FC^{2}N: Fully Channel-Concatenated Network for Single Image Super-Resolution

Xiaole Zhao[1], Ying Liao[1], Tian He[1], Yulun Zhang[2], Yadong Wu[3], and Tao Zhang[1]

[1] School of Life Science and Technology, UESTC, Chengdu, China
zxlation@foxmail.com; tao.zhang@uestc.edu.cn
[2] Department of ECE, Northeastern University, Boston, USA
yulun100@gmail.com
[3] Sichuan University of Science and Engineering, Yibin, China

Abstract. Most current image super-resolution (SR) methods based on convolutional neural networks (CNNs) use residual learning in network structural design, which favors to effective back propagation and hence improves SR performance by increasing model scale. However, residual networks suffer from representational redundancy by introducing identity paths that impede the full exploitation of model capacity. Besides, blindly enlarging network scale can cause more problems in model training, even with residual learning. In this paper, a novel fully channel-concatenated network (FC^{2}N) is presented to make further mining of representational capacity of deep models, in which all interlayer skips are implemented by a simple and straightforward operation, weighted channel concatenation (WCC), followed by a 1×1 conv layer. Based on the WCC, the model can achieve the *joint* attention mechanism of linear and nonlinear features in the network, and presents better performance than other state-of-the-art SR models with fewer model parameters. To our best knowledge, FC^{2}N is the first SR model that does not adopt residual learning and reaches network depth over **400** layers. Moreover, it shows excellent performance in both largescale and lightweight implementations, which illustrates the full exploitation of the representational capacity of the model.

Keywords: Image Super-Resolution, Non-residual Learning, Weighted Channel Concatenation, Global Feature Fusion

1 Introduction

Single image super-resolution (SISR) is a classic problem in low-level computer vision that aims at reconstructing a high-resolution (HR) image from one single low-resolution (LR) image. Although a lot of solutions have been proposed for image SR, it is still an active yet challenging research topic in computer vision community due to its ill-poseness nature and high practical values [3,47,51].

In recent years, deep learning techniques [25], especially convolutional neural networks (CNNs) [26,27] and residual learning [12], have significantly promoted the advance of image SR. A representative work that successfully adopts CNNs to SR problem is SRCNN [7], which is a three-layer CNN network that can learn

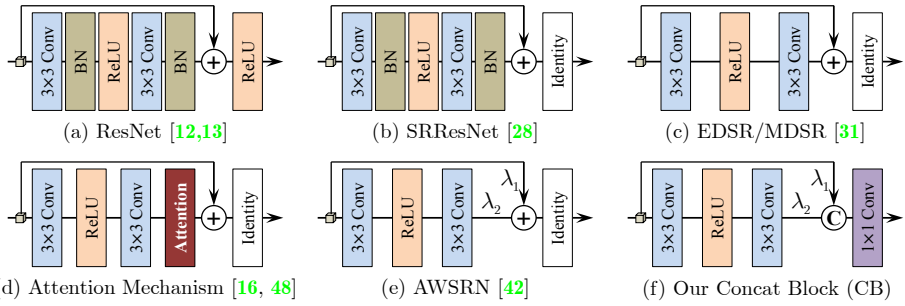


Fig. 1. Typical building blocks. "Identity" and "Attention" represent identity mapping and attention mechanism respectively, and "**C**" denotes channel concatenation.

an end-to-end mapping between LR and HR images and achieve satisfactory SR performance at that time. Subsequently, many studies were conducted to design and build more accurate and efficient SR networks, such as [22], [38], [21], [51], [31], [49], [48] etc. One of the major trends in these models is models get larger and deeper for further performance gain. As model scale increases, the training difficulty caused by information weakening and over/underfitting becomes more serious, and more tricks are needed to ensure an effective training [29]. Residual learning [12] is probably one of the most commonly-used techniques to ease this training difficulty, which is a simple element-wise addition of features at different layers. It can help extend deep SR models to previously unreachable depths and capabilities by introducing identity paths that carries gradients throughout the extent of very deep models [41]. These identity paths, however, result in a large amount of representational redundancy in deep residual networks [18], hindering the full mining of model capabilities.

To make full use of features and further mine the representational capacity of deep networks, many SR models tend to combine residual learning with channel concatenation, which is believed to help new feature exploration and learning good representations [17,5,16,51], e.g., MemNet [39], AWSRN [42], RDN [49], MSRN [29], CARN [2], CSSFN [50] and DBDN [43] etc. On the one hand, most of these models utilize residual learning [12,13] for stable and effective model training, but as mentioned above, it is not conducive to full exploration of model capacities. On the other hand, channel concatenation in these models is adopted to directly connect different layers, which ignores the contribution of adaptive connection strength to model capacities, i.e., weighted channel concatenation (WCC). Although the weighted interlayer connections are considered to impede effective back propagation in case of residual learning [13], we believe that they are more in line with the manner neurons behave in the human brain, therefore more physiologically sound.

Considering these problems, we present a novel architecture for single image SR tasks in this work, in which all skip connections are simply implemented by channel concatenation without any residual connection. Each branch of these channel concatenations is attached by a weighting factor to further explore the representational capacity of the model. As shown in Fig.1 and Fig.2, we construct

the building modules of our model by this WCC, including concat block (CB) for effective local feature utilization and concat group (CG) for the ease of model training. Moreover, we improve the global feature fusion (GFF) [49] by weighting the branches of GFF's skip connections, i.e., WGFF. These structural ingredients built with our WCC can combine linear and nonlinear features from fine to coarse grained, allowing us to build deeper and more expressive models without residual learning. Since all interlayer skips are conducted by channel concatenation, the proposed model is terms as fully channel-concatenated network (FC^{2}N).

In summary, the main contributions of this work are four-fold: (1) We present a novel FC^{2}N for efficient and accurate image SR, in which channel concatenation is used to conduct all interlayer skips. It can reach much deeper without residual learning, and gives better performance boosting with fewer parameters. (2) We propose a new interlayer connection, i.e., WCC, for efficient feature fusion. It is simple and can be easily embedded into other models for performance gain. We demonstrate that residual block [31] and weighted residual block [42] are special cases of our CB. (3) We improve the GFF strategy [49] by our WCC, exploring a more efficient WGFF that makes the utilization of hierarchical features more flexible and reasonable, thus promoting the model performance. (4) We construct a CG module to build a very deep and trainable network with a concat in concat (CIC) structure, which is similar to the residual in residual (RIR) structure in RCAN [48] but allows features are integrated in a more effective way.

