# BSR: A Balanced Framework for Single Image Super Resolution

*Abstract*—**The reconstruction effect of Single Image Super Resolution (SISR) has been greatly improved over the traditional statistic and feature based methods since the wide application of Deep Convolutional Neural Networks (DCNNs). Most recent studies mainly focused on the complexity of the neural network models and the stability of the training process without paying much attention to imbalance problems in the fields of super resolution. In this paper, we study three imbalance effects as sample imbalance, feature imbalance, and object function imbalance. A novel framework, which is called Balanced Super Resolution (BSR) is thus proposed to tackle these issues. Specifically, we propose a random filter sampling algorithm to form balanced training sets during batch training. Meanwhile, feature mapping group, which is a kind of residual structure, is introduced to forward various groups of low-level information to high-level. A light spatial attention mechanism is also proposed to improve the effectiveness of residual features. Furthermore, we study the object functions in traditional SISR networks and deploy a hybrid L1/L2/Lp structure that favors visually-stable SR output. The proposed design achieves persistently better image quality than state-of-the-art DCNN methods in both subjective and objective measurements.**

*Index Terms*— **balance, deep convolutional neural network, framework, super-resolution**

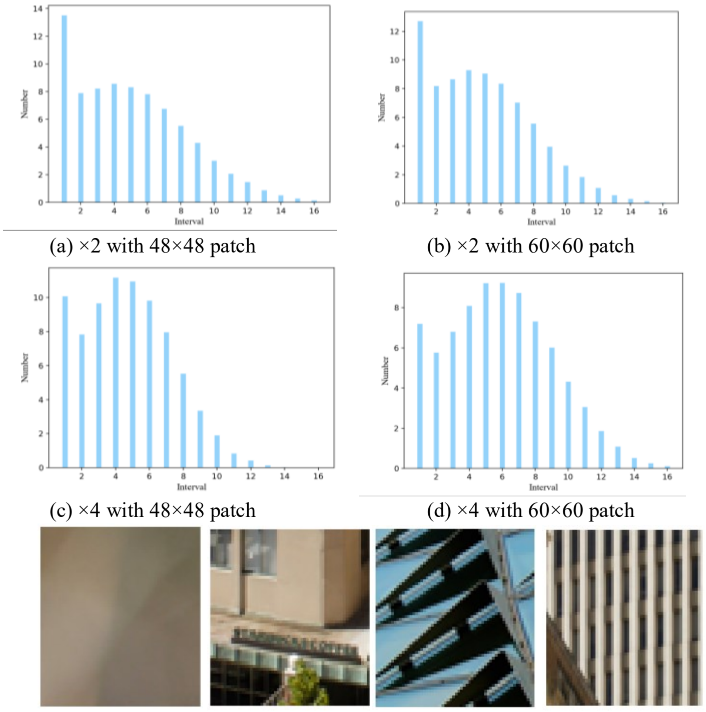
## I. Introduction

SUPER resolution is a traditional signal processing algorithm which obtains one or more high-resolution (HR) images from one or more low-resolution (LR) versions of the same scene by increasing the number of pixels per unit area in an image[1]. Single image SR is a challenging problem as a specific LR input can correspond to numerous HR images with different visual quality.

With the rapid development of signal processing techniques, a substantial amount of statistical methods is deployed to solve this one-to-many mapping issue. Nowadays these methods can be classified into three main categories, interpolation-based[2], reconstruction-based[3], and learning-based[4]. Learning-based methods, especially those deep convolutional neural networks, gain much attention for SR applications since the introduction of pioneer work SRCNN in [5] due to their extraordinary performance in both peak signal-to-noise ratio (PSNR) and perceptual quality as compared with non-deep-learning based methods.

A typical DCNN for SR usually has three function blocks, feature extraction, feature mapping, and HR reconstruction, respectively. Most state-of-the-art methods focus on the feature mapping stage which should maximize the DCNN performance on non-linear mapping, and thus deeper networks are more preferable in literature [6][7][8][9]. In addition, skip connection[10][11] has become a useful network structure in DCNN for SR which helps to improve training stability and attention on the underlying lower level characteristics[12].

Most recent works [8][13][14] of DCNN tend to maximize objective or subjective image quality by various network architecture designs or different training strategies. However, they tend to ignore the inherent features of each input image without considering the texture variance. The trained model would be biased if the input samples have imbalanced statistical characteristics[15][16][17].



(a) ×2 with 48×48 patch

(b) ×2 with 60×60 patch

(c) ×4 with 48×48 patch

(d) ×4 with 60×60 patch

(e) Samples with gradients interval value of 1, 4, 8, and 12

Fig. 1. Gradient distribution in DIV2K dataset

As illustrated in Fig. 1 (a) – (d), the gradient distribution from random samples of DIV2K [19] dataset is calculated, where we partition the entire distribution into 16 evenly distributed intervals. Although the exact distributions of ×2 and ×4 scaling with 48×48 and 60×60 patch sizes are different, the following observation holds: the gradient of DIV2K is not evenly distributed and the patches featuring smaller gradients account for the majority. This is partially because DIV2K mainly contains 2K (resolution of 1920×1080 and above) images which feature content of plain textures such as sky, cloud, and monochromatic objects. From Fig. 1(e), we can see that typical patches with different gradients interval values of 1, 4, 8, and 12, where 8 and 12 represent complex and repeatedly artificial patterns. Therefore, if we train a neural network with too many samples from this "biased" dataset, it would inevitably tend to demonstrate more fuzzy HR images than the network trained with samples full of texture details.

