Signal Processing 162 (2019) 115–125

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast and accurate single image super-resolution via an energy-aware improved deep residual network

Yanpeng Cao[a,b], Zewei He[a,b,c], Zhangyu Ye[a,b], Xin Li[c], Yanlong Cao[a,b], Jiangxin Yang[a,b,*]

[a] State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China
[b] Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China
[c] School of Electrical Engineering and Computer Science (EECS), Louisiana State University, Baton Rouge, LA 70803, USA

A R T I C L E I N F O

Article history:
Received 8 December 2018
Revised 7 March 2019
Accepted 25 March 2019
Available online 3 April 2019

Keywords:
Super-resolution
Loss function
Residual network
Skip connections
Energy aware

A B S T R A C T

Recently, convolutional neural network (CNN) based single image super-resolution (SISR) solutions have demonstrated significant progress on restoring accurate high-resolution image based on its corresponding low-resolution version. However, most state-of-the-art SISR approaches attempt to achieve higher accuracy by pursuing deeper or more complicated models, which adversely increases computational cost. To achieve a good balance between restoration accuracy and computational speed, we make simple but effective modifications to the structure of residual blocks and skip-connections between stacked layers, and then propose a novel energy-aware training loss to adaptively adjust the restoration of high-frequency and low-frequency image regions. Extensive qualitative and quantitative evaluation results on benchmark datasets verify the effectiveness of the proposed techniques that they significantly improve SISR accuracy while causing no/ignorable extra computational loads.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Recently, single image super-resolution (SISR) technique, which aims to recover the high-resolution (HR) image from its corresponding low-resolution (LR) version, has attracted substantial attention from both the academic and industrial communities, facilitating a broad range of applications such as security surveillance, autonomous driving, and medical analysis [1]. Over the past decades, many machine learning algorithms have been developed for SISR, such as sparse coding [2,3], local linear regression [4,5] and random forest [6]. The underlying principle is that the HR image can be reconstructed by learning the nonlinear LR-to-HR mapping relationship via numerous training pairs.

Due to its powerful learning capacity, deep learning has become a prevalent tool for computer vision tasks [7–10]. Nowadays, the SISR problem has achieved significant progress by adopting deep convolutional neural networks (CNNs) architectures. SRCNN [11] proposed a three-layer CNN model to learn the nonlinear LR-to-HR mapping function. It is the first time that deep learning technique is applied to SISR problem. Although SRCNN is a lightweight CNN model, it achieved significantly better image restoration performances than many previous machine-learning-based SR methods such as sparse coding (SC) [2,3], neighbor embedding (NE) [12] and anchored neighborhood regression (ANR) [4].

Following this pioneering work, many researchers attempted to achieve more accurate SISR results by either increasing the depth of the network or deploying more complex architectures. For instance, VDSR [13] is a 20-layer deep super-resolution convolutional network (VDSR), and more recent DRRN [14], SRDenseNet [15], and MemNet [16] SISR models contain 52, 68, and 80 layers, respectively. However, the above mentioned SISR methods typically contain a large number of network parameters and require heavy computational loads. Even with graphics processing unit (GPU) acceleration, their running time is still far from real-time, which adversely decrease their applicability for image pre-processing tasks. For instance, the running times of DRRN [14] and MemNet [16] processing a 640×480 image on a PC equipped with NVIDIA GTX 1080Ti GPU (11GB memory) are 10.9856 s and 15.0263 s, respectively. Recently, a number of fast-speed SISR methods [17,18] have been proposed to overcome the slow running time limit. However, performances of these shallow CNN models, evaluated using several widely-used metrics (peak signal-to-noise ratio - PSNR, structural similarity index - SSIM, and information fidelity criterion - IFC), are not comparable with the ones adopting deeper networks.

* Corresponding author at: School of Mechanical Engineering, No. 38 Zheda Rd, Hangzhou 310027, China.
E-mail addresses: caoyp@zju.edu.cn (Y. Cao), zeweihe@zju.edu.cn (Z. He), sdcaoyl@zju.edu.cn (Y. Cao), yangjx@zju.edu.cn (J. Yang).

https://doi.org/10.1016/j.sigpro.2019.03.018
0165-1684/© 2019 Elsevier B.V. All rights reserved.

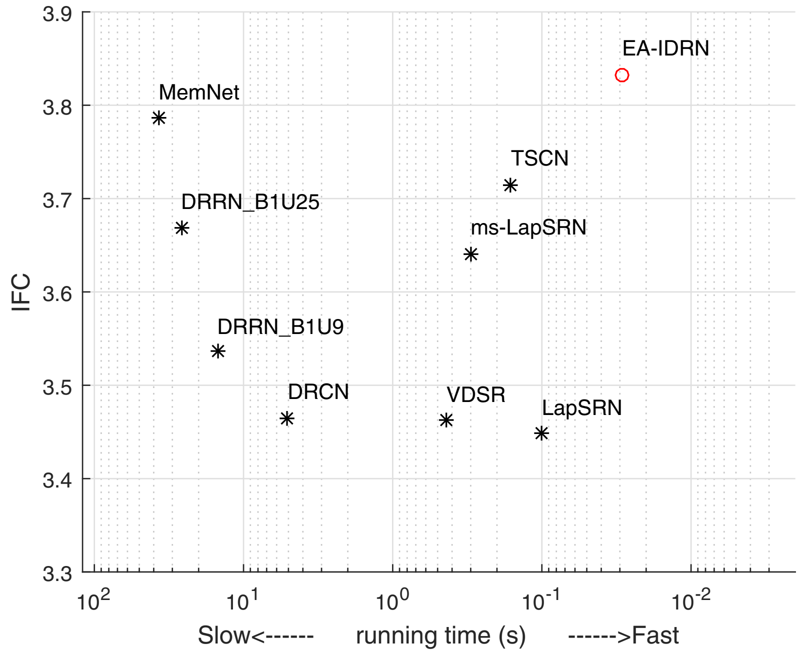

Fig. 1. Compared with the state-of-the-art SISR solutions [13,14,16,21,24–26], our proposed EA-IDRN model achieves the highest IFC value on Urban100 dataset with the scale factor $\times 4$ and runs real-time. All SISR methods are performed on a PC equipped with NVIDIA GTX 1080Ti GPU (11GB memory), Cuda 8.0 and Cudnn 5.1.

To achieve fast and accurate SISR, we present a number of effective techniques which can improve SISR accuracy while causing no/ignorable extra computational loads. **The first improvement is to optimize the network architecture.** The recently introduced residual blocks [14,19] and skip-connections [15,20–22] are proven helpful techniques to facilitate better training of deep CNNs for higher-level computer vision problems such as image classification and detection. We investigate residual blocks and skip-connections with different structures in an attempt to identify the optimal way to build up a residual network for the low-level SISR task. **The second improvement is the design of a novel energy-aware training loss.** Although $L2$ loss is the most widely used one in the state-of-the-art SISR approaches, it correlates poorly with human perception of image quality [23] and suffers from losing image details. Motivated by the fact that high-frequency edges or textures tend to disappear during image downsampling, we propose a novel energy-aware training loss to adaptively adjust the restoration of image regions with different characteristics.