2 Related Work

2.1 Image Super-Resolution with Deep Learning

The pioneering work that utilizes deep learning techniques to solve single image super-resolution tasks in the modern sense is SRCNN [7], which is a three-layer network that maps LR images to HR images in an end-to-end manner. Through introducing global residual learning (GRL), Kim *et al.* [21] increased the network depth up to 20 layers and achieved significant performance improvement. Tai *et al.* [39] presented a very deep memory network (MemNet) to solve the problem of long-term dependency. Instead, some other works, e.g, DRCN [22] and DRRN [38], focused on weight sharing to reduce the scale of model parameters. Although these methods achieve superior performance, they use the bicubic-interpolated version of original LR images as input, which inevitably loses some details and increases computational burden greatly [37,48].

This problem can be alleviated by placing nonlinear inference in LR image space and upscaling image resolution at the end of the network, such as transpose convolution [8] and ESPCNN [37]. Benefitting from this, some SR models can improve performance by significantly increasing their network scale, e.g., EDSR [31], RDN [49], D-DBPN [11] and RCAN [48] etc. However, the performance gain of these methods depends largely on the increase of model scale, e.g., EDSR [31] has about 43M model parameters and 70-layer network depth, and RCAN [48] also has more than 16M parameters and 400-layer network depth.

To generate more realistic results, especially for large scaling factors, Ledig *et al.* [28] proposed to introduce generative adversarial network (GAN) [9] into

image SR framework (SRGAN). They developed a new network structure based on ResNet [12,13] (SRResNet) and treated it as the generator of a GAN [9] with a perceptual loss [20]. The idea was also introduced into EnhanceNet [35] that combined automated texture synthesis and perceptual losses. However, although these GAN-based models can ease over-smoothing artifacts and present visually pleasuring results, their results may not be faithfully recovered with wrong image content and unpleasing artifacts [24,48].

2.2 Interlayer Bypass Connections

A simple and direct way to improve the performance of deep models is to increase model scale, e.g., network parameters, depth and width. However, more problems will arise as model scale increases, and more training tricks are needed to ensure effective training [29]. To alleviate the training difficulty caused by the increased model scale, interlayer connections are widely used in network design. Residual connection [12,13] and channel concatenation [17] are two typical interlayer skip connections. Although residual connection is a commonly used option in image SR, e.g., [21], [38], [39], [31] and [48], there is a large amount of representational redundancy in residual networks [18,40], which hints that residual learning may hinder the full mining of model capacities. In fact, when model scale is relatively fixed, the performance of residual models still has potential to be improved [2,42].

Channel concatenation is another way to implement skip connections in the context of image SR, such as MemNet [39], SRDenseNet [40], RDN [49], MSRN [29], as well as AWSRN [42] etc. However, these models usually combine channel concatenation with residual connections, expecting to make full use of features and mitigate the training difficulty. Instead, in the proposed FC^{2}N, all interlayer connections are implemented by our WCC. In this way, the model can adaptively build a reasonable amount of interlayer connections with appropriate strength and thereby make full exploitation of the representational capacity.

3 Fully Channel-Concatenated Network

3.1 Weighted Channel Concatenation

Most current deep models are modularized architectures that consist of many stacked building blocks, e.g., ResNet [12], MemNet [39], DRRN [38], SRResNet [28], EDSR [31], AWSRN [42], RCAN [48] etc. The structure of some typical building blocks is outlined in Fig.1. In the context of image SR, Conv-ReLU-Conv based residual block [31] and its variants are broadly adopted as the building modules of deep SR models, such as Fig.1(d) and Fig.1(e). Most these building blocks, however, are combined with the strategy of residual learning for efficient model training.

The building block of our FC^{2}N model is also based on the Conv-ReLU-Conv structure, but it avoids using residual learning. Instead, we adopt WCC followed by a 1×1 conv layer to integrate the input and output of the Conv-ReLU-Conv structure, as shown in Fig.1(f). Temporarily, let $\mathbf{x}_t \in \mathbb{R}^{H\times W\times C}$ be the input of a CB and $\mathcal{H}(\cdot)$ the function corresponding to the nonlinear mapping branch, the

WCC can be formulated as:

$$\mathbf{x}_{t+1} = \mathcal{L}([\lambda_1 \mathbf{x}_t, \lambda_2 \mathcal{H}(\mathbf{x}_t)]), \tag{1}$$

where $\mathbf{x}_{t+1}$ is the output of the CB, and $[\ldots]$ represents the operation of channel concatenation. $\mathcal{L}(\cdot)$ denotes the 1×1 conv, and λ_1 and λ_2 are the corresponding weighting factors, as shown in Fig.1(f).

Let's rewrite $\mathbf{x}_t = [x_t^1, \ldots, x_t^i, \ldots, x_t^C]$ and $\mathcal{H}(\mathbf{x}_t) = [x_h^1, \ldots, x_h^j, \ldots, x_h^C]$, both of which have C feature maps with spatial size of $H \times W$. Denote the kernel of 1×1 conv as $\mathbf{K} \in \mathbb{R}^{2C \times C}$ with $2C$ input channels and C output channels[1], and omit the biases, then $\mathbf{x}_{t+1}$ is obtained by:

$$x_{t+1}^u = \sum_{i=1}^{C} \lambda_1 x_t^i \mathbf{K}(i, u) + \sum_{j=1}^{C} \lambda_2 x_h^j \mathbf{K}(C + j, u), \tag{2}$$

where x_{t+1}^u is the u-th feature map of $\mathbf{x}_{t+1}$, i.e., $\mathbf{x}_{t+1} = [x_{t+1}^1, \ldots, x_{t+1}^u, \ldots, x_{t+1}^C]$. Here u is the index of output channel. In fact, both Fig.1(c) and Fig.1(e) are special cases of our CB blocks, i.e., Fig.1(f). When

$$\mathbf{K}(i, i) = \mathbf{K}(C + i, i) = 1, \quad i = 1, \ldots, C, \tag{3}$$

all other elements in $\mathbf{K}$ are 0, and $\lambda_1 = \lambda_2 = 1$, then a CB block degrades to the residual block of EDSR [31]. If λ_1 and λ_2 act as learnable weighting factors at this time, then it degrades to the residual block in AWSRN [42].

In addition, our WCC can also achieve attention mechanism [15,48] that can be viewed as a guidance to bias the allocation of available processing resources towards the most informative components of an input [15]. But unlike previous self-attention, 1×1 conv layer contributes to the *joint* attention to the identity mapping and the output features of the nonlinear mapping branch.