Another disadvantage of traditional DCNN of SR is the missing of multiple feature scales. As the neural networks get deeper, higher level of abstraction features become dominant which lead to blur or even unpleasant textures in the final reconstructed images. This is because in essence, low-level vision task as SR is different from high-level tasks such as classification or object detection where higher abstraction features are preferable for final decision. Therefore, LR images which contain most low-frequency information should be forwarded and leveraged to generate the final HR outputs.

Finally, in most DCNNs, the optimization target is typically the minimization of the mean squared error (MSE) between the recovered and ground truth HR images, which helps to maximize PSNR, as the goal of SR is to output a scaled-up image as close to HR as possible. However, commonly used object functions pay more attention to pixels with larger absolute values as they introduce larger variance in final loss calculation. It is a consensus that *L1* norm leads to sharper edges than *L2* norm as *L1* has a bigger loss than *L2* when the difference is small[20][21][22].

In this work, we propose a balanced super-resolution framework (BSR), and according to our knowledge, this is the first work in the literature to study imbalance problem in SR DCNNs. The main contributions are three-fold: (1) we propose an efficient random filter sampling method to form a balanced training batch; (2) we propose a multi-scale feature map with spatial attention mechanism network architecture, which is consisted of around 240 convolution layers; (3) we adopt a hybrid *L1/L2/Lp* object function and study its effectiveness. As shown in Fig. 2, from a test image of Urban100[23] dataset, our BSR achieves better visual results compared with state-of-the-art methods.

The rest of the paper is organized as follows. In section II, relevant background and literatures on SR are depicted. We describe the proposed BSR in Section III. Experimental results are presented and analyzed in Section IV, and we conclude the paper in Section V.

## II. Related work

The sole purpose of SISR is to find an accurate mapping between LR and HR images. Numerous SR methods have been studied in the computer vision community, and can be classified into three categories[24]: interpolation-based, reconstruction-based, and learning-based.

As a pioneer work in deep learning based method, SRCNN[5] learns the mapping from LR to HR images in an end-to-end manner, and achieves superior performance against previous interpolation and reconstruction based works.

EDSR[25] won the 2018 NTIRE [26] competition mainly due to the removal of batch normalization (BN) layers, where BN restrains the scale of feature space via a trainable scale factor and translation factor which needs to be avoided in SR mapping.

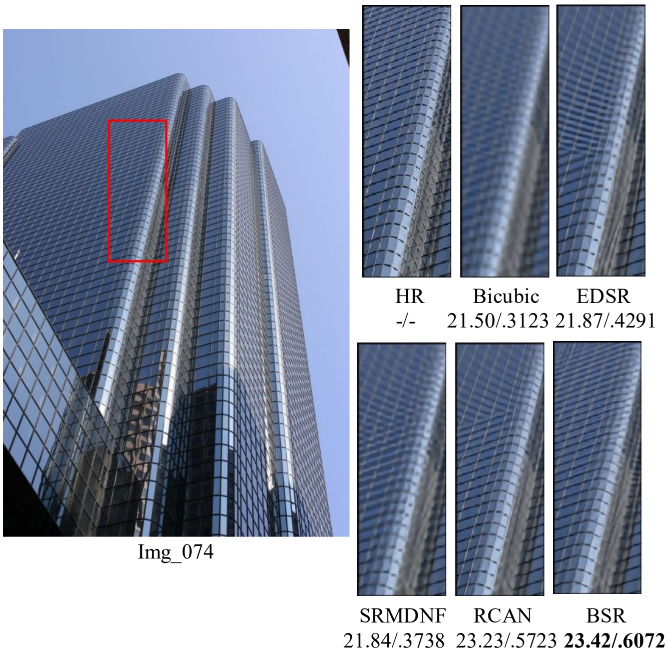


Fig. 2. Visual results comparison with bicubic degradatio

SRGAN [27], which is a pioneer work of applying generative adversarial network (GAN) and perceptual loss in SR, targets to recover photo-realistic textures from heavily down-sampled images. Although it can alleviate the blurring and over-smoothing artifacts to some degree, its predicted results may not be faithfully reconstructed and produce unpleasing artifacts [6].

RCAN [6] proposed a residual in residual structure to form very deep network as many as 400 layers which achieves excellent results. SAN [8] utilizes a novel trainable second-order channel attention module as a substitute for channel attention layer in RCAN to adaptively rescale the channel-wise features.

Unfortunately, all the works above focus on network structure to achieve better subjective/objective results, and none of them pay attention to various imbalance issues presented in this paper. Here we propose a balanced SR framework, which we will detail in next section.

## III. Balanced SR

### A. Architecture

The entire framework of proposed BSR is illustrated in Fig. 3, and the working process is depicted as below:

*Step0*: Down-sample. The original HR images are down-sampled using either bicubic or blur-down method to generate corresponding LR images.

*Step1*: Select images. In LR image datasets, we randomly pick up images to form a batch.

*Step2*: Select patch. From the selected images in *step1*, we randomly select patches to form a batch.

*Step3*: Calculate patch information capacity (PIC). For each batch, the corresponding PIC is computed. The detailed calculation process is depicted in Section B below.

*Step4*: Qualify patch. Based on the current batch's statistic distribution (determined by previously qualified batches) and current patch's PIC, we either qualify or disqualify current patch. If current patch is disqualified, we'll move forward to quality the next patch until the total number of qualified patches specified in batch training is satisfied.

*Step5*: Form balanced batch. The selected patches with qualified PIC distribution form a balanced batch to train the proposed BSR.