Based on the above improvements (**a.** an optimized residual network architecture and **b.** an energy-aware training loss), we present a compact but powerful Energy-Aware Improved Deep Residual Network (EA-IDRN) SISR model which is very accurate and runs real-time. Our proposed 23-layer EA-IDRN model outperforms other state-of-the-art SISR methods, even including the 52-layer DRRN [14] and 80-layer MemNet [16] on Urban100 dataset with the scale factor $\times 4$ and has a real-time running speed (> 30 fps), as illustrated in Fig. 1. Overall, the contributions of this paper are mainly summarized as follows:

- We investigate a number of design options for building up deep residual networks and make modifications to the structure of residual blocks for feature extraction and skip-connections between stacked residual blocks. Experimental results show these simple but effective modifications increase SISR performance substantially.
- We propose a novel loss function, incorporating the pixel-wise gradient responses with the L_1 loss, to achieve more accurate SISR results. The energy-aware loss adaptively assigns high-valued back-propagated gradients for textured/edge image regions, thus can better recover high-frequency image details which tend to lose in low-resolution images.
- Based on the above improvements, we present an Energy-Aware Improved Deep Residual Network (EA-IDRN) SISR model, which is trained end-to-end, for fast and accurate restoration of low-resolution images. EA-IDRN achieves higher accuracy and significantly faster running time compared with state-of-the-art deep-learning-based SISR approaches [11,13,14,16,21,24–27].

The remainder of this paper is organized as follows. We first review a number of learning-based SISR methods and different choices of loss functions in Section 2. Then Section 3 provides details of important components in our proposed Energy-Aware Improved Deep Residual Network (EA-IDRN). Qualitative and quantitative evaluation results are provided in Section 4 to show the effectiveness of our method. Finally, Section 5 concludes this paper.

2. Related work

Over the past decades, substantial approaches [2,4,5,28–35] have been proposed to solve the single image super-resolution (SISR) problem. Although the interpolation-based [28,29,36,37] and reconstruction-based methods [38,39] are extremely simple and fast, they cannot achieve satisfactory restoration results (difficult to recover high-frequency signals such as edges or textures). In this paper, we focus on learning-based SISR methods which aim to infer the complex LR-to-HR mapping function based on a large number of training samples.

2.1. Machine-learning-based approaches

Freeman et al. firstly utilized learning-based methods for low-level SISR problems [40,41]. However, the solution space is too vast to explore, therefore many subsequent methods embedded some prior information to improve the accuracy and speed of SISR. Sparse coding super-resolution (SCSR) methods proposed by Yang et al. [2,3] assumed that LR and HR images share the same sparse representation. In [12,42], neighbor embedding (NE) algorithms generated super-resolution results based on the assumption that low-dimensional non-linear manifolds in LR and HR feature space have a similar local geometry. To mitigate the computational load, Timofte et al. [4,5] proposed to apply a number of linear regressors to anchor the neighbors locally. Timofte et al. also presented seven ways to improve the super-resolution performance [32]. In addition, several methods [43,44] exploited the self-similarity property in natural images and constructed the internal training pairs (without external datasets) based on the scale-space pyramid.

2.2. Deep-learning-based approaches

Recently, deep learning techniques have been employed to achieve breakthrough results of SISR. Dong et al. [11,27] proposed the first deep learning based SISR method. The Super-Resolution Convolutional Neural Network (SRCNN) is a light-weight model (three layers) but significantly surpasses the state-of-the-art methods [2,4] at that time. Following this pioneering work, Kim et al. presented deeper networks (VDSR [13] and DRCN [21]) to achieve better generalization capacity for more accuracy image restoration. However, the gradient vanishing problem caused by deeper networks will make the training process unstable. Global residual learning [13] and recursive layers [21] are employed to ease the problem. To achieve higher reconstruction accuracy, Tai et al. developed the 52-layer DRRN model [14] which utilizes local and global residual learning and recursive layers and the 80-layer MemNet model [16] which contains persistent memory units and multiple supervisions. It is experimentally shown that training/deploying a deconvolution layer [17] (or a sub-pixel layer [18]) to directly construct the HR images at the end of the network provides a feasible way to reduce the computational cost and achieve higher restoration accuracy. By adopting this post-upsampling strategy, improved SISR results are achieved in EDSR [45] (using enhanced residual blocks) [19], SRDenseNet [15](using dense blocks) [46], and RDN [47] (using residual dense blocks). More recently, Zhang et al. [48] presented a very deep residual channel attention networks model (RCAN), which reaches the state-of-the-art SISR performance. It is noted that EDSR, SRDenseNet, RDN and RCAN are trained using the high-resolution DIV2K [49,50] dataset (containing 800 training images of 2K resolution) or ImageNet [51] subset. Their training processes take a long time to complete as well as the predicting processes.

It is noted that most previous researchers attempted to achieve more accurate SISR results by either increasing depth of the network or deploying more complex architectures. However, these very deep CNN models contain a large number of parameters and cannot deliver real-time speed. It is critically important to increase the efficiency of SISR models to facilitate image pre-processing tasks. Hui et al. [25] proposed a compact two-stage convolutional network (TSCN) to achieve fast inference time and state-of-the-art SR results on four benchmark datasets simultaneously. Lai et al. presented LapSRN and ms-LapSRN models [24,26] in which progressive reconstruction strategy is employed to improve both restoration accuracy and testing speed. Towards fast and accurate SISR, we present a novel Energy-Aware Improved Deep Residual Network (EA-IDRN) SISR model, which performs favorably compared to state-of-the-art SISR methods in terms of both accuracy and efficiency.

2.3. Loss function

The loss function provides critical information to guide the tuning of weights and biases of DNN models. Despite its importance, the design of a loss function suitable for the changeling SISR task has not received too much attention yet. By far, mean square error (MSE) loss or L_2 loss is the most widely used loss function for SISR [11,13,14,16,17,21,27]. The reason behind its popularity is that its calculation is similar to the one used in PSNR which is a major SISR evaluation indicator. However, SISR models based on L_2 loss function are difficult to handle the uncertainty inherent in recovering lost high-frequency details such as textures or edges and tend to generate over-smoothed outputs [52,53]. Recently, Lim et al. [45] experimentally reported that L_1 loss is a better option than L_2 loss to achieve improved SISR performance. Furthermore, Lai et al. [24] proposed a robust Charbonnier loss function which essentially is a differentiable variant of $L1$. Researchers from Twitter presented a perceptual loss function which contains an adversarial loss and a content loss. However, the artificially generated high-frequency details may be “fake” texture patterns, which are not suitable for some applications demands accurate restoration. In this paper, we present a novel energy-aware training loss to adaptively adjust the restoration of image regions with different characteristics and achieve high-fidelity SISR results.

3. Our approach

In this section, we present an end-to-end Energy-Aware Improved Deep Residual Network (EA-IDRN) model for fast and accurate SISR. The architecture of proposed EA-IDRN model is illustrated in Fig. 2. We first present a baseline deep residual network which consists of many stacked residual blocks. Then we make improvements to this baseline architecture by modifying the core residual blocks and the skip-connections between them. Finally, we propose a novel energy-aware loss function to achieve high-quality image restoration.

