# Lightweight Image Super-Resolution with Information Multi-distillation Network

Zheng Hui
School of Electronic Engineering, Xidian University
Xi'an, China
zheng_hui@aliyun.com

Xinbo Gao
School of Electronic Engineering, Xidian University
Xi'an, China
xbgao@mail.xidian.edu.cn

Yunchu Yang
School of Electronic Engineering, Xidian University
Xi'an, China
yc_yang@aliyun.com

Xiumei Wang*
School of Electronic Engineering, Xidian University
Xi'an, China
wangxm@xidian.edu.cn

## ABSTRACT

In recent years, single image super-resolution (SISR) methods using deep convolution neural network (CNN) have achieved impressive results. Thanks to the powerful representation capabilities of the deep networks, numerous previous ways can learn the complex non-linear mapping between low-resolution (LR) image patches and their high-resolution (HR) versions. However, excessive convolutions will limit the application of super-resolution technology in low computing power devices. Besides, super-resolution of any arbitrary scale factor is a critical issue in practical applications, which has not been well solved in the previous approaches. To address these issues, we propose a lightweight information multi-distillation network (IMDN) by constructing the cascaded information multi-distillation blocks (IMDB), which contains distillation and selective fusion parts. Specifically, the distillation module extracts hierarchical features step-by-step, and fusion module aggregates them according to the importance of candidate features, which is evaluated by the proposed contrast-aware channel attention mechanism. To process real images with any sizes, we develop an adaptive cropping strategy (ACS) to super-resolve block-wise image patches using the same well-trained model. Extensive experiments suggest that the proposed method performs favorably against the state-of-the-art SR algorithms in term of visual quality, memory footprint, and inference time. Code is available at https://github.com/Zheng222/IMDN.

## CCS CONCEPTS

• **Computing methodologies → Computational photography**; *Reconstruction*; Image processing.

## KEYWORDS

image super-resolution; lightweight network; information multi-distillation; contrast-aware channel attention; adaptive cropping strategy

**ACM Reference Format:**
Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. 2019. Lightweight Image Super-Resolution with Information Multi-distillation Network. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351084

## 1 INTRODUCTION

Single image super-resolution (SISR) aims at reconstructing a high-resolution (HR) image from its low-resolution (LR) observation, which is inherently ill-posed because many HR images that can be downsampled to an identical LR image. To address this problem, numerous image SR methods [11, 12, 25, 27, 36, 38] based on deep neural architectures [7, 9, 23] have been proposed and shown prominent performance.

Dong *et al.* [4, 5] first developed a three-layer network (SRCNN) to establish a direct relationship between LR and HR. Then, Wang *et al.* [31] proposed a neural network according to the conventional sparse coding framework and further designed a progressive upsampling style to produce better SR results at the large scale factor (*e.g.*, ×4). Inspired by VGG model [23] that used for ImageNet classification, Kim *et al.* [12, 13] first pushed the depth of SR network to 20 and their model outperformed SRCNN by a large margin. This indicates a deeper model is instructive to enhance the quality of generated images. To accelerate the training of deep network, the authors introduced global residual learning with a high initial learning rate. At the same time, they also presented a deeply-recursive convolutional network (DRCN), which applied recursive learning to SR problem. This way can significantly reduce the model parameters. Similarly, Tai *et al.* proposed two novel networks, and one is a deep recursive residual network (DRRN) [24], another is a persistent memory network (MemNet) [25]. The former mainly utilized recursive learning to reach the goal of economizing parameters. The latter model tackled the long-term dependency problem existed in the previous CNN architecture by several memory blocks that stacked with a densely connected structure [9]. However, these two

[*] Corresponding author

*MM '19, October 21–25, 2019, Nice, France*

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351084

algorithms required a long time and huge graphics memory consumption both in the training and testing phases. The primary reason is the inputs sent to these two models are interpolation version of LR images and the networks have not adopted any downsampling operations. This scheme will bring about a huge computational cost. To increase testing speed and shorten the testing time, Shi *et al.* [22] first performed most of the mappings in low-dimensional space and designed an efficient sub-pixel convolution to upsample the resolutions of feature maps at the end of SR models.

To the same end, Dong *et al.* proposed fast SRCNN (FSRCNN) [6], which employed a learnable upsampling layer (transposed convolution) to accomplish post-upsampling SR. Afterward, Lai *et al.* presented the Laplacian pyramid super-resolution network (LapSRN) [14] to progressively reconstruct higher-resolution images. Some other work such as MS-LapSRN [15] and progressive SR (ProSR) [29] also adopt this progressive upsampling SR framework and achieve relatively high performance. EDSR [18] made a significant breakthrough in term of SR performance, which won the competition of NTIRE 2017 [1, 26]. The authors removed some unnecessary modules (*e.g.*, Batch Normalization) of the SRResNet [16] to obtain better results. Based on EDSR, Zhang *et al.* incorporated densely connected block [9, 27] into residual block [7] to construct a residual dense network (RDN). Soon they exploited the residual-in-residual architecture for the very deep model and introduced channel attention mechanism [8] to form the very deep residual attention networks (RCAN) [36]. More recently, Zhang *et al.* also introduced spatial attention (non-local module) into the residual block and then constructed residual non-local attention network (RNAN) [37] for various image restoration tasks.

The major trend of these algorithms is increasing more convolution layers to improve performance that measured by PSNR and SSIM [30]. As a result, most of them suffered from large model parameters, huge memory footprints, and slow training and testing speeds. For instance, EDSR [18] has about 43M parameters, 69 layers, and RDN [38] achieved comparable performance, which has about 22M parameters, over 128 layers. Another typical network is RCAN [36], its depth up to 400 but the parameters are about 15.59M. However, these methods are still not suitable for resource-constrained equipment. For the mobile devices, the desired practice should be to pursuing higher SR performance as much as possible when the available memory and inference time are constrained in a certain range. Many cases require not only the performance but also high execution speed, such as video applications, edge devices, and smartphones. Accordingly, it is significant to devise a lightweight but efficient model for meeting such demands.