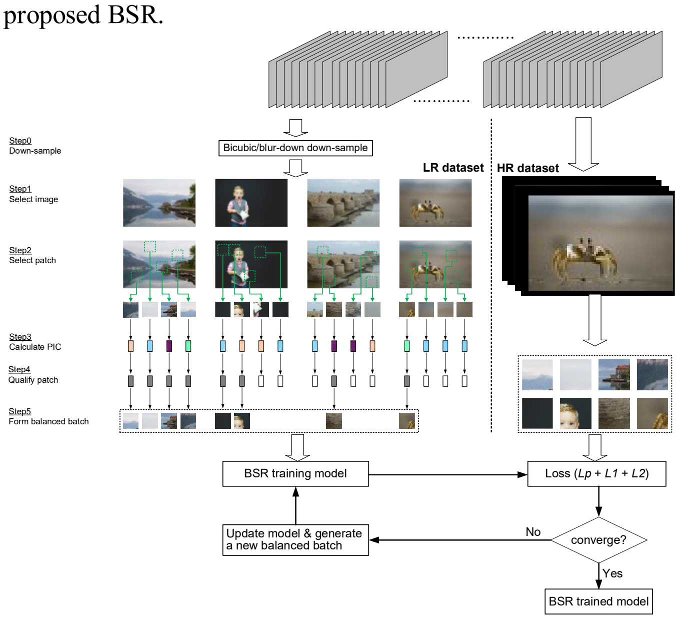


Fig. 3 Balanced SR framework

After one iteration of batch training, if the BSR is not converged, the current model gets updated and a new balanced batch is generated using *step1* to *step5*.

## B. Random Filter Sampling (RFS)

Traditional neural network training process usually involves selecting several LR images randomly from a specific dataset and crop to a fixed size which is called a patch in order to form a batch input. The output is the corresponding high-resolution SR patch. A patch pair can be described as: $P_{in}: [x, y, w, h] \rightarrow P_{out}: [x \times scale, y \times scale, w \times scale, h \times scale]$ where x and y are the selected patch's left upper corner coordinates, and w and h are the patch's width and height, accordingly.

For each patch, we propose a metric for its information measurement, which is called patch information capacity and defined as following:

$$PIC = \sum_{ch=0}^{2} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} G_{sobel}[P_{in}(x, y, w, h)] \qquad (1)$$

It represents how much texture information is contained in the patch. The gradient magnitude is calculated by Sobel operations as following:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (2)$$

$$G_{sobel} = |G_x| + |G_y| \qquad (3)$$

where $G_x$ and $G_y$ are the vertical and horizontal derivative approximation respectively, and $G_{sobel}$ is the total magnitude of gradient. We use sum of absolute value to calculate $G_{sobel}$ instead of traditional squared-root to reduce the effect of outliers and speed up calculation. Gradients for all pixels from all three channels (red, green, blue) of current patch are summed together to obtain patch's PIC. Please note that for each pixel of the input patch, there are three $G_{sobel}$ values calculated for Red, Green, and Blue channels, individually. We select the maximum value of these three as the input to compute *L1* norm of entire patch. We use *L1* norm instead of *L2* norm due to its robustness. As *L2* norm squares the error, it is more sensitive to outliers in the training dataset.

During batch construction process, suppose each batch contains N patches from M randomly selected LR candidate images where M is usually smaller than or equal to N. To obtain an evenly distributed probability of texture information measured in PIC for current batch, the entire PIC distribution is split into $p$ intervals $K = [0\ k_1 \ldots k_p]$. The PIC of N balanced patches should be uniform distributed.

The entire RFS can be summarized in Table 1:

Table 1 Flow chart of RFS

```
Input: M – number of randomly selected LR candidate images
       N – number of patches for each batch
       K – set of PIC distribution interval
       T – training dataset
1.  Initialize sampled vector V = [NULL ... NULL]_{1×k}. Randomly
select M low resolution images from training dataset T.
2.  While V ≠ K:
2.1) Randomly crop each input image to generate one patch $p_0$,
$p_1$ … $p_{m-1}$
2.2) Compute PIC for each patch as equation (1), and output PIC
vector [$PIC_0$, $PIC_1$ ... $PIC_m$]
2.3) For i = 0: k-1:
       If $PICi \in$ interval $N_i$ and $V_i < K_i$:
          Put current patch into batch
Output: Batch
```

## C. Network Architecture

The overall network architecture is illustrated in Fig. 4, which is consisted of three main building blocks, shallow feature extraction, feature mapping group (FMG), and reconstruction, respectively.

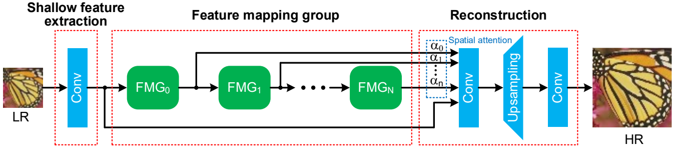


Fig. 4 Network architecture

The working horse of proposed BSR is multiple FMGs. Each FMG is consisted of five residual blocks (RB), while each RB includes several layers as shown in Fig. 6 below.

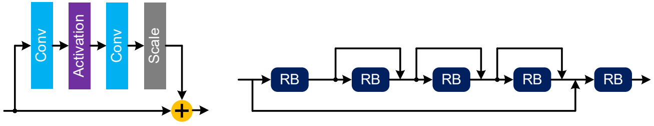


Fig. 5 **Left**: One residual block (RB). **Right**: One FMG

### 1) Shallow feature extraction

The shallow feature extraction block is a single convolution layer to generate low level features $F_o$:

$$F_o = H_{SF}(I_{LR})$$

where $H_{SF}$ represents the convolution layer.