3.1. Baseline architecture

Recently, deep residual networks (RseNet) have been successfully employed to solve the ill-posed SISR problem [14,20,45,54]. In this paper, we investigate different design options in an attempt to construct better ResNet structures. For this purpose, we firstly present a baseline architecture of RseNet and then make modifications to it. As illustrated in Fig. 3(a), two convolutional layers are embedded to extract the low-level features F_{-1} and F_0 on the original LR input I_{LR} as

$$F_{-1} = w_{-1} * I_{LR} + b_{-1}, \tag{1}$$

$$F_0 = w_0 * F_{-1} + b_0, \tag{2}$$

where $w_{-1:0}$ and $b_{-1:0}$ represent the filtering weights and biases of the first and second convolutional layers, $*$ denotes the convolutional operation. F_0 is then fed into a number of stacked residual blocks to extract high-level features for HR image reconstruction. Without loss of generality, we employ the residual block used in EDSR [45]. After adding N residual blocks, our baseline model also embeds a deconvolution layer at the end of the network to reconstruct the final HR output I_{SR} as

$$I_{SR} = Deconv(F_N \uparrow s), \tag{3}$$

where F_N is the output of the N_{th} residual block, and $Deconv(\cdot)$ represents the deconvolution operation which generates the input signal by a sum over convolutions of the feature maps (as opposed to the input) with learned filters [55]. $\uparrow$ denotes the up-sampling operation, and s denotes the up-sampling factor. The advantage of deploying a deconvolution layer is two-fold. First, it avoids artifacts induced by hand-crafted image pre-processing techniques (e.g., bicubic interpolation). Moreover, it accelerates SISR reconstruction process by conducting computationally expensive convolutional operations on LR images [17,24].

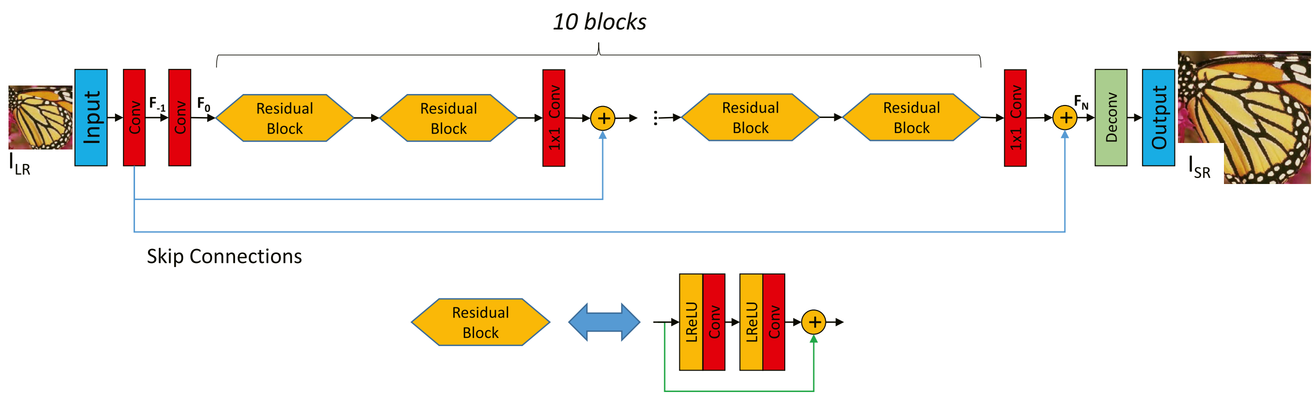

Fig. 2. The architecture of our proposed EA-IDRN model. A number of enhanced residual blocks are stacked, and skip connections and 1×1 convolutional layers are added between them to boost the overall SISR performance. The final output is directly reconstructed from low-resolution features by a deconvolution layer.

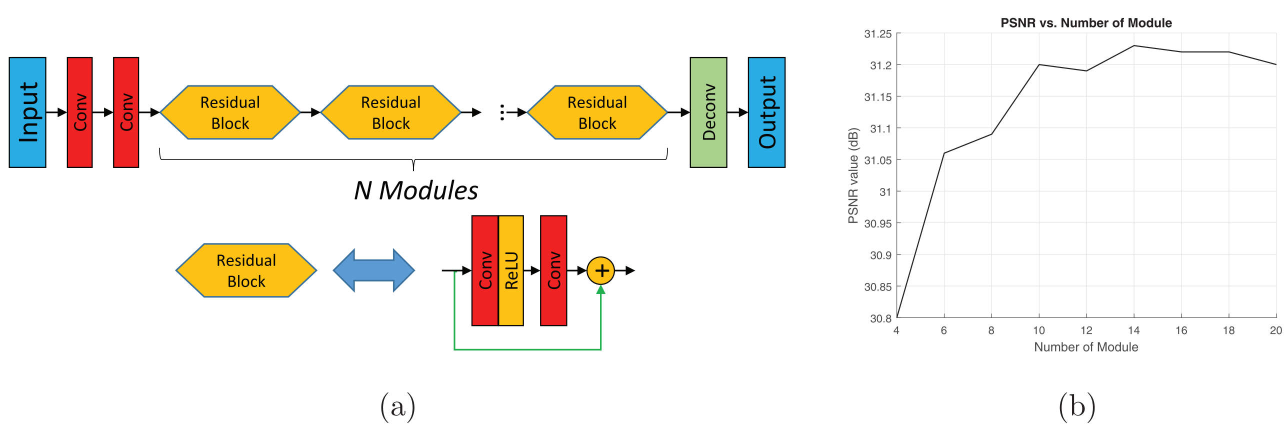

Fig. 3. (a) The architecture of the baseline deep residual network in which N residual blocks [45] are stacked. (b) The curve of PSNR vs. Number of residual blocks. All of the quantitative results are obtained on the Urban100 dataset with the scale factor $\times 2$.

The performance of the baseline models containing different number of residual blocks N are evaluated on the Urban100 dataset with the scale factor $\times 2$ and the comparative results (PSNR vs. Number of residual blocks) are shown in Fig. 3(b). It is noted that using more residual blocks can generally achieve a higher PSNR value when the number of residual blocks N is less than 10. However, the performance is marginally improved or even drops when N is larger than 10. The underlying reason is that stacking too many layers/blocks will incur the gradient vanishing/exploring problem and cause network training difficulty. Moreover, adding more residual blocks significantly increases model parameters, and thus decreases the running-time. In our baseline, we set $N = 10$ to achieve a good balance between model complexity and good performance, and then present a number of effective techniques to improve accuracy of SISR while incurring negligible computational overhead.

3.2. Network optimization

Residual blocks [14,19] and skip-connections [21,56] are two important building blocks of deep residual networks. In this section, we investigate different design options of residual blocks and skip-connections for fast and accurate SISR.

Residual block: Fig. 4 shows residual blocks with different structure designs including (a) the one used in SRResNet [20], (b) the one used in EDSR [45], (c) our modified version A, and (d) our modified version B. Following the research work of Lim et al. [45], we remove the batch normalization (BN) layers in our modified versions. The advantages are two-fold. First, it can reduce the GPU memory usage, therefore save the running time. Second, it avoids the feature normalization operation and increases the range flexibility of network. It is noticed that both SRResNet and EDSR adopt the ReLU activation function in their residual blocks. If the activation value of a certain neuron is below zero (dead neuron [57]), it will not be activated using the ReLU function and the derivatives for all preceding neurons linked to it will become zeros according to the chain rule of derivation. Too many inactive neuron chains will impede the back-propagation of derivatives and reduce the learning ability of network. In our modified version A, we replace the ReLU function (i.e., $f(x) = max(0, x)$) with the Leaky ReLU (i.e., $f(x) = max(0.2x, x)$) which assigns a nonzero slope for zero or negative activation signals. This modification ensures that there is always a non-zero gradient flowing backwards during the training process. In our modified version B, we add an extra Leaky ReLU layer before convolution operation to embed more nonlinear terms into network following the research work presented by He et al. [56]. Since the Leaky ReLU layer does not involve additional parameters and its mathematical calculation is simple, only a small amount of computational cost is added in our modified version B. The above mentioned four residual blocks have different structure designs and will lead to different SISR performances. Their comparative evaluation is provided in Section 4.3.