3.2 Overall Network Structure

The overall structure of the FC^{2}N network is shown in Fig.2. Similar to many previous models, it mainly includes 3 stages: shallow feature extraction, nonlinear mapping and image reconstruction, which are denoted as $\mathcal{F}_E(\cdot)$, $\mathcal{F}_N(\cdot)$ and $\mathcal{F}_R(\cdot)$ respectively. Assume that the model takes $\mathbf{x}$ as input and outputs $\mathbf{y}$. The shallow features are first extracted by a single 3×3 conv layer:

$$\mathbf{x}_0 = \mathcal{F}_E(\mathbf{x}), \tag{4}$$

where $\mathbf{x}_0$ is the extracted shallow feature maps, and $\mathcal{F}_E(\cdot)$ denotes the 3×3 conv layer. Subsequently, $\mathbf{x}_0$ is fed into the nonlinear mapping subnet for nonlinear inference. This generates deep features and can be formulated as:

$$\mathbf{x}_{n+1} = \mathcal{F}_N(\mathbf{x}_0), \tag{5}$$

[1] In implementation, the shape of $\mathbf{K}$ is $[k, k, C_{\text{in}}, C_{\text{out}}]$, where k is kernel size, C_{in} and C_{out} are the number of input and output channels. Since the spatial size of $\mathbf{K}$ is 1×1 here, we remove the singular dimensions for simplification.

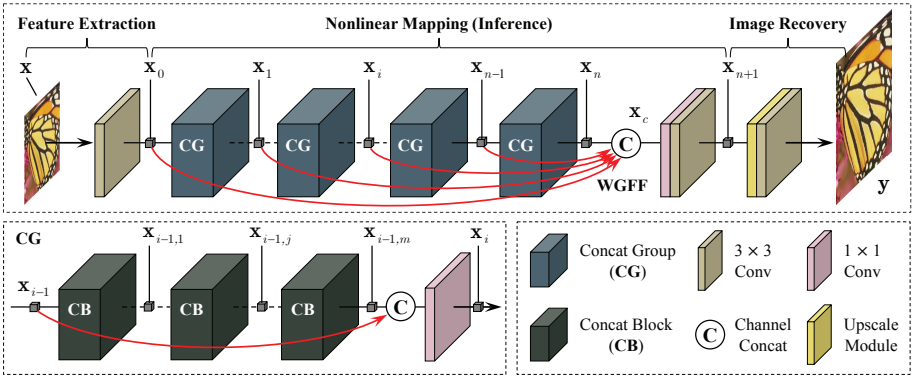


Fig. 2. Overall structure of our FC^{2}N model. The red arrows in WGFF and CG denote WCC. Note that there are no residual connections in the entire network.

where $\mathbf{x}_{n+1}$ is the generated features, and $\mathcal{F}_N(\cdot)$ represents the entire nonlinear mapping, which consists of n cascaded CGs combined with the WGFF. Let's denote the function of the i-th CG as $\mathcal{G}_i(\cdot)$, i.e., $\mathbf{x}_i = \mathcal{G}_i(\mathbf{x}_{i-1})$. Then we can obtain the nonlinear mapping from $\mathbf{x}_0$ to $\mathbf{x}_n$ iteratively:

$$\mathbf{x}_n = \mathcal{G}_n(\mathbf{x}_{n-1}) = \mathcal{G}_n\Big[\mathcal{G}_{n-1}\Big(\cdots\mathcal{G}_1(\mathbf{x}_0)\cdots\Big)\Big]. \tag{6}$$

It is worth noting that simply stacking multiple CGs can easily lead to training failure. Therefore, we use WGFF to integrate intermediate features at different depths, which also helps ease the training difficulty. The deep features $\mathbf{x}_{n+1}$ is therefore obtained by:

$$\mathbf{x}_{n+1} = \mathcal{F}_D(\mathbf{x}_c) = \mathcal{F}_D([\lambda_0\mathbf{x}_0, \lambda_1\mathbf{x}_1, \ldots, \lambda_n\mathbf{x}_n]), \tag{7}$$

where λ_i is the weighting factor for the corresponding intermediate feature. $\mathcal{F}_D(\cdot)$ corresponds to the 1×1 conv followed by a 3×3 conv, and $[\ldots]$ represents the channel concatenation. This WGFF can further explore model representational capacity by adding negligible model parameters. As in [31,49,51], we adopt the ESPCNN [37] to upsample deep features and a 3×3 conv layer to recover HR images: $\hat{\mathbf{y}} = \mathcal{F}_R(\mathbf{x}_{n+1})$, where $\hat{\mathbf{y}}$ is the final SR output and $\mathcal{F}_R(\cdot)$ corresponds to the upscale module followed by a 3×3 conv layer.

We choose to optimize L_1 loss as previous works [31,48,51]. Given a training set $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$, where N is the total number of training samples, the L_1 loss is given by:

$$L_1(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{F}\big(\mathbf{x}^{(i)};\boldsymbol{\theta}\big) - \mathbf{y}^{(i)}\right\|_1, \tag{8}$$

where $\mathcal{F}(\cdot)$ corresponds to the entire network, and $\boldsymbol{\theta}$ denotes the parameter set of the model. This is optimized by the Adam [23] method. More details on model training will be shown in section 4.1.

Table 1. Quantitative results of ablation study on the proposed WCC. All models are evaluated on Set5 [4] with SR×4, and valid curves are shown in Fig.3 ($m = 8, n = 16$)

Configs	CB	/	0	1	0	1	0	1	0	1
	CG	/	0	0	1	1	0	0	1	1
	WGFF	/	0	0	0	0	1	1	1	1
SR×4	PSNR	32.53	32.62	32.63	32.59	32.64	32.62	32.62	32.65	32.67

3.3 Concat Group (Concat In Concat)

A CG is simply composed of m stacked CBs with an additional WCC, as shown in Fig.2. Let $\mathbf{x}_{i-1} = \mathbf{x}_{i-1,0}$ be the input of the first CB in the i-th CG, then we can obtain the local features as following:

$$\mathbf{x}_{i-1,j} = \mathcal{B}_{i,j}(\mathbf{x}_{i-1,j-1}), \quad j = 1, 2, \ldots, m, \tag{9}$$

where $\mathcal{B}_{i,j}(\cdot)$ represents the function corresponding to the j-th CB in the i-th CG. Similar to the entire nonlinear mapping, we have:

$$\mathbf{x}_{i-1,m} = \mathcal{B}_{i,m}\Big[\mathcal{B}_{i,m-1}\Big(\cdots \mathcal{B}_{i,1}(\mathbf{x}_{i-1,0})\cdots\Big)\Big]. \tag{10}$$

To promote the information flow of the network, we further adopt WCC to merge the input of the first CB and the output of the last CB, as shown in Fig.2. This CIC structure can not only ease local representation learning, but also allow the linear and nonlinear features are fused in a fine-grained manner. Therefore, we generate the final output of the i-th CG:

$$\mathbf{x}_i = \mathcal{L}_i([\lambda_{i-1,0}\mathbf{x}_{i-1,0},\ \lambda_{i-1,m}\mathbf{x}_{i-1,m}]), \tag{11}$$

where $\mathcal{L}_i(\cdot)$ stands for the 1×1 conv in the i-th CG, and $[\ldots]$ denotes channel concatenation. $\lambda_{i-1,0}$ and $\lambda_{i-1,m}$ are two weighting factors. These operations are similar to local feature fusion (LFF) in some previous work, such as [49] and [48]. However, we use WCC followed by a 1×1 conv layer instead of a 3×3 conv layer followed by a residual connection [48] to fuse local features.