### 2) Feature mapping group

As shown in Fig. 4, FMGs are cascaded to obtain LR features at different scales:

$$F_{i=1\ldots n} = H_{FMG}(F_0)$$

where $i$ and $H_{FMG}$ are feature scale factor and FMG mapping function.

The basic idea of FMGs is to provide balanced input for the up-sampling block of the last stage. The lowest level of feature information $F_0$ is derived from the input of FMG0 which is the direct output of shallow feature extraction block, while the highest level of feature information $F_n$ is the input of the reconstruction block.

As DCNN goes deeper, low level information usually gets diminished, and this is why most recent DCNNs for SISR always have a long skip connection between the low level feature stage and final reconstruction stage. We call this scale imbalance when the texture information from LR gradually fades out as the number of convolution layers grows. The output from very deep convolution layers contains abstract information which is good for high level vision tasks such as object detection and classification, but less important for low level vision tasks such as SR where texture reconstruction plays a vital role.

Many DCNNs such as [6][28][29][30] could recover the contour of the objects but lost the details around, which causes unpleasant visual effects such as blurry. Works such as RCAN[6] solves imbalance among feature channels in the same layer via channel attention mechanism. The proposed BSR focuses on the imbalance in full scale space and forwards all the previous lower level information to the final stage.

Each FMG is consisted of residual blocks and skip connections (SC) as shown in Fig. 5. Each residual block includes two paths, a residual path which is consisted of convolution layers and one activation layer, and an identity path as [25]. Each FMG is used to estimate a certain scale feature and contributes to evaluate higher level features by concatenation of all FMGs output to the final reconstruction stage.

#### *3) Reconstruction*

All the FMG outputs, as well as the information from shallow feature extraction block, are fed into the final reconstruction block simultaneously:

$$I_{SR} = H_{SR}(F_{i=1\ldots n})$$

where $H_{SR}$ contains operations such as scaling and reconstruction that are realized by shuffle operation.

We introduce a light spatial attention mechanism here to maximize the effect of previous block. Channel-wise concatenation with weights $\alpha$ is deployed to ensure the network focuses more on the region-of-interest (ROI). Different FMG outputs are assigned with different weights (all smaller than 1) as they represent different scale features. It should also be noted that there's a direct path from shallow feature extraction stage to reconstruction stage with no special attention (weight = 1) which indicates lower level features carry highest attention.

### *D. Object Function*

A key difference between traditional tasks in computer vision such as classification and SR is the dimensionality of the final output. For classification, it outputs a scalar or a vector while SR outputs a 2D or 3D matrix. Therefore, in SR, the imbalance exists where a single pixel or a small group of pixels gets more attention than necessary in final output if they introduce larger gradient.

An ideal SR algorithm would output a SR image which should be as close to HR image as possible. Object functions, especially for those are composed of *L2* norm, favor a high PSNR but might lead to poor perceptual quality[4]. Adding *L1* norm regularization or using *L1* norm directly is considered to improve the sharpness of reconstruction which has already been deployed in many algorithms. *Lp* norm can further improve sparsity as it generates larger gradient even when the input difference is small.

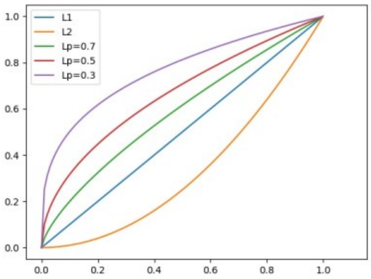


Fig. 6 Output comparison from different norms

As is shown in Fig. 6, Lp norm has a larger output difference value than L1 and L2 norm when the input difference is small.

## IV. Experimental Results

### *A. Dataset and Evaluation Metrics*

Totally 5 standard benchmark datasets are utilized to verify the performance, which include Set5, Set14, BSD100, Urban100 and Magna 109. The proposed algorithm is applied to two popular degradation models, classic bicubic degradation and blur-down degradation. The classic bicubic uses the most common setting from the recent SR literature (Matlab imresize, default settings)[15].

Table 2 Comparison results of PSNR (dB) and SSIM on various datasets with bicubic degradation model