Skip-connections: Adding skip connections between multiple-stacked layers enables gradient signal to back-propagate directly from the higher-level features to lower-level ones, alleviating the gradient vanishing/exploring problem [22]. In our implementation, element-wise addition and channel-wise concatenation achieve very similar SISR performances, while the concatenation fusion

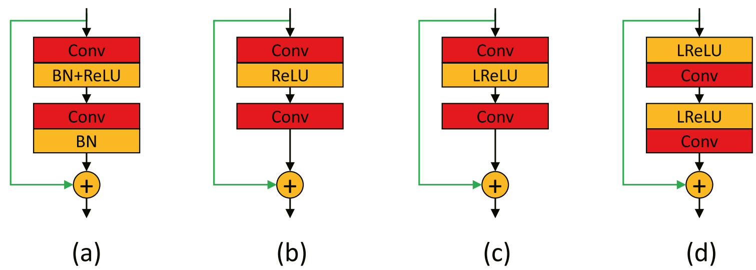

Fig. 4. Residual blocks with different structure designs. (a) Residual block used in SRResNet [20], (b) residual block used in EDSR [45] (removing the BN operations), (c) modified residual block version A (replacing the ReLU activation layer with Leaky ReLU), and (d) modified residual block version B (adding an extra Leaky ReLU layer before convolutional layers). The performance of these four different residual blocks are systematically evaluated in Section 4.3.

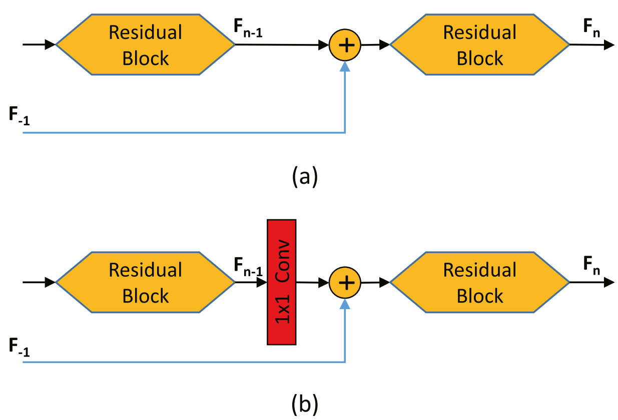

Fig. 5. Various types of skip connections. (a) Skip connections are directly added between different layers [15,20–22], (b) Skip connections and 1×1 convolutional layers are added between different layers. The performance of these two different designs is systematically evaluated in Section 4.3.

will increase the channel number of features, adding extra computational time for subsequent convolutional operations. Thus, we adopt the addition function for feature fusion. In Fig. 5, we consider two different design options. The first one directly adds skip connections between different layers, which is adopted in most CNN-based SISR approaches [15,20–22]. In this case, the output of the n_{th} residual block F_n is calculated as

$$F_n = Res(F_{-1} + F_{n-1}), \tag{4}$$

where $Res(\cdot)$ denotes the residual block. The second option utilizes a 1×1 convolutional layer and skip connections to connect different layers and the output of the n_{th} residual block is

$$F_n = Res(F_{-1} + w_n * F_{n-1} + b_n), \tag{5}$$

where w_n and b_n represent the filtering weights and biases of the nth 1×1 convolutional layer respectively. We will experimentally evaluate these two different designs and discuss the best way to deploy skip connections to achieve good accuracy and fast speed in Section 4.3.

3.3. Loss function

The loss function calculates the pixel-wise difference between the ground truth HR image and the restored SISR output, which is critical to update the weights and biases of DNN models. Despite its importance, the design of a loss function suitable for the changeling SISR task has not attracted much research attention. Most deep learning based SISR methods [11,13–17,21,27] adopt L_2 loss (i.e., mean square error loss or Euclidean loss) as the training loss. The formulation of L_2 loss and the back-propagated derivative for a pixel p are formulated as

$$\mathcal{L}_{L_2}(P) = \frac{1}{2}\sum_{p \in P} ||I^{SR}(p) - I^{GT}(p)||_2^2, \tag{6}$$

$$\partial \mathcal{L}_{L_2}(P)/\partial I^{SR}(p) = I^{SR}(p) - I^{GT}(p), \tag{7}$$

where p indicates the index of a pixel in the image patch P, I^{GT} and I^{SR} represent the ground truth and restored images, and $||\cdot||_2$ denotes the L_2 norm. Note that, although $\mathcal{L}_{L_2}(P)$ is a function of the patch as a whole, the derivatives are back-propagated for each pixel in this patch. However, as pointed out by Zhao et al. in [58], the widely used L_2 loss function suffers from some well-known limitations including L_2 correlates poorly with human perception [23] and L_2 loss is based on the assumption of a Gaussian noise model which is not valid in general.

More recently, some researchers experimentally discover that CNN models trained with L_1 loss can achieve higher restoration accuracy [45,47,58]. The L_1 loss function and its derivative for a pixel p are formulated as

$$\mathcal{L}_{L_1}(P) = \sum_{p \in P} ||I^{SR}(p) - I^{GT}(p)||_1, \tag{8}$$

$$\partial \mathcal{L}_{L_1}(P)/\partial I^{SR}(p) = sign[I^{SR}(p) - I^{GT}(p)], \tag{9}$$

where $||\cdot||_1$ denotes the L_1 norm. However, the back-propagated derivative of L_1 loss is fixed to 1 or -1 for image regions with different characteristics (e.g., textured or smooth) which is suboptimal. Motivated by the fact that high-frequency edges or textures tend to disappear during image degradation while low-frequency smooth image regions remain almost unchanged, we propose a novel gradient energy-aware (EA) training loss to adaptively adjust the strategy how high-frequency and low-frequency image regions are restored. We formulate the EA loss as the weighted sum of a L_1 loss and a gradient energy component as

$$\begin{aligned}\mathcal{L}_{EA}(P) &= \alpha \sum_{p \in P} ||I^{SR}(p) - I^{GT}(p)||_1 + \beta \sum_{p \in P} E(p)||I^{SR}(p) - I^{GT}(p)||_1 \\ &= \sum_{p \in P} [\alpha + \beta E(p)]||I^{SR}(p) - I^{GT}(p)||_1,\end{aligned} \tag{10}$$

$$\partial \mathcal{L}_{EA}(P)/\partial I^{SR}(p) = [\alpha + \beta E(p)] \cdot sign[I^{SR}(p) - I^{GT}(p)], \tag{11}$$

where α and β denote the weights for L_1 loss and the energy-aware item, respectively. In our implementation, we simply set $\alpha = \beta = 0.5$. $E(p)$ represents the pixel-wise gradient energy which is calculated as

$$E(p) = \frac{1}{2}(|G_x(p)| + |G_y(p)|), \tag{12}$$

where $G_x(p)$ and $G_y(p)$ denote the horizontal and vertical gradient values of pixel p which are calculated using two 3×3 Sobel kernels. It is noted that he energy-aware loss $\mathcal{L}_{EA}$ adaptively assigns

Table 1
The settings of deconvolution layers with the scale factors $\times 2$, $\times 3$ and $\times 4$.

scales	$\times 2$	$\times 3$	$\times 4$
Kernel size	4×4	5×5	6×6
padding	1	1	1

high-valued back-propagated gradients for high-frequency image regions, emphasizing the recovery of edges and textures which are mostly lost in low-resolution images. Comparative results of different lose functions are provided in Section 4.3.