Concerning the reduction of the parameters, many approaches adopted the recursive manner or parameter sharing strategy, such as [13, 24, 25]. Although these methods did reduce the size of the model, they increased the depth or the width of the network to make up for the performance loss caused by the recursive module. This will lead to spending a great lot of calculating time when performing SR processing. To address this issue, the better way is to design the lightweight and efficient network structures that avoid using recursive paradigm. Ahn *et al.* developed CARN-M [2] for mobile scenario through a cascading network architecture, but it is at the cost of a substantial reduction on PSNR. Hui *et al.* [11] proposed an information distillation network (IDN) that explicitly divided the preceding extracted features into two parts, one was retained and another was further processed. Through this way, IDN achieved good performance at a moderate size. But there is still room for improvement in term of performance.

Another factor that affects the inference speed is the depth of the network. In the testing phase, the previous layer and the next layer have dependencies. Simply, conducting the computation of the current layer must wait for the previous calculation is completed. But multiple convolutional operations at each layer can be processed in parallel. Therefore, the depth of model architecture is an essential factor affecting time performance. This point will be verified in Section 4.

As to solving the different scale factors (×2, ×3, ×4) SR problem using a single model, previous solutions pretreated an image to the desired size and using the fully convolutional network without any downsampling operations. This way will inevitably lead to a substantial increase in the amount of calculation.

To address the above issues, we propose a lightweight information multi-distillation network (IMDN) for better balancing performance against applicability. Unlike most previous small parameters models that use recursive structure, we elaborately design an information multi-distillation block (IMDB) inspired by [11]. The proposed IMDB extracts features at a granular level, which retains partial information and further treats other features at each step (layer) as illustrated in Figure 2. For aggregating features distilled by all steps, we devise a contrast-aware channel attention layer, specifically related to the low-level vision tasks, to enhance collected various refined information. Concretely, we exploit more useful features (edges, corners, textures, *et al.* ) for image restoration. In order to handle SR of any arbitrary scale factor with a single model, we need to scale the input image to the target size, and then employ the proposed adaptive cropping strategy (see in Figure 4) to obtain image patches of appropriate size for lightweight SR model with downsampling layers.

The contributions of this paper can be summarized as follows:

- We propose a lightweight information multi-distillation network (IMDN) for fast and accurate image super-resolution. Thanks to our information multi-distillation block (IMDB) with contrast-aware attention (CCA) layer, we achieve competitive results with a modest number of parameters (refer to Figure 6).
- We propose the adaptive cropping strategy (ACS), which allows the network included downsampling operations (*e.g.*, convolution layer with a stride of 2) to process images of any arbitrary size. By adopting this scheme, the computational cost, memory occupation, and inference time can dramatically reduce in the case of treating indefinite magnification SR.
- We explore factors affecting actual inference time through experiments and find the depth of the network is related to the execution speed. It can be a guideline for guiding a lightweight network design. And our model achieves an excellent balance among visual quality, inference speed, and memory occupation.

# 2 RELATED WORK

## 2.1 Single image super-resolution

With the rapid development of deep learning, numerous methods based on convolutional neural network (CNN) have been the mainstream in SISR. The pioneering work of SR is proposed by Dong *et al.* [4, 5] named SRCNN. The SRCNN upscaled the LR image with bicubic interpolation before feeding into the network, which would cause substantial unnecessary computational cost. To address this issue, the authors removed this pre-processing and upscaled the image at the end of the net to reduce the computation in [6]. Lim *et al.* [18] modified SRResNet [16] to construct a more in-depth and broader residual network denoted as EDSR. With the smart topology structure and a significantly large number of learnable parameters, EDSR dramatically advanced the SR performance. Zhang *et al.* [38] introduced channel attention [8] into the residual block to further boost very deep network (more than 400 layers without considering the depth of channel attention modules). Liu [19] explored the effectiveness of non-local module applied to image restoration. Similarly, Zhang *et al.* [37] utilized non-local attention to better guide feature extraction in their trunk branch for reaching better performance. Very recently, Li *et al.* [17] exploited feedback mechanism that enhancing low-level representation with high-level ones.

For lightweight networks, Hui *et al.* [11] developed the information distillation network for better exploiting hierarchical features by separation processing of the current feature maps. And Ahn [2] designed an architecture that implemented a cascading mechanism on a residual network to boost the performance.

## 2.2 Attention model

Attention model, aiming at concentrating on more useful information in features, has been widely used in various computer vision tasks. Hu *et al.* [8] introduced squeeze-and-excitation (SE) block that models channel-wise relationships in a computationally efficient manner and enhances the representational ability of the network, showing its effectiveness on image classification. CBAM [32] modified the SE block to exploit both spatial and channel-wise attention. Wang *et al.* [28] proposed the non-local module to generate the wide attention map by calculating the correlation matrix between each spatial point in the feature map, then the attention map guided dense contextual information aggregation.

# 3 METHOD

## 3.1 Framework

In this section, we describe our proposed information multi-distillation network (IMDN) in detail, its graphical depiction is shown in Figure 1(a). The upsampler (see Figure 1(b)) includes one 3 × 3 convolution with $3 \times s^2$ output channels and a sub-pixel convolution. Given an input LR image $\mathbf{I}^{LR}$, its corresponding target HR image $\mathbf{I}^{HR}$. The super-resolved image $\mathbf{I}^{SR}$ can be generated by

$$\mathbf{I}^{SR} = H_{IMDN}\left(\mathbf{I}^{LR}\right), \tag{1}$$

where $H_{IMDN}(\cdot)$ is our IMDN. It is optimized with mean absolute error (MAE) loss followed most of previous works [2, 11, 18, 36, 38]. Given a training set $\left\{\mathbf{I}_i^{LR}, \mathbf{I}_i^{HR}\right\}_{i=1}^{N}$ that has $N$ LR-HR pairs. Thus, the loss function of our IMDN can be expressed by

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|H_{IMDN}\left(I_i^{LR}\right) - I_i^{HR}\right\|_1, \tag{2}$$

where $\Theta$ indicates the updateable parameters of our model and $\|\cdot\|_1$ is $l_1$ norm. Then we give more details about the entire framework.

We first conduct LR feature extraction implemented by one 3 × 3 convolution with 64 output channels. Then, the key component of our network utilizes multiple stacked information multi-distillation blocks (IMDBs) and assembles all intermediate features to fusing by a 1 × 1 convolution layer. This scheme, intermediate information collection (IIC), is beneficial to guarantee the integrity of the collected information and can further boost the SR performance by increasing very few parameters. The final upsampler only consists of one learnable layer and a non-parametric operation (sub-pixel convolution) for saving parameters as much as possible.