| Method | Scale | 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 |
| | ×3 | 30.39 | 0.8682 | 27.55 | 0.7742 | 27.21 | 0.7385 | 24.46 | 0.7349 | 26.95 | 0.8556 |
| | ×4 | 28.42 | 0.8104 | 26.00 | 0.7027 | 25.96 | 0.6675 | 23.14 | 0.6577 | 24.89 | 0.7866 |
| SRCNN | ×2 | 36.66 | 0.9542 | 32.45 | 0.9067 | 31.36 | 0.8879 | 29.50 | 0.8946 | 35.60 | 0.9663 |
| | ×3 | 32.75 | 0.9090 | 29.30 | 0.8215 | 28.41 | 0.7863 | 26.24 | 0.7989 | 30.48 | 0.9117 |
| | ×4 | 30.48 | 0.8628 | 27.50 | 0.7513 | 26.90 | 0.7101 | 24.52 | 0.7221 | 27.58 | 0.8555 |
| FSRCNN | ×2 | 37.05 | 0.9560 | 32.66 | 0.9090 | 31.53 | 0.8920 | 29.88 | 0.9020 | 36.67 | 0.9710 |
| | ×3 | 33.18 | 0.9140 | 29.37 | 0.8240 | 28.53 | 0.7910 | 26.43 | 0.8080 | 31.10 | 0.9210 |
| | ×4 | 30.72 | 0.8660 | 27.61 | 0.7550 | 26.98 | 0.7150 | 24.62 | 0.7280 | 27.90 | 0.8610 |
| VDSR | ×2 | 37.53 | 0.9590 | 33.05 | 0.9130 | 31.90 | 0.8960 | 30.77 | 0.9140 | 37.22 | 0.9750 |
| | ×3 | 33.67 | 0.9210 | 29.78 | 0.8320 | 28.83 | 0.7990 | 27.14 | 0.8290 | 32.01 | 0.9340 |
| | ×4 | 31.35 | 0.8830 | 28.02 | 0.7680 | 27.29 | 0.0726 | 25.18 | 0.7540 | 28.83 | 0.8870 |
| LapSRN | ×2 | 37.52 | 0.9591 | 33.08 | 0.9130 | 31.08 | 0.8950 | 30.41 | 0.9101 | 37.27 | 0.9740 |
| | ×3 | 33.82 | 0.9227 | 29.87 | 0.8320 | 28.82 | 0.7980 | 27.07 | 0.8280 | 32.21 | 0.9350 |
| | ×4 | 31.54 | 0.8850 | 28.19 | 0.7720 | 27.32 | 0.7270 | 25.21 | 0.7560 | 29.09 | 0.8900 |
| MemNet | ×2 | 37.78 | 0.9597 | 33.28 | 0.9142 | 32.08 | 0.8978 | 31.31 | 0.9195 | 37.72 | 0.9740 |
| | ×3 | 34.09 | 0.9248 | 30.00 | 0.8350 | 28.96 | 0.8001 | 27.56 | 0.8376 | 32.51 | 0.9369 |
| | ×4 | 31.74 | 0.8893 | 28.26 | 0.7723 | 27.40 | 0.7281 | 25.50 | 0.7630 | 29.42 | 0.8942 |
| EDSR | ×2 | 38.11 | 0.9602 | 33.92 | 0.9195 | 32.32 | 0.9013 | 32.93 | 0.9351 | 39.10 | 0.9773 |
| | ×3 | 34.65 | 0.9280 | 30.52 | 0.8462 | 29.25 | 0.8093 | 28.80 | 0.8653 | 34.17 | 0.9476 |
| | ×4 | 32.46 | 0.8968 | 28.80 | 0.7876 | 27.71 | 0.7420 | 26.64 | 0.8033 | 31.02 | 0.9148 |
| SRMDNF | ×2 | 37.79 | 0.9601 | 33.32 | 0.9159 | 32.05 | 0.8985 | 31.33 | 0.9204 | 38.07 | 0.9761 |
| | ×3 | 34.12 | 0.9254 | 30.04 | 0.8382 | 28.97 | 0.8025 | 27.57 | 0.8398 | 33.00 | 0.9403 |
| | ×4 | 31.96 | 0.8925 | 28.35 | 0.7787 | 27.49 | 0.7337 | 25.68 | 0.7731 | 30.09 | 0.9024 |
| RDN | ×2 | 38.24 | 0.9614 | 34.01 | 0.9212 | 32.34 | 0.9017 | 32.89 | 0.9353 | 39.18 | 0.9780 |
| | ×3 | 34.71 | 0.9296 | 30.57 | 0.8468 | 29.26 | 0.8093 | 28.80 | 0.8653 | 34.13 | 0.9484 |
| | ×4 | 32.47 | 0.8990 | 28.81 | 0.7871 | 27.72 | 0.7419 | 26.61 | 0.8028 | 31.00 | 0.9151 |
| RCAN | ×2 | 38.27 | 0.9614 | 34.12 | 0.9216 | 32.41 | 0.9027 | 33.34 | 0.9384 | 39.44 | 0.9786 |
| | ×3 | 34.74 | 0.9299 | 30.65 | 0.8482 | 29.32 | 0.8111 | 29.09 | 0.8702 | **34.44** | 0.9499 |
| | ×4 | 32.63 | 0.9002 | 28.87 | 0.7889 | 27.77 | 0.7436 | 26.82 | 0.8087 | 31.22 | 0.9173 |
| SAN | ×2 | **38.31** | **0.9620** | 34.07 | 0.9213 | 32.42 | **0.9028** | 33.10 | 0.9370 | 39.32 | 0.9792 |
| | ×3 | **34.75** | 0.9300 | 30.59 | 0.8476 | **29.33** | 0.8112 | 28.93 | 0.8671 | 34.30 | 0.9494 |
| | ×4 | **32.64** | **0.9003** | **28.92** | 0.7888 | 27.78 | 0.7436 | 26.79 | 0.8068 | 31.18 | 0.9169 |
| BSR | ×2 | 38.30 | 0.9616 | **34.15** | **0.9220** | **32.42** | 0.9027 | **33.64** | **0.9404** | **39.54** | 0.9790 |
| | ×3 | 34.74 | **0.9301** | **30.66** | **0.8501** | 29.31 | **0.8113** | **29.19** | **0.8710** | 34.43 | **0.9550** |
| | ×4 | 32.64 | 0.9002 | 28.90 | **0.7890** | **27.79** | **0.7436** | **26.79** | **0.8088** | **31.23** | **0.9174** |