4. Experimental results

4.1. Dataset and evaluation metric

Training datasets: Following [13,21], we train our networks on RGB91 dataset from Yang et al. [3] and another 200 images from Berkeley Segmentation Dataset (BSD) [59]. To expand our training dataset, three data augmentation techniques are utilized including: 1. Rotation: rotate image by 90°, 180° and 270°. 2. Flipping: horizontally flip image. 3. Scaling: downscale image with the scale factors of 0.9, 0.8, 0.7, 0.6 and 0.5. After data augmentation, we randomly crop these images into 48×48 sub-images to generate our HR images. The LR images are obtained by down-sampling corresponding HR images through bicubic interpolation.

Testing datasets: Five commonly used public benchmark datasets are utilized for evaluating the performance of our EA-IDRN method. Set5 [42] and Set14 [60] datasets are widely used in SISR, B100 [59] contains 100 natural images collected from BSD, and Urban100 [43] consists of 100 real-world urban scene images which are rich of structures. In addition, Manga109 [61], a collection of 109 Japanese comic images, is also employed.

Evaluation metrics: We adopt peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) [62] to evaluate the SISR performance. In addition, information fidelity criterion (IFC) metric, which correlates well with human perception [63], is also utilized to assesses the image quality. Since training is performed on the luminance channel [5,27] (Y channel of YCbCr color space), all three metrics are calculated on the Y channel accordingly. For fair comparison, we crop off boundary image pixels according to [27].

4.2. Implementation details

Our EA-IDRN model consists of $N = 10$ residual blocks. All convolutional layers have 64 channels of features and the kernel sizes are fixed to 3×3 except the 1×1 convolutional layers before skip connections. Zero-padding is employed to avoid the sizes of feature maps from shirking. In addition, the settings of deconvolution layers for scale factors $\times 2$, $\times 3$ and $\times 4$ are summarized in Table 1.

We implement our EA-IDRN with Caffe [64] framework and train this model on NVIDIA GTX 1080Ti with Cuda 8.0 and Cudnn 5.1 for 50 epochs (when fine-tunning the $\times 3$ and $\times 4$ models, 10 epochs are enough). Adam [65] solver is utilized to optimize the weights by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. The batch size is set to 64 and the learning rate is fixed to $1e-4$. For weights initialization, we adopt the method described in [66] and the biases are initialized to zeros. It takes roughly 16 hours to train our proposed EA-IDRN for scale factor $\times 2$. We train our EA-IDRN for scale factor $\times 3$ and $\times 4$ by initializing the weights with pre-trained $\times 2$ model. This fine-tuning strategy accelerates the training process and enhances the performance [45]. The source code will be made publicly available in the future.

4.3. Performance analysis

In this section, we set up ablation experiments to evaluate the performance (accuracy and running time) of different designs of (1) residual blocks, (2) skip connections, and (3) loss function. According to Section 3.1, our baseline model contains 10 stacked residual blocks to achieve a good balance between model complexity and good performance.

Residual blocks: We evaluate four residual blocks with different designs as illustrated in Fig. 4. For fair comparison, these four different residual blocks are evaluated based on the same baseline model trained using L_1 loss without skip connections and the comparative results (PSNR, SSIM, IFC, and running time) are illustrated in Table 2. We made three important observations. First of all, it is demonstrated that removing BN layers in the residual block to increase the range flexibility of network is an effective technique to achieve higher SISR accuracy with faster running time. For instance, the PSNR index increases from 31.11 dB to 31.20 dB while the running time reduces from 0.098s to 0.063s. Our experimental results are consistent with previous research of Lim et al. [45]. Second, using the Leak ReLU layer to replace the original ReLU in a residual block is another effective technique to improve SISR accuracy without adding extra computational costs. In our experiments, the PSNR index increases from 31.20 to 31.25 while the running time remains at 0.063 s. The underlying reason is that the Leak ReLU function will assign a nonzero slope for negative signals to avoid dead neuron and ensure a non-zero gradient always flowing backwards during the training process [57]. Finally, adding a Leaky ReLU layer before convolution operation can embed more nonlinear terms into network and lead to further improvement of SISR accuracy. The PSNR index increases from 31.25 dB to 31.28 dB. Since the mathematical calculation of Leaky ReLU function is simple, only a small amount of computational overhead is added in our modified pre-activation residual module. Our experiments show that adding 10 Leaky ReLU layers only increase the running time by 0.002 s.

Skip-connections: We evaluate two different options to incorporate skip connections into deep residual networks as illustrated in Fig. 5. The experiments are conducted based on the baseline model which employs the best performing modified residual block B and is trained using L_1 loss. Table 3. illustrates the experimental results (PSNR, SSIM, IFC, and running time) on Urban100 dataset with the scale factor $\times 2$. Although directly adding skip connections between layers at different depths is a proven effective

Table 2
We calculate the PSNR SSIM IFC values and running times of the baseline model using residual blocks (RB) with different designs on Urban100 dataset with the scale factor $\times 2$.

	RB in SRResNet [20]	RB in EDSR [45]	Modification A	Modification B
PSNR (dB)	31.11	31.20	31.25	31.28
SSIM	0.9180	0.9192	0.9196	0.9200
IFC	9.3893	9.4065	9.4384	9.5303
Time (s)	0.098	0.063	0.063	0.065

Table 3
We calculate the PSNR SSIM IFC values and running times of the baseline model using skip connection (SC) with different designs on Urban100 dataset with the scale factor $\times 2$.

	No SC	Direct SC 10 times	1×1 Conv. SC 10 times	1×1 Conv. No SC 10 times	1×1 Conv. SC 5 times
PSNR (dB)	31.28	31.29	31.36	31.24	31.34
SSIM	0.9200	0.9202	0.9206	0.9190	0.9205
IFC	9.5303	9.5153	9.5638	9.5037	9.5622
Time (s)	0.065	0.066	0.078	0.077	0.071

Table 4
We calculate the PSNR SSIM IFC values and running times of the baseline model using different loss functions on Urban100 dataset with the scale factor $\times 2$.