## 3.2 Information multi-distillation block

As depicted in Figure 2, our information multi-distillation block (IMDB) is constructed by progressive refinement module, contrast-aware channel attention (CCA) layer, and a 1 × 1 convolution that is used to reduce the number of feature channels. The whole block adopts residual connection. The main idea of this block is extracting useful features little by little like DenseNet [9]. Then we give more details to these modules.

**Table 1: PRM architecture. The columns represent layer, kernel-size, stride, input channels, and output channels. The symbols, C, and L denote a convolution layer, and Leaky ReLU ($\alpha$ = 0.05).**

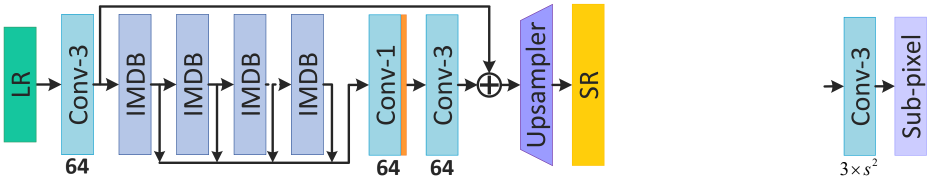
| Layer | Kernel | Stride | Input_channel | Output_channel |
|---|---|---|---|---|
| CL | 3 | 1 | 64 | 64 |
| CL | 3 | 1 | 48 | 64 |
| CL | 3 | 1 | 48 | 64 |
| CL | 3 | 1 | 48 | 16 |

### 3.2.1 *Progressive refinement module.*

As labeled with the gray box in Figure 2, the progressive refinement module (PRM) first adopts the 3 × 3 convolution layer to extract input features for multiple subsequent distillation (refinement) steps. For each step, we employ channel split operation on the preceding features, which will produce two-part features. One is preserved and the other portion is fed into the next calculation unit. The retained part can be regarded as the refined features. Given the input features $F_{in}$, this procedure in the $n$-th IMDB can be described as

$$\begin{aligned}
F^n_{refined\_1}, F^n_{coarse\_1} &= Split^n_1\left(CL^n_1\left(F^n_{in}\right)\right),\\
F^n_{refined\_2}, F^n_{coarse\_2} &= Split^n_2\left(CL^n_2\left(F^n_{coarse\_1}\right)\right),\\
F^n_{refined\_3}, F^n_{coarse\_3} &= Split^n_3\left(CL^n_3\left(F^n_{coarse\_2}\right)\right),\\
F^n_{refined\_4} &= CL^n_4\left(F^n_{coarse\_3}\right),
\end{aligned} \tag{3}$$

where $CL^n_j$ denotes the $j$-th convolution layer (including Leaky ReLU) of the $n$-th IMDB, $Split^n_j$ indicates the $j$-th channel split layer



(a) IMDN

(b) Upsampler

**Figure 1: The architecture of information multi-distillation network (IMDN). (a) The orange box represents Leaky ReLU activation function and the details of IMDB is shown in Figure 2. (b) s represents the upscale factor.**

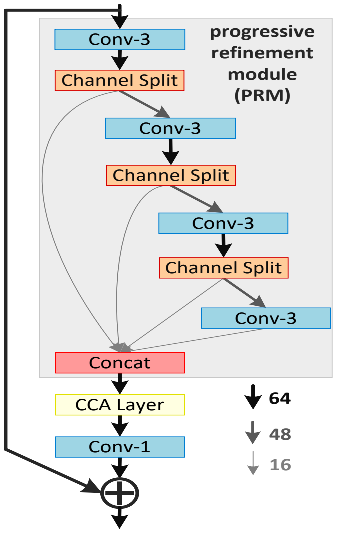


**Figure 2: The architecture of our proposed information multi-distillation block (IMDB). Here, 64, 48, and 16 all represent the output channels of the convolution layer. "Conv-3" denotes the $3 \times 3$ convolutional layer, and "CCA Layer" indicates the proposed contrast-aware channel attention (CCA) that is depicted in Figure 3. Each convolution followed by a Leaky ReLU activation function except for the last $1 \times 1$ convolution. We omit them for concise.**

of the $n$-th IMDB, $F^n_{refined\_j}$ represents the $j$-th refined features (preserved), and $F^n_{coarse\_j}$ is the $j$-th coarse features to be further processed. The hyperparameter of PRM architecture is shown in Table 1. The following stage is concatenating refined features from each step. It can be expressed by

$$F^n_{distilled} = \\ Concat\left(F^n_{refined\_1}, F^n_{refined\_2}, F^n_{refined\_3}, F^n_{refined\_4}\right), \tag{4}$$

where *Concat* denotes concatenation operation along the channel dimension.

*3.2.2 Contrast-aware channel attention layer.* The initial channel attention is employed in image classification task and is well-known as the squeeze-and-excitation (SE) module. In the high-level field, the importance of a feature map depends on activated high-value areas, since these regions in favor of classification or detection. Accordingly, global average/maximum pooling is utilized to capture

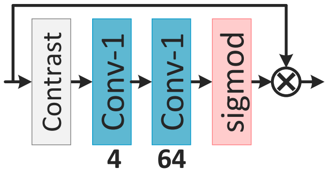


**Figure 3: Contrast-aware channel attention module.**

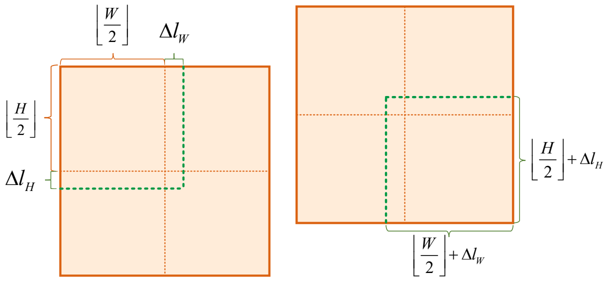
the global information in these high-level or mid-level vision. Although the average pooling can indeed improve the PSNR value, it lacks the information about structures, textures, and edges that are propitious to enhance image details (related to SSIM). As depicted in Figure 3, the contrast-aware channel attention module is special to low-level vision, *e.g.*, image super-resolution, and enhancement. Specifically, we replace global average pooling with the summation of standard deviation and mean (evaluating the contrast degree of a feature map). Let's denote $X = [x_1, \ldots, x_c, \ldots, x_C]$ as the input, which has $C$ feature maps with spatial size of $H \times W$. Therefore, the contrast information value can be calculated by