3.4 Implementation Details

To verify the effective mining of model representational capacity, we implement both largescale and lightweight FC^{2}N models by setting $m = 8, n = 16$ and $m = 4, n = 4$ respectively. In addition, we also use *wide activation* [45] strategy to allow more low-level information pass through the network while still keep its highly nonlinearity. Specifically, the number of feature channels in the CB is set to $\{32, 128, 32\}$. By shrinking the input and output features and extending the features before ReLU layers, wide activation favors to activating more low-level information without increasing model parameters [45,42].

The last layer has 3 filters as it outputs RGB images. Elsewhere, the number of feature channels is set to 32. Except for the 1×1 conv layers annotated in Fig.1 and Fig.2, all other layers have 3×3 kernels, and zero-padding is applied to keep the spatial size of features unchanged. For interlayer connections in WGFF, CG and CB, all weighting factors are learnable and initialized as 1.0.

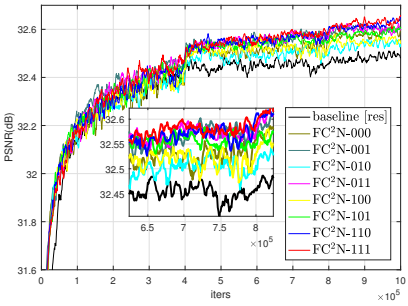


Fig. 3. Valid PSNR curves on the WCC

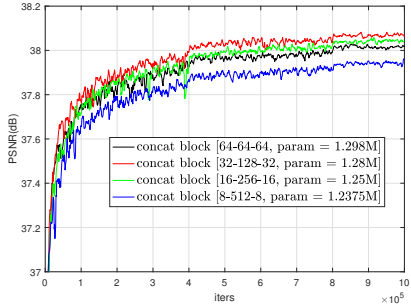


Fig. 4. Valid curves on wide activation

Table 2. Impact of n and m on model performance (PSNR / params)

$n \backslash m$	2	4	6	8
2	31.75dB / 0.40M	32.03dB / 0.70M	32.12dB / 1.01M	32.31dB / 1.31M
4	32.02dB / 0.71M	32.23dB / 1.31M	32.33dB / 1.92M	32.40dB / 2.53M
8	32.23dB / 1.33M	32.36dB / 2.54M	32.52dB / 3.76M	32.56dB / 4.97M
16	32.40dB / 2.57M	32.53dB / 5.00M	32.62dB / 7.43M	32.67dB / 9.86M

4 Experimental Results

4.1 Settings

As in [31,49,48,11], we adopt 800 training images from DIV2K dataset [1] as our training set. Data augmentation is performed on training images by randomly horizontal and vertical flips, 90° rotations and data range complementarity. Five benchmark datasets, including Set5 [4], Set14 [46], B100 [33], Urban100 [19] and Manga109 [34], are used for evaluation. The results are typically evaluated with PSNR and SSIM [44] on Y channel of YCbCr space. For model training, 48×48 LR image patches are extracted from LR images, while the size of HR patches corresponds to the scaling factors. Batch size is set to 16 as in previous works [31,49,48]. The objective function is minimized by the Adam optimizer [23] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is initialized as 2×10^{-4} for all layers and halved for every 4.0×10^5 training steps. Both largescale and lightweight FC^{2}N models are trained for 10^6 iterations in total. To compare the computational overhead of deep models in the lightweight case, we also introduce MultiAdds [2] as an evaluation metric[2].

4.2 Model Analysis

Weighted Channel Concatenation To show the superiority of our WCC to residual learning, we build a baseline residual model, in which all skip connections in CGs and CBs are replaced by *unweighted* residual connections and WGFF is changed to GFF [49]. This model corresponds to the "baseline [res]" in Fig.3 and the first column in Table 1 (PSNR = 32.53 dB). To investigate the impact of learnable weighting factors, we also conduct ablation study on whether channel concatenations in CB, CG and WGFF are weighted. They are indicated by "1" if

[2] The source code of our FC^{2}N is available at https://github.com/zxlation/FC2N.

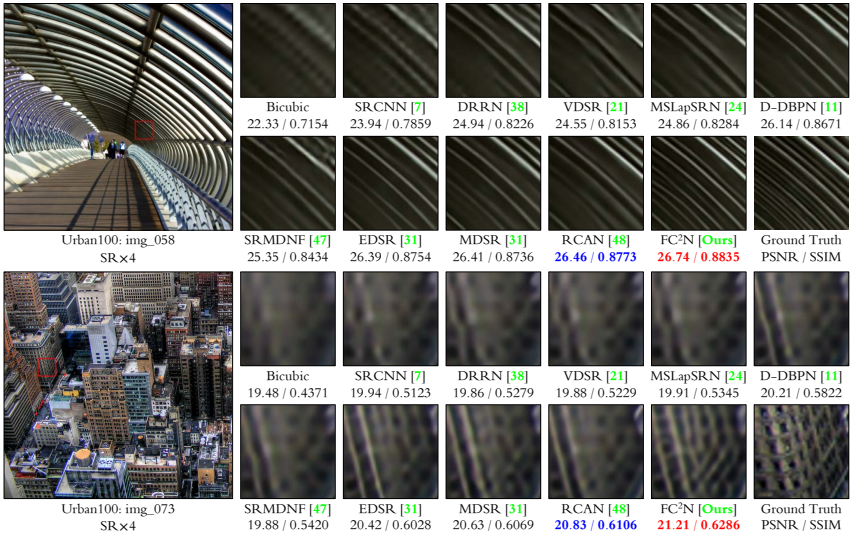


Fig. 5. Visual comparison between other SR methods and our largescale FC^{2}N. The best and second best results are marked in red and blue respectively.

they are weighted by learnable parameters, otherwise denoted as "0". Notations and quantitative results are shown in Fig.3 and Table 1. Note that three numbers representing model suffix correspond to WGFF, CG and CB respectively. For example, FC^{2}N-110 represents the model with WGFF and CG are weighted but CB is unweighted, which corresponds to the penultimate column of Table 1. In addition, we find that eliminating WCC in either CB, CG or WGFF is likely to result in training failure, so we do not perform ablation study in this case.

Comparison with Residual Baseline: As illustrated in Fig.3 and Table 1, by comparing with FC^{2}N-000, it can be seen that about 0.1dB performance gain can be achieved by changing the residual connections in CGs and CBs to *unweighted* channel concatenations. The validation curves of baseline [res] and FC^{2}N-000 in Fig.3 also illustrate that, after the decays of learning rate, channel concatenation can make better SR performance improvement than residual connection although both models have roughly the same model parameters (9.6M vs. 9.8M).