Table 3 Comparison results of PSNR (dB) and SSIM on various datasets with blur-down degradation model

| Method | Scale | Set5 | | Set14 | | BSD100 | | Urban100 | | Manga109 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Bicubic | ×3 | 28.78 | 0.8308 | 26.38 | 0.7271 | 26.33 | 0.6918 | 23.52 | 0.6862 | 25.46 | 0.8149 |
| SPMSR | ×3 | 32.21 | 0.9001 | 28.89 | 0.8105 | 28.13 | 0.7740 | 25.84 | 0.7856 | 29.64 | 0.9003 |
| SRCNN | ×3 | 32.05 | 0.8944 | 28.80 | 0.8074 | 28.13 | 0.7736 | 25.70 | 0.7770 | 29.47 | 0.8924 |
| FSRCNN | ×3 | 26.23 | 0.8124 | 24.44 | 0.7106 | 24.86 | 0.6832 | 22.04 | 0.6745 | 23.04 | 0.7927 |
| VDSR | ×3 | 33.25 | 0.9150 | 29.46 | 0.8244 | 28.57 | 0.7893 | 26.61 | 0.8136 | 31.06 | 0.9234 |
| IRCNN | ×3 | 33.38 | 0.9182 | 29.63 | 0.8281 | 28.65 | 0.7922 | 26.77 | 0.8154 | 31.15 | 0.9245 |
| SRMDNF | ×3 | 34.01 | 0.9242 | 30.11 | 0.8364 | 28.98 | 0.8009 | 27.50 | 0.8370 | 32.97 | 0.9391 |
| RDN | ×3 | 34.58 | 0.9280 | 30.53 | 0.8447 | 29.23 | 0.8079 | 28.46 | 0.8582 | 33.97 | 0.9465 |
| RCAN | ×3 | 34.70 | 0.9288 | 30.63 | 0.8462 | 29.32 | 0.8093 | 28.81 | 0.8647 | 34.38 | 0.9483 |
| SAN | ×3 | 34.75 | 0.9290 | **30.68** | **0.8466** | 29.33 | **0.8101** | 28.83 | 0.8646 | 34.46 | 0.9487 |
| BSR | ×3 | **34.76** | **0.9292** | 30.64 | 0.8464 | **29.34** | 0.8100 | **28.83** | **0.8648** | **34.46** | **0.9488** |

As most of literature studies, we select PSNR and SSIM as comparison metrics. PSNR is calculated for all three channels in RGB color space, while SSIM is evaluated for Y component only in YCbCr space.

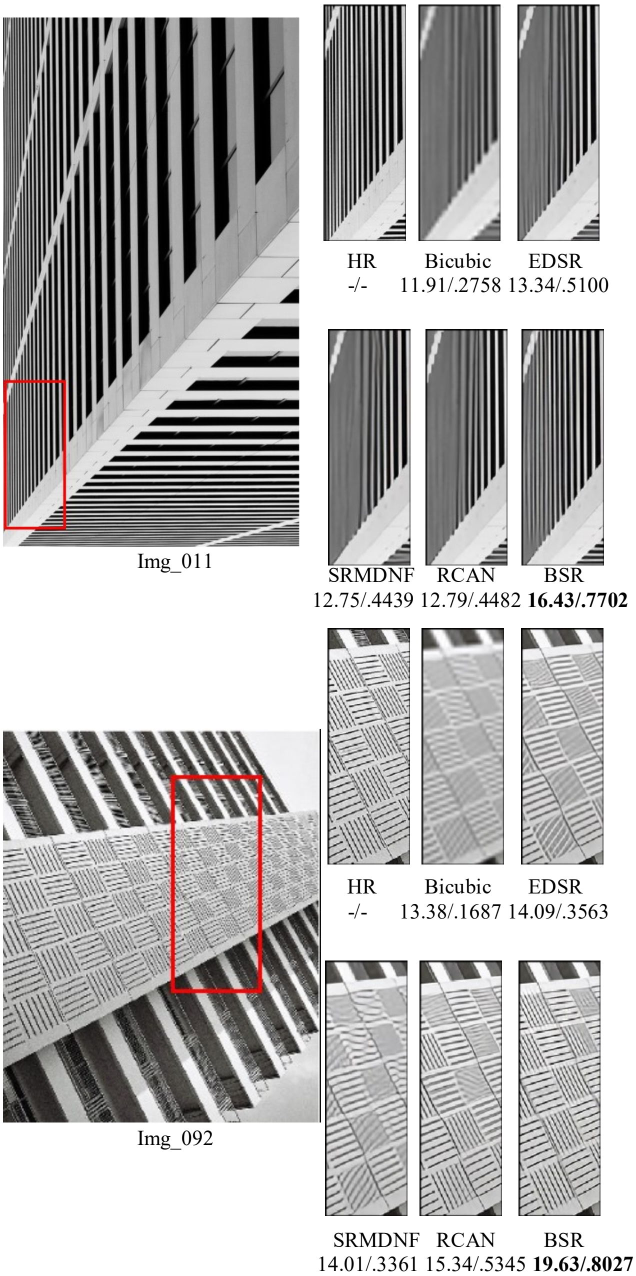
The proposed model is implemented in PyTorch framework[17], and trained with Nvidia GeForce RTX2080. We set the initial learning rate as 2e-4 and decrease by 0.1 after every 100 epochs. Data augment operations are also deployed where each input image is rotated by 90°/180°/270° randomly.

### B. *Experimental Results*

To evaluate performance, the proposed BSR is applied to restore LR images generated by bicubic and blur-down degradation model. Totally 11 state-of-the-art DCNN based SR methods, such as SRCNN[5], FSRCNN[28], VDSR[6], LapSRN[29], MemNet[31], EDSR[25], SRMD[32], NLRN[33], DBPN[34], RDN[12], RCAN[7], and SAN[8] are compared with BSR. The comparison results are depicted in Table 2.

Besides bicubic degradation, comparison of various algorithms such as SPMSR[35],IRCNN[36] using blur-down degradation model is illustrated in Table 3. For simplicity, we only perform ×3 scaling here while other scaling factors show the similar trend. The proposed BSR is able to achieve highest scores in most evaluation matrices in all experiments.