$$\begin{aligned} z_c &= H_{GC}\left(x_c\right) \\ &= \sqrt{\frac{1}{HW}\sum_{(i,j)\in x_c}\left(x_c^{i,j} - \frac{1}{HW}\sum_{(i,j)\in x_c} x_c^{i,j}\right)^2} + \\ &\quad \frac{1}{HW}\sum_{(i,j)\in x_c} x_c^{i,j}, \end{aligned} \tag{5}$$

where $z_c$ is the $c$-th element of output. $H_{GC}(\cdot)$ indicates the global contrast (GC) information evaluation function. With the assistance of the CCA module, our network can steadily improve the accuracy of SISR.

## 3.3 Adaptive cropping strategy

The adaptive cropping strategy (ACS) is special to image of any arbitrary size super-resolving. Meanwhile, it can also deal with the SR problem of any scale factor with a single model (see Figure 5). We slightly modify the original IMDN by introducing two downsampling layer and construct the current IMDN_AS (IMDN for any scales). Here, the LR and HR images have the same spatial size (height and width). To handle images whose height and width are not divisible by 4, we first cut the entire images into 4 parts and then feed them into our IMDN_AS. As illustrated in Figure 4, we can obtain 4 overlapped image patches through ACS. Take the first patch in the upper left corner as an example, and we give the


$\left\lfloor\frac{W}{2}\right\rfloor$
$\Delta l_W$
$\left\lfloor\frac{H}{2}\right\rfloor$
$\Delta l_H$
$\left\lfloor\frac{H}{2}\right\rfloor+\Delta l_H$
$\left\lfloor\frac{W}{2}\right\rfloor+\Delta l_W$

(a) The first image patch (b) The last image patch

**Figure 4: The diagrammatic sketch of adaptive cropping strategy (ACS). The cropped image patches in the green dotted boxes.**

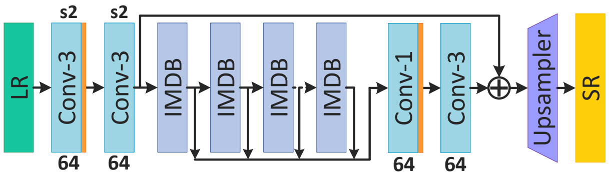


**Figure 5: The network structure of our IMDN_AS. “s2” represents the stride of 2.**

details about ACS. This image patch must satisfy

$$\begin{aligned}\left(\left\lfloor\frac{H}{2}\right\rfloor+\Delta l_H\right)\%4=0,\\ \left(\left\lfloor\frac{W}{2}\right\rfloor+\Delta l_W\right)\%4=0,\end{aligned}\tag{6}$$

where $\Delta l_H$, $\Delta l_W$ are extra increments of height and width, respectively. They can be computed by

$$\begin{aligned}\Delta l_H&=padding_H-\left(\left\lfloor\frac{H}{2}\right\rfloor+padding_H\right)\%4,\\ \Delta l_W&=padding_W-\left(\left\lfloor\frac{W}{2}\right\rfloor+padding_W\right)\%4,\end{aligned}\tag{7}$$

where $padding_H$, $padding_W$ are preset additional lengths. In general, their values are setting by

$$padding_H=padding_W=4k, k\geq 1.\tag{8}$$

Here, $k$ is an integer greater than or equal to 1. These four patches can be processed in parallel (they have the same sizes), after which the outputs are pasted to their original location, and the extra increments ($\Delta l_H$ and $\Delta l_W$) are discarded.

# 4 EXPERIMENTS

## 4.1 Datasets and metrics

In our experiments, we use the DIV2K dataset [1], which contains 800 high-quality RGB training images and widely used in image restoration tasks [18, 36–38]. For evaluation, we use five widely used benchmark datasets: Set5 [3], Set14 [33], BSD100 [20], Urban100 [10], and Manga109 [21]. We evaluate the performance of the super-resolved images using two metrics, including peak signal-to-noise ratio (PSNR) and structure similarity index (SSIM) [30]. As with existing works [2, 11, 12, 18, 24, 36, 38], we calculate the values on the luminance channel (*i.e.*, Y channel of the YCbCr channels converted from the RGB channels).

Additionally, for any/unknown scale factor experiments, we use RealSR dataset from NTIRE2019 Real Super-Resolution Challenge[1]. It is a novel dataset of real low and high resolution paired images. The training data consists of 60 real low, and high resolution paired images, and the validation data contains 20 LR-HR pairs. It is noteworthy that the LR and HR have the same size.

## 4.2 Implementation details

To obtain LR DIV2K training images, we downscale HR images with the scaling factors (×2, ×3, and ×4) using bicubic interpolation in MATLAB R2017a. The HR image patches with a size of 192 × 192 are randomly cropped from HR images as the input of our model, and the mini-batch size is set to 16. For data augmentation, we perform randomly horizontal flip and 90 degree rotation. Our model is trained by ADAM optimizer with the momentum parameter $\beta_1=0.9$. The initial learning rate is set to $2\times 10^{-4}$ and halved at every $2\times 10^5$ iterations. We set the number of IMDB to 6 in our IMDN and IMDN_AS. We apply PyTorch framework to implement the proposed network on the desktop computer with 4.2GHz Intel i7-7700K CPU, 64G RAM, and NVIDIA TITAN Xp GPU (12G memory).

## 4.3 Model analysis

In this subsection, we investigate model parameters, the effectiveness of IMDB, the intermediate information collection scheme, and adaptive cropping strategy.