Learnable Weighting Factors: In Table 1, by comparing FC^{2}N-001, FC^{2}N-010 and FC^{2}N-100 with FC^{2}N-000, we can see that learnable weighting factors in all WGFF, CGs and CBs favor to performance improvement. In addition, the best performance given by FC^{2}N-111 shows that the WCC in WGFF, CGs and CBs is beneficial to maximizing the performance gain. In particular, WGFF regularly integrates shallow and deep features in an adaptive manner, hence improving the information flow throughout the entire network. This can also be verified by the comparison between FC^{2}N-011 and FC^{2}N-111. In fact, the proposed WCC also helps stabilize model training. Although Table 1 presents the results of all configurations, the models that do not make adequate use of weighting, such as FC^{2}N-000, FC^{2}N-001 and FC^{2}N-010 etc., show some training difficulty as they

Table 3. Quantitative comparison w.r.t **large-scale** FC^{2}N ($n = 16, m = 8$). Best and second best results are marked in red and blue respectively (PSNR (dB)/SSIM).

Methods	r	Params	Set5 [4]	Set14 [46]	B100 [33]	Urban100 [19]	Manga109 [34]
Bicubic	2	/	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	26.88 / 0.8403	30.80 / 0.9339
EDSR [31]	2	40.7M	38.11 / 0.9602	33.92 / 0.9195	32.32 / 0.9013	32.93 / 0.9351	39.10 / 0.9773
D-DBPN [11]	2	5.82M	38.09 / 0.9600	33.85 / 0.9190	32.27 / 0.9000	32.55 / 0.9324	38.89 / 0.9775
RDN [49]	2	22.1M	38.24 / 0.9614	34.01 / 0.9212	32.34 / 0.9017	32.89 / 0.9353	39.18 / 0.9780
RCAN [48]	2	15.4M	38.27 / 0.9614	34.11 / 0.9216	32.41 / 0.9026	33.34 / 0.9384	39.44 / 0.9786
SAN [6]	2	15.7M	38.31 / 0.9620	34.07 / 0.9213	32.42 / 0.9028	33.10 / 0.9370	39.32 / 0.9790
SRFBN [30]	2	N/A	38.11 / 0.9609	33.82 / 0.9196	32.29 / 0.9010	33.62 / 0.9328	39.08 / 0.9779
OISR [14]	2	N/A	38.21 / 0.9612	33.94 / 0.9206	32.36 / 0.9019	33.03 / 0.9365	N/A
FC^{2}N [Ours]	2	9.82M	38.29 / 0.9616	34.14 / 0.9224	32.40 / 0.9025	33.18 / 0.9379	39.51 / 0.9787
FC^{2}N$^+$ [Ours]	2	9.82M	38.33 / 0.9617	34.24 / 0.9224	32.44 / 0.9029	33.34 / 0.9388	39.62 / 0.9790
FC^{2}N^{++} [Ours]	2	9.82M	38.34 / 0.9618	34.25 / 0.9225	32.45 / 0.9030	33.39 / 0.9392	39.68 / 0.9792
Bicubic	3	/	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	24.46 / 0.7349	26.95 / 0.8556
EDSR [31]	3	43.7M	34.65 / 0.9280	30.52 / 0.8462	29.25 / 0.8093	28.80 / 0.8653	34.17 / 0.9476
RDN [49]	3	22.3M	34.71 / 0.9296	30.57 / 0.8468	29.26 / 0.8093	28.80 / 0.8653	34.13 / 0.9484
RCAN [48]	3	15.6M	34.74 / 0.9299	30.65 / 0.8482	29.32 / 0.8111	29.09 / 0.8702	34.44 / 0.9499
SAN [6]	3	15.9M	34.75 / 0.9300	30.59 / 0.8476	29.33 / 0.8112	28.93 / 0.8671	34.30 / 0.9494
SRFBN [30]	3	N/A	34.70 / 0.9292	30.51 / 0.8461	29.24 / 0.8084	28.73 / 0.8641	34.18 / 0.9481
OISR [14]	3	N/A	34.72 / 0.9297	30.57 / 0.8470	29.29 / 0.8103	28.95 / 0.8680	N/A
FC^{2}N [Ours]	3	9.87M	34.76 / 0.9302	30.66 / 0.8485	29.31 / 0.8106	29.04 / 0.8700	34.63 / 0.9504
FC^{2}N$^+$ [Ours]	3	9.87M	34.85 / 0.9307	30.76 / 0.8495	29.36 / 0.8114	29.22 / 0.8725	34.87 / 0.9514
FC^{2}N^{++} [Ours]	3	9.87M	34.85 / 0.9307	30.78 / 0.8497	29.37 / 0.8115	29.24 / 0.8727	34.91 / 0.9515
Bicubic	4	/	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577	24.89 / 0.7866
EDSR [31]	4	43.1M	32.46 / 0.8968	28.80 / 0.7876	27.71 / 0.7420	26.64 / 0.8033	31.02 / 0.9148
D-DBPN [11]	4	10.4M	32.47 / 0.8980	28.82 / 0.7860	27.72 / 0.7400	26.38 / 0.7946	30.91 / 0.9137
RDN [49]	4	22.3M	32.47 / 0.8990	28.81 / 0.7871	27.72 / 0.7419	26.61 / 0.8028	31.00 / 0.9151
RCAN [48]	4	15.6M	32.63 / 0.9002	28.87 / 0.7889	27.77 / 0.7436	26.82 / 0.8087	31.22 / 0.9173
SAN [6]	4	15.9M	32.64 / 0.9003	28.92 / 0.7888	27.78 / 0.7436	26.79 / 0.8068	31.18 / 0.9169
SRFBN [30]	4	N/A	32.47 / 0.8983	28.81 / 0.7868	27.72 / 0.7409	26.60 / 0.8015	31.15 / 0.9160
OISR [14]	4	N/A	32.53 / 0.8992	28.86 / 0.7878	27.75 / 0.7428	26.64 / 0.8033	N/A
FC^{2}N [Ours]	4	9.86M	32.67 / 0.9005	28.90 / 0.7889	27.77 / 0.7432	26.85 / 0.8089	31.41 / 0.9193
FC^{2}N$^+$ [Ours]	4	9.86M	32.73 / 0.9011	29.00 / 0.7906	27.83 / 0.7445	27.01 / 0.8127	31.71 / 0.9215
FC^{2}N^{++} [Ours]	4	9.86M	32.74 / 0.9012	29.02 / 0.7909	27.83 / 0.7446	27.03 / 0.8130	31.74 / 0.9216
Bicubic	8	/	24.40 / 0.6580	23.10 / 0.5660	23.67 / 0.5480	20.74 / 0.5160	21.47 / 0.6500
EDSR [31]	8	45.5M	26.96 / 0.7762	24.91 / 0.6420	24.81 / 0.5985	22.51 / 0.6221	24.69 / 0.7841
D-DBPN [11]	8	23.2M	27.21 / 0.7840	25.13 / 0.6480	24.88 / 0.6010	22.73 / 0.6312	25.14 / 0.7987
RCAN [48]	8	15.7M	27.31 / 0.7878	25.23 / 0.6511	24.98 / 0.6058	23.00 / 0.6452	25.24 / 0.8029
SAN [6]	8	16.1M	27.22 / 0.7829	25.14 / 0.6476	24.88 / 0.6011	22.70 / 0.6314	24.85 / 0.7906
FC^{2}N [Ours]	8	9.90M	27.25 / 0.7833	25.10 / 0.6479	24.87 / 0.6016	22.72 / 0.6331	25.00 / 0.7937
FC^{2}N$^+$ [Ours]	8	9.90M	27.35 / 0.7880	25.27 / 0.6512	24.96 / 0.6041	22.94 / 0.6398	25.34 / 0.8005
FC^{2}N^{++} [Ours]	8	9.90M	27.35 / 0.7872	25.29 / 0.6517	25.01 / 0.6043	22.97 / 0.6407	25.38 / 0.8015