More qualitative results are illustrated in Fig. 7 below.



Img_011

HR -/- Bicubic 11.91/.2758 EDSR 13.34/.5100

SRMDNF 12.75/.4439 RCAN 12.79/.4482 BSR **16.43/.7702**

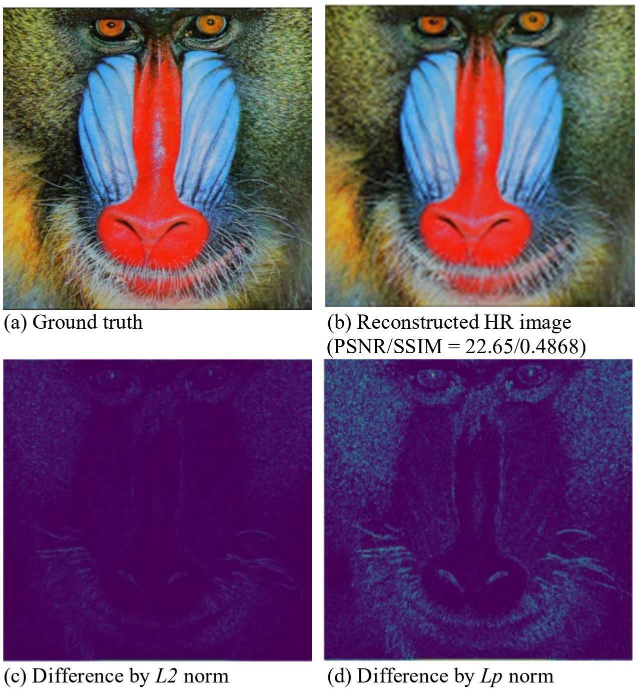
Img_092

HR -/- Bicubic 13.38/.1687 EDSR 14.09/.3563

SRMDNF 14.01/.3361 RCAN 15.34/.5345 BSR **19.63/.8027**

Fig. 7 Qualitative results

### C. Ablation Study



(a) Ground truth

(b) Reconstructed HR image (PSNR/SSIM = 22.65/0.4868)

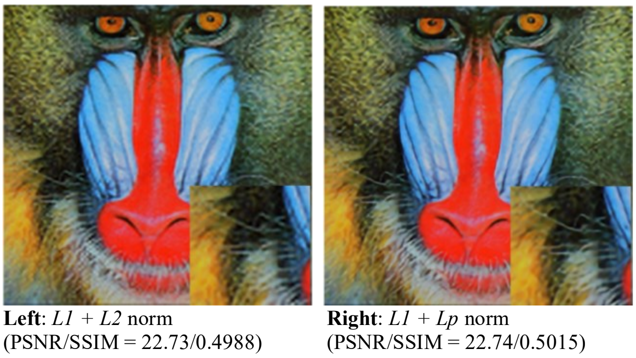
(c) Difference by *L2* norm

(d) Difference by *Lp* norm

Fig. 8 Reconstructed image vs. ground truth, calculated by different norms

As shown in Fig. 8, (a) is the ground truth from xxx. (b) is the reconstructed HR image trained with 25 epochs and the object function is *(L1 + L2)* norm. The differences between ground truth and reconstructed HR image can be measured either by *L2* norm that is illustrated in (c), or *Lp* (p=0.5) norm that is shown in (d). We can see that *Lp* norm presents more significant difference as *Lp* norm is more sensitive to image texture detail differences.

To further illustrate the effectiveness of *Lp* norm, we continue to train with 25 epochs based on the results of Fig. 8 (b). As depicted in Fig. 9, two object function combinations are evaluated, *L1 + L2* (left) and *L1 + Lp* (right). As can be seen, *L1 + Lp* norm presents a slightly better results (0.01dB in PSNR and 0.0027 in SSIM) as compared with *L1 + L2* norm which proves that *Lp* norm is better in discovering texture details.



**Left**: *L1 + L2* norm (PSNR/SSIM = 22.73/0.4988)

**Right**: *L1 + Lp* norm (PSNR/SSIM = 22.74/0.5015)

Fig. 9 Training with two object function combinations

In order to investigate the effect of each block in the proposed BSR, we perform five experiments and compare their difference as illustrated in Table 4. The results are obtained on set5 dataset (×4 scale).

Table 4 Effects of each block in SR

| | Baseline | w/o RFS | w/o SC | L2 | L2 + L1 |
|---|---|---|---|---|---|
| PNSR | **32.20** | 32.15 | 32.10 | 32.17 | 32.19 |

The baseline is our full-fledged model, which contains 10 FMGs and 12 RBs for each FMG as described in Fig. 4. It's trained with RFS and *L2/L1/Lp* joint object function.

Without RFS and only using shuffle and data augment via rotation and flip for sampling, there's a 0.5dB quality degradation (32.15dB vs. 32.20dB), which proves that RFS can significantly improve the HR quality.

If we remove SC between inner FMG output and scaling part, the degradation is

The last two columns show the effectiveness of hybrid *L1+L2+Lp* norm. As can be seen, *L1 + L2* is better than *L1* alone but less superior than all three norms combined.

## V. Conclusion

In this paper, we show the imbalance issues in SISR which includes sample imbalance, feature imbalance, and object function imbalance. To tackle these imbalance problems, a balanced SR framework is proposed which features novel random sampling algorithm during training, feature extraction group structure, as well as *Lp* object function. The proposed BSR significantly improves the SR performance. However, not all the imbalance issues in SR have been studied thoroughly and there's still lots of space for further improvements, such as how to improve neural network efficiency using multi-scale features, which would be our future research.