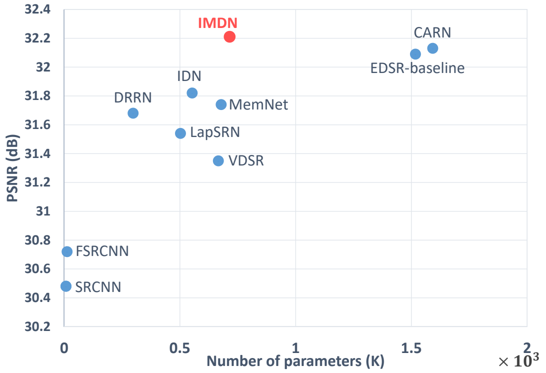


**Figure 6: Trade-off between performance and number of parameters on Set5 ×4 dataset.**

*4.3.1 Model parameters.* To construct a lightweight SR model, the parameters of the network is vital. From Table 5, we can observe that our IMDN with fewer parameters achieves comparative or better performance when comparing with other state-of-the-art methods, such as EDSR-baseline (CVPRW’17), IDN (CVPR’18), SRMDNF (CVPR’18), and CARN (ECCV’18). We also visualize the trade-off analysis between performance and model size in Figure 6. We can see that our IMDN achieves a better trade-off between the performance and model size.

**Table 2: Investigations of CCA module and IIC scheme.**

| Scale | PRM | CCA | IIC | Params | Set5 | Set14 | BSD100 | Urban100 | Manga109 |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM |
| ×4 | ✗ | ✗ | ✗ | 510K | 31.86 / 0.8901 | 28.43 / 0.7775 | 27.45 / 0.7320 | 25.63 / 0.7711 | 29.92 / 0.9003 |
| | ✓ | ✗ | ✗ | 480K | 32.01 / 0.8927 | 28.49 / 0.7792 | 27.50 / 0.7338 | 25.81 / 0.7773 | 30.16 / 0.9038 |
| | ✓ | ✓ | ✗ | 482K | 32.10 / 0.8934 | 28.51 / 0.7794 | 27.52 / 0.7341 | 25.89 / 0.7793 | 30.25 / 0.9050 |
| | ✓ | ✓ | ✓ | 499K | **32.11** / **0.8934** | **28.52** / **0.7797** | **27.53** / **0.7342** | **25.90** / **0.7797** | **30.28** / **0.9054** |

**Table 3: Comparison with original channel attention (CA) and the presented contrast-aware channel attention (CCA).**

| Module | Set5 | Set14 | BSD100 | Urban100 |
|---|---|---|---|---|
| IMDN_basic_B4 + CA | 32.0821 | 28.5086 | 27.5124 | 25.8829 |
| IMDN_basic_B4 + CCA | **32.0964** | **28.5118** | **27.5185** | **25.8916** |

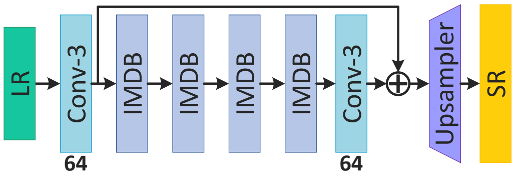


**Figure 7: The structure of IMDN_basic_B4.**

*4.3.2 Ablation studies of CCA module and IIC scheme.* To quickly validate the effectiveness of the contrast-aware attention (CCA) module and intermediate information collection (IIC) scheme, we adopt 4 IMDBs to conduct the following ablation study experiment, named IMDN_B4. When removing the CCA module and IIC scheme, the IMDN_B4 becomes IMDN_basic_B4 as illustrated in Figure 7. From Table 2, we can find out that the CCA module leads to performance improvement (PSNR: **+0.09dB**, SSIM: **+0.0012** for ×4 Manga109) only by increasing 2K parameters (which is an increase of 0.4%). The results compared with the CA module are placed in Table 3. To study the efficiency of PRM in IMDB, we replace it with three cascaded $3 \times 3$ convolution layers (64 channels) and remove the final $1 \times 1$ convolution (used for fusion). The compared results are given in Table 2. Although this network has more parameters (510K), its performance is much lower than our IMDN_basic_B4 (480K) especially on Urban100 and Manga109 datasets.

**Table 4: Quantitative evaluation of VDSR and our IMDN_AS in PSNR, SSIM, LPIPS, running time, and memory occupation.**

| Method | PSNR | SSIM | LPIPS [35] | Time | Memory |
|---|---|---|---|---|---|
| VDSR [12] | 28.75 | 0.8439 | 0.2417 | 0.0290 | 7,855M |
| IMDN_AS | **29.35** | **0.8595** | **0.2147** | **0.0041** | **3,597M** |

*4.3.3 Investigation of ACS.* To verify the efficiency of the proposed adaptive cropping strategy (ACS), we use RealSR training images to train VDSR [12] and our IMDN_AS. The results, evaluated on RealSR RGB validation dataset, are illustrated in Table 4 and we can easily observe that the presented IMDN_AS achieves better performance in term of image quality, execution speed, and footprint. Accordingly, it also suggests the proposed ACS is powerful to address SR problem of any scales.

## 4.4 Comparison with state-of-the-arts

We compare our IMDN with 11 state-of-the-art methods: SRCNN [4, 5], FSRCNN [6], VDSR [12], DRCN [13], LapSRN [14], DRRN [24], MemNet [25], IDN [11], EDSR-baseline [18], SRMDNF [34], and CARN [2]. Table 5 shows quantitative comparisons for ×2, ×3, and ×4 SR. It can find out that our IMDN performs favorably against other compared approaches on most datasets, especially at the scaling factor of ×2.

Figure 8 shows ×2, ×3 and ×4 visual comparisons on Set5 and Urban100 datasets. For “img_67” image from Urban100, we can see that grid structure is recovered better than others. It also demonstrates the effectiveness of our IMDN.

## 4.5 Running time

*4.5.1 Complexity analysis.* As the proposed IMDN mainly consists of convolutions, the total number of parameters can be computed as

$$Params = \sum_{l=1}^{L} \underbrace{n_{l-1} \cdot n_l \cdot f_l^2}_{conv} + \underbrace{n_l}_{bias}, \tag{9}$$

where $l$ is the layer index, $L$ denotes the total number of layers, and $f$ represents the spatial size of the filters. The number of convolutional kernels belong to $l$-th layer is $n_l$, and its input channels are $n_{l-1}$. Suppose that the spatial size of output feature maps is $m_l \times m_l$, the time complexity can be roughly calculated by

$$O\left(\sum_{l=1}^{L} n_{l-1} \cdot n_l \cdot f_l^2 \cdot m_l^2\right). \tag{10}$$

We assume that the size of the HR image is $m \times m$ and then the computational costs can be calculated by Equation 10 (see Table 7).