Loss function	L_2 loss	L_1 loss	EA loss
PSNR (dB)	31.24	31.34	31.39
SSIM	0.9194	0.9205	0.9209
IFC	9.4695	9.5622	9.7056
Time (s)	0.071	0.071	0.071

technique in many higher-level computer vision problems such as image classification and detection [56], our experimental results show that it only achieves marginal improvement for the low-level SISR task. The PSNR index slightly increases from 31.28 dB to 31.29 dB while the IFC index even drops from 9.5303 to 9.5153. In comparison, a noticeable improvement of restoration accuracy is achieved by adding a 1×1 convolutional layer before the skip connection. The feature maps extracted in a residual block is adaptively adjusted through a 1×1 convolutional layer before combined with output of a shallower layer, facilitating the training of more distinct features for high-quality image restoration. As the result, the PSNR index increases from 31.28 dB to 31.36 dB, SSIM index increases from 0.9200 to 0.9206, and IFC index increases from 9.5303 to 9.5638. To verify the improvement is not caused by the increase of the number of network layers (adding 10 1×1 convolutional layers), we also employ 1×1 convolutional layers without adding skip connections (1×1 Conv. No SC 10 times). As shown in Table 3, it is observed that stacking more layers into the network without adding skip connections will even lead to the decrease of SISR accuracy (PSNR decreases from 31.28 dB to 31.24 dB, SSIM decreases from 0.9200 to 0.9190, and IFC decreases from 9.5303 to 9.5037). The underlying reason for this degradation is the gradient vanishing problem triggered by deeper layers. Our experiments show that skip connections provide a useful technique to alleviate the gradient vanishing problem [15] and adding a 1×1 convolutional layer before each skip connection achieves the optimal SISR performance. Furthermore, we investigate different ways to deploy skip connections. Instead of densely connect the first convolutional layer with every single residual block (10 times), we deploy the 1×1 convolution and skip connection every two residual blocks (5 times). It is noted that the SISR accuracy marginally drops (e.g., PSNR index decreases from 31.36 dB to 31.34 dB, SSIM index decreases from 0.9206 to 0.9205, and IFC index decreases from 9.5638 to 9.5622) while the computational time substantially improves from 0.078 s to 0.071 s. Therefore, we implement 1×1 convolution and skip connection every two residual blocks to achieve a good balance between restoration accuracy and computational speed.

Loss function: We evaluate three different loss functions (L_1 loss, L_2 loss, and our proposed EA loss) using the baseline model which employs the modified residual block B as the basic module and contains 5 1×1 Conv. skip connections. Table 4 lists the PSNR, SSIM and IFC indexes and running times of three different loss functions on Urban100 dataset with the scale factor $\times 2$. We notice that training with our proposed EA loss reaches the highest PSNR

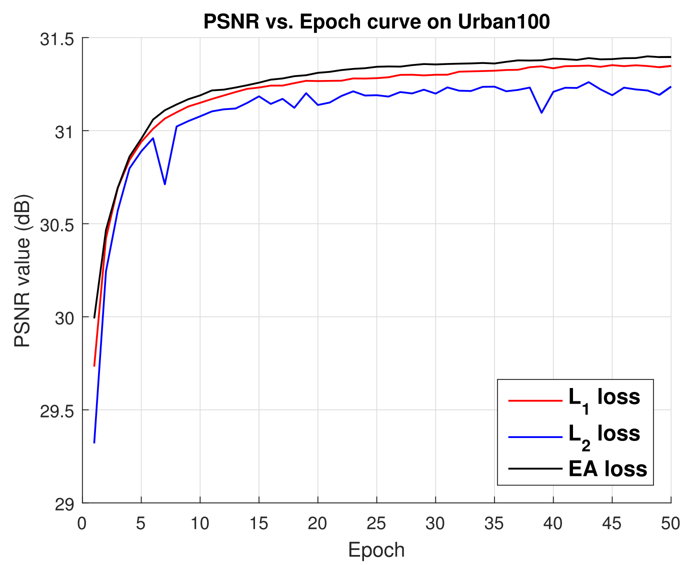

Fig. 6. The PSNR vs. Epoch curves driven by three different loss functions: $L1$, $L2$ and EA. The experiments are conducted on Urban100 dataset with the scale factor $\times 2$.

value of 31.39 dB, which surpasses the L_1 loss by 0.05 dB and the L_2 loss by 0.15 dB. Although L_2 loss is the most widely used one, it tends to over-smooth the image details and causes perceptually poor quality. Our experimental results show that L_1 loss is a better option of loss function for image restoration tasks, which are consistent with previous research works [45,47,48,67]. However, L_1 loss function cannot differentiate image regions with different characteristics (e.g., high-frequency edge region and low-frequency smooth region) and restore them equally. In comparison, our proposed gradient EA training loss allows our model to adaptively adjust how high-frequency and low-frequency image regions are restored. Gradient energy information is utilized to emphasize the information recovery for regions with textures or edges to achieve higher SISR accuracy. It is worth mentioning that EA loss does not incur any extra computational overhead during the testing phase. We also visualize the convergence process of these three different training losses in Fig. 6. It is observed that our proposed EA loss achieves higher PSNR values while using less training epochs. Fig. 7 visualize the computed gradient energy and the comparative SISR results using L_1 loss and our proposed EA loss. It is observed that the proposed energy-aware loss adaptively assigns high-valued back-propagated gradients for high-frequency image regions, leading to better SISR restoration results particularly for regions with textures or edges.

4.4. Comparisons with state-of-the-arts

In this section, we compare our proposed EA-IDRN with a number of state-of-the-art SISR methods qualitatively and quantitatively. Two machine-learning-based approaches (Aplus [5] and

Table 5
Benchmark results of several state-of-the-art SISR methods. We compare the average PSNR(dB)/SSIM values with the scale factors ×2, ×3 and ×4 on Set5, Set14, B100, Urban100 and Manga109. ***Red*** and **blue** indicate the best and the second best performance, respectively. It is noted that the metrics are calculated on Y channel.