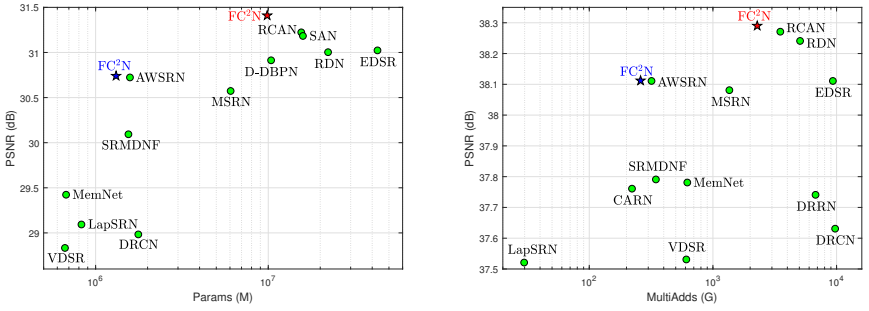
crashed many times during training. The convergence curves shown in Fig.3 also verify the above analysis.

Wide Activation In the context of single image SR, low-level information may contribute to more accurate pixel-wise predication [45,42], and wide activation is considered to help promote the propagation of low-level features in the network and boost model performance. To verify the effectiveness of wide activation, we compare several configurations in this section. Fig.4 shows the validation curves of the lightweight FC^{2}N ($n = 4, m = 4$) on Set5 [4] with SR$\times$2, which is equipped with different configurations of wide activation.

Assume that the width of identity mapping pathway is w_1 and that of the nonlinear mapping pathway is w_2. Let r_{wa} denote the ratio of w_1 and w_2:

$$r_{wa} = \frac{w_2}{w_1}. \tag{12}$$

For fairness of comparison, we keep the feature width of feature extraction and image reconstruction the same as that of FC^{2}N. It can be seen from Fig.4 that



(a) Manga109 [34] with SR×4 (b) Set5 [4] with SR×2

Fig. 6. Efficiency analysis on parameters and computational overhead. Red represents largescale implementation, blue denotes lightweight version.

properly increasing r_{wa} favors to performance improvement, e.g., $r_{wa} = 1$ and $r_{wa} = 4$. However, model performance will degrade as r_{wa} continues to increase and reaches a certain threshold, e.g., $r_{wa} = 64$. Similar phenomenon was also observed in [45] and one possible reason for this performance degradation is that the identity mapping becomes too slim, resulting in the bottleneck of low-level information propagation. The observation suggests that the wide activation that is effective in residuals learning is also effective in WCC, which experimentally demonstrates the inclusion of residual connection in WCC.

The Number of CG and CB Table 2 exhibits the testing results of different combinations of m and n, on Set5 [4] with SR×4. It can be seen that the increase in both m and n helps boost the performance of the model, which is unsurprising because increasing m and n obviously enlarges model scale, including network depth and model parameters. It is worth noting that at roughly the same model scale, larger m is more helpful to performance improvement than larger n. For instance, the model with $n = 8, m = 2$ performs slightly worse than the model with $n = 2, m = 8$, which has fewer parameters but the same network depth and width. Moreover, we can see that our FC^{2}N achieves excellent SR performance in both lightweight and largescale implementations. This implies that it consistently provides good performance-scale tradeoffs as model scale changes.

4.3 Comparison with Advanced Methods

In this section, we compare both largescale and lightweight FC^{2}N models with other advanced methods. Similar to [31], [49] and [48], we also use the geometric self-ensemble to improve performance, which is denoted as FC^{2}N$^+$. Besides, we introduce another strategy to further boost model performance, i.e., data range ensemble, which is computed as following: for each testing image $\mathbf{x}$, generate an image $\bar{\mathbf{x}}$ with complementary data range by $\bar{\mathbf{x}} = 255 - \mathbf{x}$, and then feed it into the model to produce HR output $\bar{\mathbf{y}}$. The final HR prediction of $\mathbf{x}$ is given by $[\mathbf{y} + (255 - \bar{\mathbf{y}})]/2$. When both strategies are applied, it is denoted as FC^{2}N^{++}.

Largescale Implementation Table 3 illustrates the quantitative comparison between the proposed FC^{2}N and other largescale SR models in case of largescale

Table 4. Quantitative comparison with **lightweight** FC^{2}N ($n = m = 4$). The metric of MultAdds is computed with HR images with size of 720p (1280×720).