*4.5.2 Running Time.* We use official codes of the compared methods to test their running time in a feed-forward process. From Table 6, we can be informed of actual execution time is related to the depth of networks. Although EDSR has a large number of parameters (43M), it runs very fast. The only drawback is that it takes up more graphics memory. The main reason should be the convolution computation for each layer are parallel. And RCAN has only 16M parameters, its depth is up to 415 and results in very slow inference speed. Compared with CARN [2] and EDSR-baseline [18],

[1] http://www.vision.ee.ethz.ch/ntire19/

**Table 5: Average PSNR/SSIM for scale factor ×2, ×3 and ×4 on datasets Set5, Set14, BSD100, Urban100, and Manga109. Best and second best results are highlighted and underlined.**

| Method | Scale | Params | Set5 | Set14 | BSD100 | Urban100 | Manga109 |
|---|---|---|---|---|---|---|---|
| | | | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM |
| Bicubic | ×2 | - | 33.66 / 0.9299 | 30.24 / 0.8688 | 29.56 / 0.8431 | 26.88 / 0.8403 | 30.80 / 0.9339 |
| SRCNN [4] | | 8K | 36.66 / 0.9542 | 32.45 / 0.9067 | 31.36 / 0.8879 | 29.50 / 0.8946 | 35.60 / 0.9663 |
| FSRCNN [6] | | 13K | 37.00 / 0.9558 | 32.63 / 0.9088 | 31.53 / 0.8920 | 29.88 / 0.9020 | 36.67 / 0.9710 |
| VDSR [12] | | 666K | 37.53 / 0.9587 | 33.03 / 0.9124 | 31.90 / 0.8960 | 30.76 / 0.9140 | 37.22 / 0.9750 |
| DRCN [13] | | 1,774K | 37.63 / 0.9588 | 33.04 / 0.9118 | 31.85 / 0.8942 | 30.75 / 0.9133 | 37.55 / 0.9732 |
| LapSRN [14] | | 251K | 37.52 / 0.9591 | 32.99 / 0.9124 | 31.80 / 0.8952 | 30.41 / 0.9103 | 37.27 / 0.9740 |
| DRRN [24] | | 298K | 37.74 / 0.9591 | 33.23 / 0.9136 | 32.05 / 0.8973 | 31.23 / 0.9188 | 37.88 / 0.9749 |
| MemNet [25] | | 678K | 37.78 / 0.9597 | 33.28 / 0.9142 | 32.08 / 0.8978 | 31.31 / 0.9195 | 37.72 / 0.9740 |
| IDN [11] | | 553K | 37.83 / 0.9600 | 33.30 / 0.9148 | 32.08 / 0.8985 | 31.27 / 0.9196 | 38.01 / 0.9749 |
| EDSR-baseline [18] | | 1,370K | 37.99 / 0.9604 | 33.57 / 0.9175 | 32.16 / 0.8994 | 31.98 / 0.9272 | 38.54 / 0.9769 |
| SRMDNF [34] | | 1,511K | 37.79 / 0.9601 | 33.32 / 0.9159 | 32.05 / 0.8985 | 31.33 / 0.9204 | 38.07 / 0.9761 |
| CARN [2] | | 1,592K | 37.76 / 0.9590 | 33.52 / 0.9166 | 32.09 / 0.8978 | 31.92 / 0.9256 | 38.36 / 0.9765 |
| IMDN (Ours) | | 694K | **38.00** / **0.9605** | **33.63** / **0.9177** | **32.19** / **0.8996** | **32.17** / **0.9283** | **38.88** / **0.9774** |
| Bicubic | ×3 | - | 30.39 / 0.8682 | 27.55 / 0.7742 | 27.21 / 0.7385 | 24.46 / 0.7349 | 26.95 / 0.8556 |
| SRCNN [4] | | 8K | 32.75 / 0.9090 | 29.30 / 0.8215 | 28.41 / 0.7863 | 26.24 / 0.7989 | 30.48 / 0.9117 |
| FSRCNN [6] | | 13K | 33.18 / 0.9140 | 29.37 / 0.8240 | 28.53 / 0.7910 | 26.43 / 0.8080 | 31.10 / 0.9210 |
| VDSR [12] | | 666K | 33.66 / 0.9213 | 29.77 / 0.8314 | 28.82 / 0.7976 | 27.14 / 0.8279 | 32.01 / 0.9340 |
| DRCN [13] | | 1,774K | 33.82 / 0.9226 | 29.76 / 0.8311 | 28.80 / 0.7963 | 27.15 / 0.8276 | 32.24 / 0.9343 |
| LapSRN [14] | | 502K | 33.81 / 0.9220 | 29.79 / 0.8325 | 28.82 / 0.7980 | 27.07 / 0.8275 | 32.21 / 0.9350 |
| DRRN [24] | | 298K | 34.03 / 0.9244 | 29.96 / 0.8349 | 28.95 / 0.8004 | 27.53 / 0.8378 | 32.71 / 0.9379 |
| MemNet [25] | | 678K | 34.09 / 0.9248 | 30.00 / 0.8350 | 28.96 / 0.8001 | 27.56 / 0.8376 | 32.51 / 0.9369 |
| IDN [11] | | 553K | 34.11 / 0.9253 | 29.99 / 0.8354 | 28.95 / 0.8013 | 27.42 / 0.8359 | 32.71 / 0.9381 |
| EDSR-baseline [18] | | 1,555K | **34.37** / **0.9270** | 30.28 / **0.8417** | **29.09** / **0.8052** | 28.15 / **0.8527** | 33.45 / 0.9439 |
| SRMDNF [34] | | 1,528K | 34.12 / 0.9254 | 30.04 / 0.8382 | 28.97 / 0.8025 | 27.57 / 0.8398 | 33.00 / 0.9403 |
| CARN [2] | | 1,592K | 34.29 / 0.9255 | 30.29 / 0.8407 | 29.06 / 0.8034 | 28.06 / 0.8493 | 33.50 / 0.9440 |
| IMDN (Ours) | | 703K | 34.36 / **0.9270** | **30.32** / **0.8417** | **29.09** / 0.8046 | **28.17** / 0.8519 | **33.61** / **0.9445** |
| Bicubic | ×4 | - | 28.42 / 0.8104 | 26.00 / 0.7027 | 25.96 / 0.6675 | 23.14 / 0.6577 | 24.89 / 0.7866 |
| SRCNN [4] | | 8K | 30.48 / 0.8628 | 27.50 / 0.7513 | 26.90 / 0.7101 | 24.52 / 0.7221 | 27.58 / 0.8555 |
| FSRCNN [6] | | 13K | 30.72 / 0.8660 | 27.61 / 0.7550 | 26.98 / 0.7150 | 24.62 / 0.7280 | 27.90 / 0.8610 |
| VDSR [12] | | 666K | 31.35 / 0.8838 | 28.01 / 0.7674 | 27.29 / 0.7251 | 25.18 / 0.7524 | 28.83 / 0.8870 |
| DRCN [13] | | 1,774K | 31.53 / 0.8854 | 28.02 / 0.7670 | 27.23 / 0.7233 | 25.14 / 0.7510 | 28.93 / 0.8854 |
| LapSRN [14] | | 502K | 31.54 / 0.8852 | 28.09 / 0.7700 | 27.32 / 0.7275 | 25.21 / 0.7562 | 29.09 / 0.8900 |
| DRRN [24] | | 298K | 31.68 / 0.8888 | 28.21 / 0.7720 | 27.38 / 0.7284 | 25.44 / 0.7638 | 29.45 / 0.8946 |
| MemNet [25] | | 678K | 31.74 / 0.8893 | 28.26 / 0.7723 | 27.40 / 0.7281 | 25.50 / 0.7630 | 29.42 / 0.8942 |
| IDN [11] | | 553K | 31.82 / 0.8903 | 28.25 / 0.7730 | 27.41 / 0.7297 | 25.41 / 0.7632 | 29.41 / 0.8942 |
| EDSR-baseline [18] | | 1,518K | 32.09 / 0.8938 | 28.58 / **0.7813** | 27.57 / **0.7357** | 26.04 / **0.7849** | 30.35 / 0.9067 |
| SRMDNF [34] | | 1,552K | 31.96 / 0.8925 | 28.35 / 0.7787 | 27.49 / 0.7337 | 25.68 / 0.7731 | 30.09 / 0.9024 |
| CARN [2] | | 1,592K | 32.13 / 0.8937 | **28.60** / 0.7806 | **27.58** / 0.7349 | **26.07** / 0.7837 | **30.47** / **0.9084** |
| IMDN (Ours) | | 715K | **32.21** / **0.8948** | 28.58 / 0.7811 | 27.56 / 0.7353 | 26.04 / 0.7838 | 30.45 / 0.9075 |