Scale	Method	Set5 PSNR / SSIM	Set14 PSNR / SSIM	B100 PSNR / SSIM	Urban100 PSNR / SSIM	Manga109 PSNR / SSIM
×2	Bicubic	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	26.88 / 0.8403	30.81 / 0.9341
	Aplus [5]	36.54 / 0.9544	32.28 / 0.9056	31.21 / 0.8863	29.20 / 0.8938	35.37 / 0.9680
	SelfExSR [43]	36.50 / 0.9536	32.22 / 0.9034	31.17 / 0.8853	29.52 / 0.8965	35.12 / 0.9660
	SRCNN [27]	36.66 / 0.9542	32.45 / 0.9067	31.36 / 0.8879	29.51 / 0.8946	35.60 / 0.9663
	VDSR [13]	37.53 / 0.9587	33.03 / 0.9124	31.90 / 0.8960	30.76 / 0.9140	37.15 / 0.9738
	DRCN [21]	37.63 / 0.9588	33.04 / 0.9118	31.85 / 0.8942	30.75 / 0.9133	37.63 / 0.9740
	LapSRN [24]	37.44 / 0.9581	32.96 / 0.9117	31.78 / 0.8941	30.39 / 0.9093	37.21 / 0.9731
	DRRN [14]	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188	37.88 / 0.9749
	MemNet [16]	37.78 / 0.9597	33.28 / 0.9142	32.08 / 0.8978	**31.31** / 0.9195	38.03 / ***0.9755***
	TSCN [25]	**37.88** / **0.9602**	33.28 / 0.9147	**32.09** / **0.8985**	31.29 / **0.9198**	**38.07** / 0.9750
	ms-LapSRN [26]	37.70 / 0.9590	33.25 / 0.9138	32.02 / 0.8970	31.13 / 0.9180	37.71 / 0.9747
	EA-IDRN (ours)	***37.91*** / ***0.9603***	***33.32*** /***0.9154***	***32.12*** /***0.8990***	***31.39*** / ***0.9209***	***38.23*** / **0.9751**
×3	Bicubic	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	24.46 / 0.7349	26.96 / 0.8546
	Aplus [5]	32.58 / 0.9088	29.13 / 0.8188	28.29 / 0.7835	26.03 / 0.7973	29.93 / 0.8120
	SelfExSR [43]	32.64 / 0.9097	29.15 / 0.8196	28.29 / 0.7840	26.46 / 0.8090	29.61 / 0.9050
	SRCNN [27]	32.75 / 0.9090	29.29 / 0.8215	28.41 / 0.7863	26.24 / 0.7991	30.48 / 0.9117
	VDSR [13]	33.66 / 0.9213	29.77 / 0.8314	28.82 / 0.7976	27.14 / 0.8279	32.00 / 0.9329
	DRCN [21]	33.82 / 0.9226	29.76 / 0.8311	28.80 / 0.7963	27.15 / 0.8276	32.31 / 0.9360
	LapSRN [24]	– / –	– / –	– / –	– / –	– / –
	DRRN_B1U25 [14]	34.03 / 0.9244	29.96 / 0.8349	**28.95** / 0.8004	**27.53** / ***0.8378***	**32.71** / 0.9379
	MemNet [16]	34.09 / 0.9248	***30.00*** / 0.8350	***28.96*** / 0.8001	***27.56*** / **0.8376**	***32.79*** / ***0.9388***
	TSCN [25]	**34.18** / **0.9256**	**29.99** / 0.8351	**28.95** / 0.8012	27.46 / 0.8362	32.68 / **0.9381**
	ms-LapSRN [26]	– / –	– / –	– / –	– / –	– / –
	EA-IDRN (ours)	***34.19*** / ***0.9260***	***30.00*** / ***0.8358***	***28.96*** / ***0.8018***	27.43 / 0.8366	32.69 / **0.9381**
×4	Bicubic	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577	24.91 / 0.7846
	Aplus [5]	30.28 / 0.8603	27.32 / 0.7491	26.82 / 0.7087	24.32 / 0.7183	27.03 / 0.8510
	SelfExSR [43]	30.30 / 0.8620	27.38 / 0.7516	26.84 / 0.7106	24.80 / 0.7377	26.80 / 0.8410
	SRCNN [27]	30.48 / 0.8628	27.50 / 0.7513	26.90 / 0.7103	24.52 / 0.7226	27.58 / 0.8555
	VDSR [13]	31.35 / 0.8838	28.02 / 0.7678	27.29 / 0.7252	25.18 / 0.7525	28.88 / 0.8854
	DRCN [21]	31.53 / 0.8854	28.03 / 0.7673	27.24 / 0.7233	25.14 / 0.7511	28.98 / 0.8870
	LapSRN [24]	31.52 / 0.8854	28.08 / 0.7687	27.31 / 0.7255	25.21 / 0.7545	29.08 / 0.8883
	DRRN_B1U25 [14]	31.68 / 0.8888	28.21 / 0.7721	27.38 / 0.7284	**25.44** / 0.7638	29.44 / 0.8941
	MemNet [16]	31.74 / 0.8893	**28.26** / 0.7723	27.40 / 0.7281	***25.50*** / 0.7630	***29.64*** / ***0.8967***
	TSCN [25]	***31.82*** / ***0.8907***	***28.28*** / **0.7734**	***27.42*** / ***0.7301***	**25.44** / **0.7644**	29.48 / 0.8954
	ms-LapSRN [26]	31.72 / 0.8891	28.25 / 0.7730	***27.42*** / 0.7296	***25.50*** / ***0.7661***	**29.53** / **0.8956**
	EA-IDRN (ours)	**31.76** / **0.8903**	**28.26** / ***0.7735***	**27.41** / **0.7300**	25.42 / 0.7635	29.44 / 0.8944

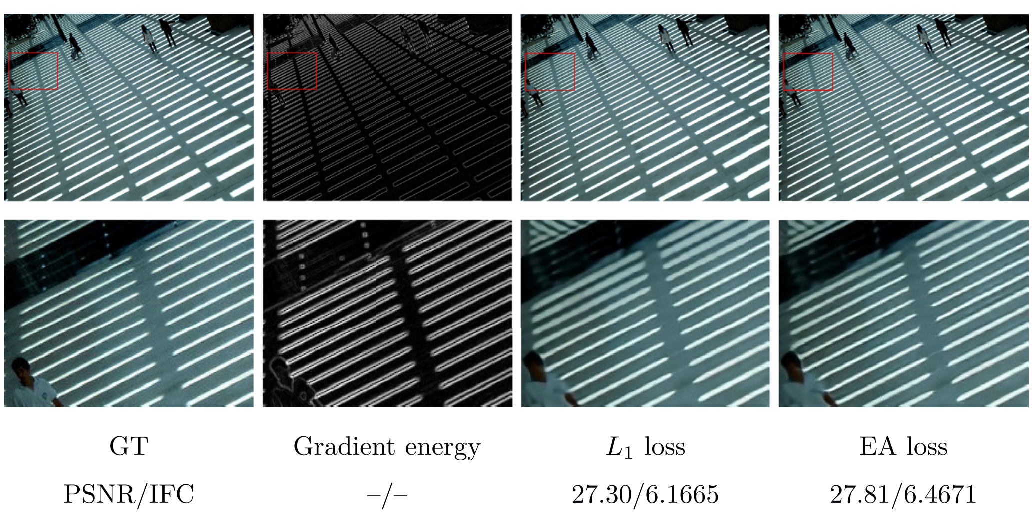

Fig. 7. Visualization of the computed gradient energy and the comparative SISR results using L_1 loss and our proposed EA loss.

SelfExSR [43]) and eight deep-learning-based methods (SRCNN [11,27], VDSR [13], DRCN [21], LapSRN [24], DRRN [14], MemNet [16], TSCN [25], and ms-LapSRN [26]) are considered. Source codes or pre-trained models of these methods are publicly available. For fair comparison, SISR models trained using very large datasets[1] (e.g., [15,20]) or high-resolution datasets[2] (e.g., [45,47,48]) are not considered in this part.

We show quantitative evaluation results on five testing datasets (Set5 [42], Set14 [60], B100 [59], Urban100 [43], and Manga109 [61]) with the scale factors ×2, ×3, ×4 in Table 5 (PSNR and

[1] ImageNet dataset or its subset [51].

[2] DIV2K dataset [49,50].

Table 6
Average IFC values with the scale factors $\times 2$, $\times 3$ and $\times 4$ on Set5, Set14, B100, Urban100 and Manga109 datasets. ***Red*** and **blue** indicate the best and the second best performance, respectively. It is noted that the IFC value is calculated on Y channel.

Dataset	Scale	Bicubic	VDSR [13]	LapSRN [24]	DRRN [14]	MemNet [16]	TSCN [25]	ms-LapSRN [26]	EA-IDRN
	$\times 2$	6.083	8.580	8.401	8.670	8.850	**9.175**	8.628	***9.314***
Set5	$\times 3$	3.580	5.203	–	5.394	5.503	**5.544**	–	***5.679***
	$\times 4$	2.329	3.542	3.515	3.700	**3.787**	3.766	3.697	***3.899***
	$\times 2$	6.105	8.159	8.042	8.280	8.469	**8.729**	8.236	***8.902***
Set14	$\times 3$	3.473	4.691	–	4.870	4.958	**4.970**	–	***5.090***
	$\times 4$	2.237	3.106	3.089	3.249	**3.309**	3.286	3.202	***3.382***
	$\times 2$	5.619	7.494	7.295	7.513	7.665	**7.871**	7.475	***7.982***
B100	$\times 3$	3.138	4.151	–	4.235	4.300	**4.350**	–	***4.423***
	$\times 4$	1.978	2.679	2.618	2.746	2.778	**2.792**	2.692	***2.868***
	$\times 2$	6.245	8.629	8.441	8.889	9.122	**9.442**	8.881	***9.706***
Urban100	$\times 3$	3.620	5.159	–	5.440	**5.560**	5.559	–	***5.703***
	$\times 4$	2.361	3.462	3.448	3.669	**3.786**	3.715	3.641	***3.833***
	$\times 2$	6.230	8.886	8.912	9.212	9.470	**9.976**	9.115	***10.220***
Manga109	$\times 3$	3.522	5.310	–	5.681	5.798	**5.846**	–	***6.004***
	$\times 4$	2.283	3.633	3.661	3.903	**4.038**	3.967	3.854	***4.096***

Table 7
Average running time for scale factors $\times 4$ on three common resolution settings including 480×360, 640×480 and 1280×720. ***Red*** and **blue** indicate the best and the second best performance, respectively.