## References

[1] K. Nasrollashi, T. B. Moeslund, "Super-resolution: A Comprehensive Survey", *Machine Vision and Applications*, 2014, pp. 1423 - 1468.

[2] L. Wang and G. Jeon, "Bayer Pattern CFA Demosaicking Based on Multi-Directional Weighted Interpolation and Guided Filter," in *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2083-2087, Nov. 2015, doi: 10.1109/LSP.2015.2458934.

[3] J. Tarquino, A. Rueda and E. Romero, "Shearlet-based sparse representation for super-resolution in diffusion weighted imaging (DWI)," *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, 2014, pp. 3897-3900, doi: 10.1109/ICIP.2014.7025791.

[4] W. Yang et. al., "Deep learning for single image super-resolution: a brief review", *IEEE Transactions on Multimedia*, Issue 12, pp. 3106 – 3121, Dec. 2019.

[5] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.

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

[7] Y. L. Zhang *et al.*, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," Proceedings of the European Conference on Computer Vision (ECCV). 2018: 286-301. 1, 3, 5, 7

[8] T. Dai, J. Cai, Y. Zhang, S. Xia and L. Zhang, "Second-Order Attention Network for Single Image Super-Resolution," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 11057-11066, doi: 10.1109/CVPR.2019.01132.

[9] Q. Huang, D. Yang, P. Wu, H. Qu, J. Yi and D. Metaxas, "MRI Reconstruction Via Cascaded Channel-Wise Attention Network," *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy, 2019, pp. 1622-1626, doi: 10.1109/ISBI.2019.8759423.

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

[11] Zhou F, Li X, Li Z. High-frequency details enhancing DenseNet for super-resolution[J]. Neurocomputing, 2018, 290: 34-42. 1, 5

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

[13] X. T. Wang *et al.*, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks", *The European Conference on Computer Vision (ECCV) Workshops,* 2018.

[14] J. Cai, H. Zeng, H. Yong, Z. Cao and L. Zhang, "Toward Real-World Single Image Super-Resolution: A New Benchmark and a New Model," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3086-3095, doi: 10.1109/ICCV.2019.00318.

[15] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 821-830, doi: 10.1109/CVPR.2019.00091.

[16] Y. Cui, M. Jia, T. Lin, Y. Song and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Long Beach, CA, USA, 2019, pp. 9260-9269, doi: 10.1109/CVPR.2019.00949.

[17] K. Oksuz, B. C. Cam, S. Kalkan and E. Akbas, "Imbalance Problems in Object Detection: A Review," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.2981890.

[18] P. Adam, G. Sam, C. Soumith, C. Gregory, Y. Edward, D. Zachary, L. Zeming, D. Alban, A. Luca , and L. Adam. Automatic differentiation in pytorch. 2017.

[19] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, 2017, pp. 1122-1131, doi: 10.1109/CVPRW.2017.150.

[20] J. K. Pant, W. Lu and A. Antoniou, "New Improved Algorithms for Compressive Sensing Based on *lp* Norm," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 3, pp. 198-202, March 2014, doi: 10.1109/TCSII.2013.2296133.

[21] M. Kloft, U. Brefeld, S. Sonnenburg, Lp-norm multiple kernel learning[J]. Journal of Machine Learning Research, 2011, 12(Mar): 953-997. 4

[22] M. Wang, "High resolution radar imaging based on compressed sensing and adaptive Lp norm algorithm," *Proceedings of 2011 IEEE CIE International Conference on Radar*, Chengdu, 2011, pp. 206-209, doi: 10.1109/CIE-Radar.2011.6159512.

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

[24] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue and Q. Liao, "Deep Learning for Single Image Super-Resolution: A Brief Review," in *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106-3121, Dec. 2019, doi: 10.1109/TMM.2019.2919431.

[25] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, 2017, pp. 1132-1140, doi: 10.1109/CVPRW.2017.151.

[26] R. Timofte *et al.*, "NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, 2018, pp. 965-96511, doi: 10.1109/CVPRW.2018.00130.

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

[28] C. Dong, C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network[C]. *European conference on computer vision. Springer(ECCV)*, Cham, 2016: 391-407. 3, 7

[29] W. Lai, J. Huang, N. Ahuja and M. Yang, "Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2599-2613, 1 Nov. 2019, doi: 10.1109/TPAMI.2018.2865304.

[30] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan and J. Sun, "Meta-SR: A Magnification-Arbitrary Network for Super-Resolution," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 1575-1584, doi: 10.1109/CVPR.2019.00167.

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

[32] K. Zhang, W. Zuo and L. Zhang, "Learning a Single Convolutional Super-Resolution Network for Multiple Degradations," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

Salt Lake City, UT, 2018, pp. 3262-3271, doi: 10.1109/CVPR.2018.00344.

[33] D. Liu, B. Wen, Y. Fan, et al., Non-local recurrent network for image restoration[C]. Advances in Neural Information Processing Systems. 2018: 1673-1682. 7

[34] M. Haris, G. Shakhnarovich and N. Ukita, "Deep Back-Projection Networks for Single Image Super-resolution," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.3002836.

[35] T. Peleg and M. Elad, "A Statistical Prediction Model Based on Sparse Representations for Single Image Super-Resolution," in *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569-2582, June 2014, doi: 10.1109/TIP.2014.2305844.

[36] K. Zhang, W. Zuo, S. Gu and L. Zhang, "Learning Deep CNN Denoiser Prior for Image Restoration," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2808-2817, doi: 10.1109/CVPR.2017.300.