**Table 6: Memory Consumption (MB) and average inference time (second).**

| Method | Scale | Params | Depth | BSD100 | Urban100 | Manga109 |
|---|---|---|---|---|---|---|
| | | | | Memory / Time | Memory / Time | Memory / Time |
| EDSR-baseline [18] | ×4 | 1.6M | 37 | 665 / 0.00295 | 2,511 / 0.00242 | 1,219 / 0.00232 |
| EDSR [18] | | 43M | 69 | 1,531 / 0.00580 | 8,863 / 0.00416 | 3,703 / 0.00380 |
| RDN [38] | | 22M | 150 | 1,123 / 0.01626 | 3,335 / 0.01325 | 2,257 / 0.01300 |
| RCAN [36] | | 16M | 415 | 777 / 0.09174 | 2,631 / 0.55280 | 1,343 / 0.72250 |
| CARN [2] | | 1.6M | 34 | 945 / 0.00278 | 3,761 / 0.00305 | 2,803 / 0.00383 |
| IMDN (Ours) | | 0.7M | 34 | 671 / 0.00285 | 1,155 / 0.00284 | 895 / 0.00279 |

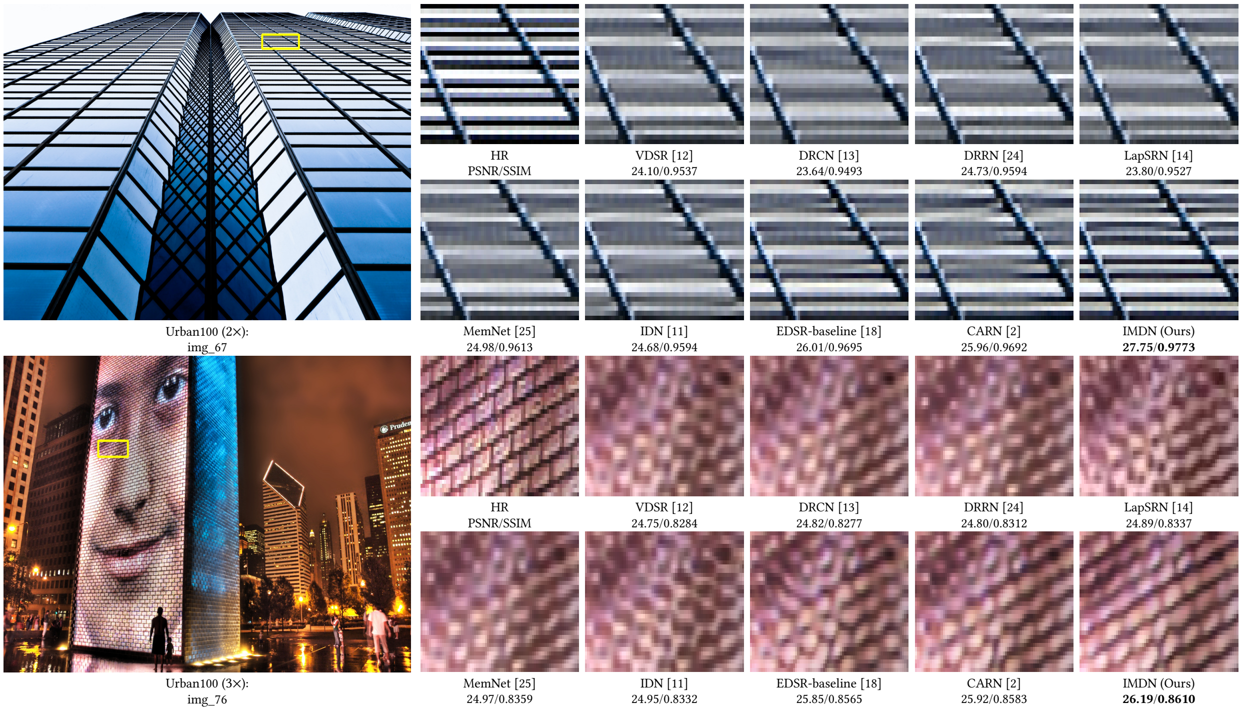


**Figure 8: Visual comparisons of IMDN with other SR methods on Set5 and Urban100 datasets.**

**Table 7: The computational costs. For representing concisely, we omit $m^2$. Least and second least computational costs are highlighted and underlined.**

| Scale | LapSRN [14] | IDN [11] | EDSR-b [18] | CARN [2] | IMDN |
|---|---|---|---|---|---|
| ×2 | **112K** | 175K | 341K | 157K | 173K |
| ×3 | 76K | **75K** | 172K | 90K | 78K |
| ×4 | 76K | 51K | 122K | 76K | **45K** |

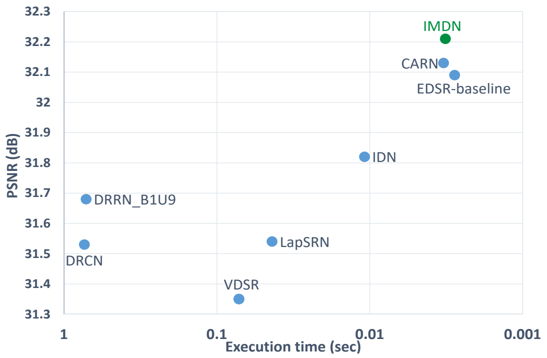


**Figure 9: Trade-off between performance and running time on Set5 ×4 dataset. VDSR, DRCN, and LapSRN were implemented by MatConvNet, while DRRN, and IDN employed Caffe package. The rest EDSR-baseline, CARN, and our IMDN utilized PyTorch.**

Our IMDN achieves dominant performance in term of memory usage and time consumption.

For more intuitive comparisons with other approaches, we provide the trade-off between the running time and performance on Set5 dataset for ×4 SR in the Figure 9. It shows our IMDN gains comparable execution time and best PSNR value.

## 5 CONCLUSION

In this paper, we propose an information multi-distillation network for lightweight and accurate single image super-resolution. We construct a progressive refinement module to extract hierarchical feature step-by-step. By cooperating with the proposed contrast-aware channel attention module, the SR performance is significantly and steadily improved. Additionally, we present the adaptive cropping strategy to solve the SR problem of an arbitrary scale factor, which is critical for the application of SR algorithms in the actual scenes. Numerous experiments have shown that the proposed method achieves a commendable balance between factors affecting practical use, including visual quality, execution speed, and memory consumption. In the future, this approach will be explored to facilitate other image restoration tasks such as image denoising and enhancement.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61432014, 61772402, U1605252, 61671339 and 61871308, in part by the National Key Research and Development Program of China under Grant 2016QY01W0200, in part by National High-Level Talents Special Support Program of China under Grant CS31117200001.

# REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 126–135.

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. 2018. Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network. In *European Conference on Computer Vision (ECCV)*. 252–268.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference (BMVC)*.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*. 184–199.

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 295–307.

[6] Chao Dong, Chen Change Loy, and Xiaoou Tang. 2016. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*. 391–407.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[8] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7132–7141.

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4700–4708.

