec. 2008.

[22] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Proc. CVPR*, 2008, pp. 1–8.

[23] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, "Super resolution using edge prior and single image detail synthesis," in *Proc. CVPR*, 2010, pp. 2400–2407.

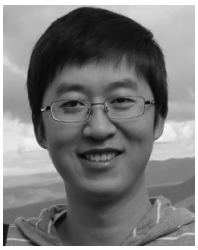
[24] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 95-1–95-8, Jul. 2007.

[25] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 36–51, Jan. 2009.

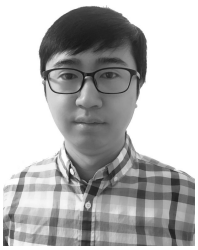
[26] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. ICCV*, 2009, pp. 349–356.

[27] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. CVPR*, 2015, pp. 5197–5206.

[28] M. E. Tipping and C. Bishop, "Bayesian image super-resolution," in *Proc. NIPS*, 2003, pp. 1303–1310.
[29] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational Bayesian super resolution," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 984–999, Apr. 2011.
[30] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[31] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3467–3478, Aug. 2012.
[32] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *Proc. CVPR*, 2013, pp. 345–352.
[33] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. CVPR*, vol. 1, 2004, p. 1.
[34] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. ICCV*, 2013, pp. 561–568.
[35] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.
[36] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.
[37] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. ICCV*, 2013, pp. 1920–1927.
[38] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. ACCV*. Springer, 2014, pp. 111–126.
[39] Z. Wang *et al.*, "Self-tuned deep super resolution," in *Proc. CVPR Workshops*, 2015, pp. 1–8.
[40] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. ECCV*. Springer, 2016, pp. 391–407.
[41] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Proc. ECCV*. Springer, 2014, pp. 49–64.
[42] R. Wang and D. Tao, "Non-local auto-encoder with collaborative stabilization for image restoration," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2117–2129, May 2016.
[43] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, "Coupled deep autoencoder for single image super-resolution," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 27–37, Jan. 2016.
[44] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. ICCV*, 2015, pp. 370–378.
[45] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3194–3207, Jul. 2016.
[46] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," 2016, *arXiv:1609.04802*. [Online]. Available: https://arxiv.org/abs/1609.04802
[47] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszär, "Amortised map inference for image super-resolution," 2016, *arXiv:1610.04490*. [Online]. Available: https://arxiv.org/abs/1610.04490
[48] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2207–2232, Sep. 1992.
[49] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–13.
[50] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," 2016, *arXiv:1606.00915*. [Online]. Available: https://arxiv.org/abs/1606.00915
[51] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. CVPR*, 2015, pp. 3791–3799.
[52] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. BMVC*, 2012, pp. 1–10.
[53] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surfaces*. Springer, 2010, pp. 711–730.
[54] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.
[55] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACMMM*, 2014, pp. 675–678.
[56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[57] T. Kadir and M. Brady, "Saliency, scale and image description," *Int. J. Comput. Vis.*, vol. 45, no. 2, pp. 83–105, 2001.
[58] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. CVPR*, 2017, pp. 624–632.



Ruxin Wang received the B.Eng. degree from Xidian University, the M.Sc. degree from the Huazhong University of Science and Technology, and the Ph.D. degree from the University of Technology Sydney. He is currently a Research Scientist with Union Visual Innovation Technology Company, Ltd., Shenzhen, Guangdong, China. He has authored and coauthored over ten research articles, including IEEE T-NNLS, T-IP, and T-Cyb. His research interests include deep learning, image restoration, and computer vision.



Mingming Gong is currently a Lecturer (Assistant Professor) with the School of Mathematics and Statistics, The University of Melbourne. He has authored and coauthored 30+ research articles, including NeurIPS, ICML, UAI, AISTATS, CVPR, ICCV, ECCV, and AAAI. His research interests include causal inference, machine learning, and computer vision. He has studied how the causal generative process of data benefits learning in non-standard settings, such as transfer learning and weakly supervised learning. He also studies principles and methods to infer causal models from various kinds of observational data, including under-sampled time series, data with measurement error, nonstationary, and heterogenous data.

Dacheng Tao (F'15) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney. His research results in artificial intelligence have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, IJCV, JMLR, AAAI, IJCAI, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He was a recipient of the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Scopus-Eureka prize. He is a fellow of the Australian Academy of Science.