Methods	r	Params	MultiAdds	Set5 [4]	Set14 [46]	B100 [33]	Manga109 [34]
Bicubic	2	/	/	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	30.80 / 0.9339
SRCNN [7]	2	57K	52.7G	36.66 / 0.9542	32.42 / 0.9063	31.36 / 0.8879	35.74 / 0.9661
VDSR [21]	2	665K	612.6G	37.53 / 0.9587	33.03 / 0.9124	31.90 / 0.8960	37.22 / 0.9729
DRRN [38]	2	297K	6,796.9G	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	37.92 / 0.9760
LapSRN [24]	2	813K	29.9G	37.52 / 0.9590	33.08 / 0.9130	31.80 / 0.8950	37.27 / 0.9740
SRMDNF [47]	2	1,513K	347.7G	37.79 / 0.9600	33.32 / 0.9150	32.05 / 0.8980	38.07 / 0.9761
NLRN [32]	2	N/A	N/A	38.00 / 0.9603	33.46 / 0.9159	32.19 / 0.8992	N/A
AWSRN [42]	2	1,397K	320.5G	38.11 / 0.9608	33.78 / 0.9189	32.26 / 0.9006	38.87 / 0.9776
CARN [2]	2	1,592K	222.8G	37.76 / 0.9590	33.52 / 0.9166	32.09 / 0.8978	N/A
MSRN [29]	2	5,930K	1,365.4G	38.08 / 0.9607	33.70 / 0.9186	32.23 / 0.9002	38.69 / 0.9772
FC^{2}N [Ours]	2	1,277K	294.0G	38.11 / 0.9608	33.70 / 0.9179	32.21 / 0.9001	38.81 / 0.9775
FC^{2}N$^+$ [Ours]	2	1,277K	294.0G	38.16 / 0.9610	33.78 / 0.9189	32.25 / 0.9006	39.02 / 0.9779
FC^{2}N^{++} [Ours]	2	1,277K	294.0G	38.17 / 0.9611	33.79 / 0.9191	32.28 / 0.9007	39.04 / 0.9780
Bicubic	3	/	/	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	26.95 / 0.8556
SRCNN [7]	3	57K	52.7G	32.75 / 0.9090	29.28 / 0.8209	28.41 / 0.7863	30.59 / 0.9107
VDSR [21]	3	665K	612.6G	33.66 / 0.9213	29.77 / 0.8314	28.82 / 0.7976	32.01 / 0.9310
DRRN [38]	3	297K	6,796.9G	34.03 / 0.9244	29.96 / 0.8349	28.95 / 0.8004	32.74 / 0.9390
SRMDNF [47]	3	1,530K	156.3G	34.12 / 0.9250	30.04 / 0.8370	28.97 / 0.8030	33.00 / 0.9403
NLRN [32]	3	N/A	N/A	34.27 / 0.9266	30.16 / 0.8374	29.06 / 0.8026	N/A
AWSRN [42]	3	1,476K	150.6G	34.52 / 0.9281	30.38 / 0.8426	29.16 / 0.8069	33.85 / 0.9463
CARN [2]	3	1,592K	118.8G	34.29 / 0.9255	30.29 / 0.8407	29.06 / 0.8034	N/A
MSRN [29]	3	6,114K	625.7G	34.46 / 0.9278	30.41 / 0.8437	29.15 / 0.8064	33.67 / 0.9456
FC^{2}N [Ours]	3	1,323K	135.8G	34.53 / 0.9282	30.44 / 0.8437	29.16 / 0.8068	33.83 / 0.9462
FC^{2}N$^+$ [Ours]	3	1,323K	135.8G	34.60 / 0.9287	30.51 / 0.8449	29.20 / 0.8074	34.11 / 0.9476
FC^{2}N^{++} [Ours]	3	1,323K	135.8G	34.61 / 0.9287	30.52 / 0.8450	29.21 / 0.8075	34.16 / 0.9478
Bicubic	4	/	/	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	24.89 / 0.7866
SRCNN [7]	4	57K	52.7G	30.48 / 0.8628	27.49 / 0.7503	26.90 / 0.7101	27.66 / 0.8505
VDSR [21]	4	665K	612.6G	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	28.83 / 0.8809
DRRN [38]	4	297K	6,796.9G	31.68 / 0.8888	28.21 / 0.7720	27.38 / 0.7284	29.46 / 0.8960
LapSRN [24]	4	813K	149.4G	31.54 / 0.8850	28.19 / 0.7720	27.32 / 0.7280	29.09 / 0.8845
SRMDNF [47]	4	1,555K	89.3G	31.96 / 0.8930	28.35 / 0.7770	27.49 / 0.7340	30.09 / 0.9024
NLRN [32]	4	N/A	N/A	31.92 / 0.8916	28.36 / 0.7745	27.48 / 0.7346	N/A
AWSRN [42]	4	1,587K	91.1G	32.27 / 0.8960	28.69 / 0.7843	27.64 / 0.7385	30.72 / 0.9109
CARN [2]	4	1,592K	90.9G	32.13 / 0.8937	28.60 / 0.7806	27.58 / 0.7349	N/A
MSRN [29]	4	6,078K	349.8G	32.26 / 0.8960	28.63 / 0.7836	27.61 / 0.7380	30.57 / 0.9103
FC^{2}N [Ours]	4	1,314K	82.6G	32.23 / 0.8956	28.68 / 0.7836	27.62 / 0.7377	30.74 / 0.9110
FC^{2}N$^+$ [Ours]	4	1,314K	82.6G	32.36 / 0.8970	28.75 / 0.7851	27.68 / 0.7390	31.04 / 0.9136
FC^{2}N^{++} [Ours]	4	1,314K	82.6G	32.37 / 0.8971	28.76 / 0.7853	27.68 / 0.7391	31.06 / 0.9137

implementation ($n = 16, m = 8$). It can be observed that our FC^{2}N outperforms most of other methods on all datasets. In particular, FC^{2}N^{++} further improves the performance of FC^{2}N$^+$ on the whole, which verifies the effectiveness of data range ensemble. When the scaling factor is 8, our FC^{2}N performs slightly worse than RCAN [48] but still better than other SR models. However, our FC^{2}N only uses about 60% model parameters of RCAN [48], indicating that it provides a better trade-off between model performance and network scale.

Fig.5 shows the visual comparison between other SR methods and our FC^{2}N on two testing images from Urban100 [19] with SR×4. For "img_058", most of previous methods generate blurring artifacts at the fringes, especially for those in the lower left parts of the cropped images. However, only the FC^{2}N generate the result much closer to the ground truth. For image "img_073", the blurring effect in the results of other methods in texture region is more obvious, but our FC^{2}N can still produce the result that can imply the potential structure more clearly. These comparisons show the good representational capacity our FC^{2}N.

Lightweight Implementation To show that our FC^{2}N makes full exploitation of representational capacity, we also compare our lightweight FC^{2}N ($m = n = 4$)

Fig. 7. Visual comparison of largescale FC^{2}N with adversarial training ($n = 16, m = 8$)

with other lightweight SR models, as shown in Table 4. The size of HR images is assumed to be 1280×720 (i.e., 720p) for the calculation of MultiAdds, as in [2] and [42]. We can see that our FC^{2}N gives the best trade-off between performance and model scale, and computational burden. It provides comparable performance to AWSRN [42] with fewer parameters and computational overhead.

Efficiency Analysis To further illustrate the efficiency superiority of our FC^{2}N to other models, we plot model performance versus parameters and MultiAdds in Fig.6(a) and Fig.6(b) respectively. As can be seen, our FC^{2}N shows the best trade-off on parameter utilization and calculation efficiency for both largescale and lightweight implementations.

4.4 Adversarial Training

The proposed FC^{2}N can also be extended to adversarial training [9] for visually pleasuring SR results. We build a GAN model with our largescale FC^{2}N as the generator (G) and a 5-layer FCNN [36] as the discriminator (D). The first three conv layers in D have 3×3 kernels, and the output channels are set to 64, 128 and 256 respectively. Then, a global average pooling is used to squeeze the spatial dimensions, which is followed by two 1×1 layers that have 1024 and 1 output channels respectively. To train this GAN model, we adopt the same loss function for G as [10,11], which has the form of $\lambda_1 L_{mse} + \lambda_2 L_{vgg} + \lambda_3 L_{adv} + \lambda_4 L_{sty}$. We set $\lambda_1 = 1.0, \lambda_2 = 0.1, \lambda_3 = 2 \times 10^{-3}, \lambda_4 = 1.0$ in our experiment.