[10] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5197–5206.

[11] Zheng Hui, Xiumei Wang, and Xinbo Gao. 2018. Fast and Accurate Single Image Super-Resolution via Information Distillation Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 723–731.

[12] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1646–1654.

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Deeply-recursive convolutional network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1637–1645.

[14] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 624–632.

[15] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2018. Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, and Andrew Cunningham. 2017. Photo-Realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4681–4690.

[17] Zhen Li, Jinglei Yang, Zheng Liu, Xiaoming Yang, Gwanggil Jeon, and Wei Wu. 2019. Feedback Network for Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 136–144.

[19] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. 2018. Non-Local Recurrent Network for Image Restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1680–1689.

[20] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision (ICCV)*. 416–423.

[21] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2017. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications* 76, 20 (2017), 21811–21838.

[22] Wenzhe Shi, Jose Caballero, Huszár, Ferenc, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1874–1883.

[23] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference for Learning Representations (ICLR)*.

[24] Ying Tai, Jian Yang, and Xiaoming Liu. 2017. Image super-resolution via deep recursive residual network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3147–3155.

[25] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. 2017. MemNet: A Persistent Memory Network for Image Restoration. In *IEEE International Conference on Computer Vision (ICCV)*. 4539–4547.

[26] Radu Timofte, Shuhang Gu, Jiqing Wu, Luc Van Gool, Lie Zhang, and et al. 2017. NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 965–976.

[27] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. 2017. Image Super-Resolution Using Dense Skip Connections. In *IEEE International Conference on Computer Vision (ICCV)*. 4799–4807.

[28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7794–7803.

[29] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkin-Hornung, and Christopher Schroers. 2018. A Fully Progressive Approach to Single-Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 977–986.

[30] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[31] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. 2015. Deep networks for image super-resolution with sparse prior. In *IEEE International Conference on Computer Vision (ICCV)*. 370–378.

[32] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional Block Attention Module. In *The European Conference on Computer Vision (ECCV)*. 3–19.

[33] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces (ICCS)*. 711–730.

[34] Kai Zhang, Wangmeng Zuo, and Lei Zhang. [n. d.]. Learning a Single Convolutional Super-Resolution Network for Multiple Degradations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3262–3271.

[35] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595.

[36] Yulun Zhang, kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In *European Conference on Computer Vision (ECCV)*. 286–301.

[37] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. 2019. Residual Non-local Attention Networks for Image Restoration. In *International Conference on Learning Representations (ICLR)*.

[38] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual Dense Network for Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2472–2481.