Fig.7 shows a visual result for SR×4. The model with adversarial training generates more plausible result on texture regions, even better than the ground truth (e.g., hair in the Lena image). However, the recovered results may not be faithfully produced with unexpected artifacts.

5 Conclusions

In this paper, we present a novel and simple network structure aimed at effective image SR tasks, i.e., FC^{2}N. Compared with previous advanced models, a major technical novelty of our FC^{2}N is the introduction of WCC as all skip connections in the network, and the avoidance to use residual learning. Through WCC, the model can not only adaptively select effective interlayer skips and make full use of hierarchical features, but also pay joint attention to the linear and nonlinear features. The CIC structure with CGs and CBs can also ease local representation learning and allow the model to fuse features from a fine to coarse level. Extensive experiments show that our FC^{2}N model outperforms most advanced models in both lightweight and largescale implementations, verifying its effective mining of model representational capacity.

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPR. pp. 1122–1131 (2017) 8
2. Ahn, N., Kang, B., *et al.*: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV. pp. 256–272 (2018) 2, 4, 8, 12, 13
3. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. IEEE Trans. Pattern Analy. Mach. Intell. **24**(9), 1167–1183 (2002) 1
4. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC. pp. 1–10 (2012) 7, 8, 10, 11, 12
5. Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J.: Dual path networks. In: NIPS. pp. 4470–4478 (2017) 2
6. Dai, T., Cai, J., Zhang, Y., Xia, S., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR. pp. 11065–11074 (2019) 10
7. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016) 1, 3, 12
8. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV 2016. pp. 391–407 (2016) 3
9. Goodfellow, I.J., Pouget-Abadie, J., *et al.*: Generative adversarial nets. In: NIPS. pp. 2672–2680 (2014) 3, 4, 13
10. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for single image super-resolution. arXiv preprint **arxiv:1904.05677** (2019) 13
11. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR 2018. pp. 1664–1673 (2018) 3, 8, 10, 13
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1, 2, 4
13. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. pp. 630–645 (2016) 2, 4
14. He, X., Mo, Z., Wang, P., Liu, Y., Yang, M., Cheng, J.: Ode-inspired network design for single image super-resolution. In: CVPR. pp. 1732–1741 (2019) 10
15. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. arXiv preprint **arxiv:1709.01507** (2017) 5

16. Hu, Y., Li, J., Huang, Y., Gao, X.: Channel-wise and spatial feature modulation network for single image super-resolution. arXiv preprint **arXiv:1809.11130v1** (2018) 2
17. Huang, G., Liu, Z., *et al.*: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017) 2, 4
18. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: ECCV. pp. 646–661 (2016) 2, 4
19. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. pp. 5197–5206 (2015) 8, 10, 12
20. Johnson, J., Alahi, A., Li, F.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. pp. 694–711 (2016) 4
21. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR 2016. pp. 1646–1654 (2016) 2, 3, 4, 12
22. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR 2016. pp. 1637–1645 (2016) 2, 3
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 6, 8
24. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR 2017. pp. 5835–5843 (2017) 4, 12
25. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015) 1
26. LeCun, Y., Boser, B.E., *et al.*: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989) 1
27. LeCun, Y., Boser, B.E., *et al.*: Handwritten digit recognition with a back-propagation network. In: NIPS. pp. 396–404 (1989) 1
28. Ledig, C., Theis, L., Huszar, F., *et al.*: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR 2017. pp. 105–114 (2017) 3, 4
29. Li, J., Fang, F., Mei, K., Zhang, G.: Multi-scale residual network for image super-resolution. In: ECCV. pp. 527–542 (2018) 2, 4, 12
30. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: CVPR. pp. 3867–3876 (2019) 10
31. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops 2017. pp. 1132–1140 (2017) 2, 3, 4, 5, 6, 8, 10, 11
32. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: NeurIPS. pp. 1680–1689 (2018) 12
33. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. pp. 416–425 (2001) 8, 10, 12
34. Matsui, Y., Ito, K., *et al.*: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications **76**(20), 21811–21838 (2017) 8, 10, 11, 12
35. Sajjadi, M.S.M., Schölkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. pp. 4501–4510 (2017) 4
36. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. TPAMI **39**(4), 640–651 (2017) 13
37. Shi, W., Caballero, J., Huszar, F., *at al.*: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR 2016. pp. 1874–1883 (2016) 3, 6
38. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR 2017. pp. 2790–2798 (2017) 2, 3, 4, 12

39. Tai, Y., Yang, J., Liu, X., Xu, C.: MemNet: A persistent memory network for image restoration. In: ICCV 2017. pp. 4549–4557 (2017) 2, 3, 4
40. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: ICCV 2017. pp. 4809–4817 (2017) 4
41. Veit, A., Wilber, M.J., Belongie, S.J.: Residual networks behave like ensembles of relatively shallow networks. In: NIPS. pp. 550–558 (2016) 2
42. Wang, C., Li, Z., Shi, J.: Lightweight image super-resolution with adaptive weighted learning network. arXiv preprint **arxiv:1904.02358** (2019) 2, 3, 4, 5, 7, 10, 12, 13
43. Wang, Y., Shen, J., Zhang, J.: Deep bi-dense networks for image super-resolution. In: DICTA. pp. 1–8 (2018) 2
44. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Processing **13**(4), 600–612 (2004) 8
45. Yu, J., Fan, Y., *et al.*: Wide activation for efficient and accurate image super-resolution. arXiv preprint **arxiv:1808.08718** (2018) 7, 10, 11
46. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-repre sentations. In: Proc. 7th Int. Conf. Curves Surf. pp. 711–730 (2010) 8, 10, 12
47. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR 2018. pp. 3262–3271 (2018) 1, 12
48. Zhang, Y., Li, K., Li, K., *et al.*: Image super-resolution using very deep residual channel attention networks. In: ECCV 2018. pp. 294–310 (2018) 2, 3, 4, 5, 6, 7, 8, 10, 11, 12
49. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR 2018. pp. 2472–2481 (2018) 2, 3, 4, 6, 7, 8, 10, 11
50. Zhao, X., Zhang, H., Liu, H., Qin, Y., Zhang, T., Zou, X.: Single MR image super-resolution via channel splitting and serial fusion network. arXiv preprint **arxiv:1901.06484** (2019) 2
51. Zhao, X., Zhang, Y., Zhang, T., Zou, X.: Channel splitting network for single MR image super-resolution. TIP **28**(11), 5649–5662 (2019) 1, 2, 6