Dataset	Resolution	VDSR [13]	LapSRN [24]	DRRN [14]	MemNet [16]	TSCN [25]	ms-LapSRN [26]	EA-IDRN
Time (s)	480×360	0.0413	**0.0278**	5.0254	8.2985	0.0349	0.0538	***0.0080***
	640×480	0.0859	**0.0443**	10.9856	15.0263	0.0668	0.0827	***0.0118***
	1280×720	0.2789	**0.1188**	25.0669	35.2654	0.1528	0.2270	***0.0307***

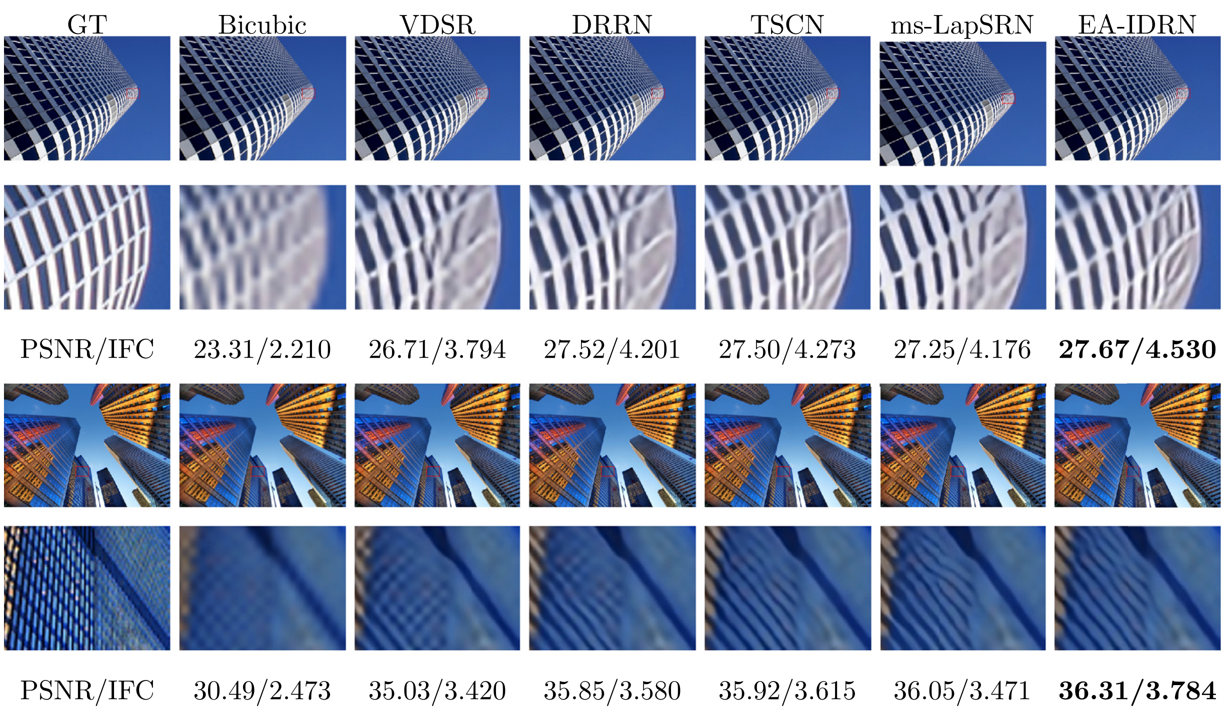

Fig. 8. Qualitative comparisons. Top: image "img005" from Urban100 dataset with scale factor $\times 4$. Bottom: image "img012" from Urban100 dataset with scale factor $\times 4$. Please zoom in on screen for better visualization.

SSIM indexes) and Table 6 (IFC index). In many cases, our proposed light-weight EA-IDRN model (23 layers in total) achieves even higher PSNR and SSIM values than some very deep networks (depth $>$ 50), such as the 52-layer DRRN [14] and 80-layer MemNet [16]. It is worth mentioning that our EA-IDRN model achieves the new state-of-the-art results based on the IFC index, which correlates better with human perception [63], for all five testing datasets with 3 different scale factors ($\times 2$, $\times 3$, and $\times 4$).

In Table 7, we show the averaged running time of different SISR methods to process 100 input images of three different resolutions including 480×360, 640×480 and 1280×720. The testing are conducted on a PC which is equipped with NVIDIA GTX 1080Ti GPU (11 GB memory). It is noted that our proposed EA-IDRN runs much faster than other state-of-the-art SISR methods[3] and can still achieve real-time speed ($>$30 fps) on processing 1280×720 images.

Some comparative results with state-of-the-art deep-learning-based SISR methods are shown in Fig. 8. Overall our EA-IDRN model can achieve better image restoration results. It is observed

[3] Caffe-implemented DRRN [14] and MemNet [16] are quite time-consuming. The main reason behind this phenomenon is due to the limitation of GPU memory, these two methods need to divide the testing image into small patches and test on these patches, then collect all the time together.

Table 8
Detailed Comparisons with EDSR and RDN. Depth indicates the number of convolution and deconvolution layers (only layers with kernel size larger than 1×1 are counted). The running times are tested on 1280×720 resolution images with scale $\times 4$ by a PC equipped with NVIDIA GTX 1080Ti (11GB memory), Cuda 8.0 and Cudnn 5.1.

Models	EA-IDRN		EDSR	RDN
Training Data	RGB91+B200	DIV2K	DIV2K	DIV2K
Depth	23	23	69	133
Set5 $\times 2$	37.91 dB	37.98 dB	38.11 dB	38.24 dB
Set14 $\times 2$	33.32 dB	33.52 dB	33.92 dB	34.01 dB
B100 $\times 2$	32.12 dB	32.17 dB	32.32 dB	32.34 dB
Time (s)	0.0307	0.0307	10.0579	6.3548

that parallel lines and lattice texture pattern in red highlighted region processed by our EA-IDRN method are much sharper and clearer than the results of other SISR methods. Moreover, our method can effectively suppress undesired artifacts or distortions which are helpful for other high-level image processing tasks such as target detection and medical analysis.

It is possible to make use of the high-resolution DIV2K dataset (containing 800 training images of 2K resolution) to train our EA-IDRN model. The comparative evaluation results (in terms of accuracy and computational speed) with EDSR [45] and RDN [47] are shown in Table 8. It is noted that using a larger or higher resolution training dataset (e.g., DIV2K or ImageNet) can generally lead to higher SISR accuracy. For instance, our EA-IDRN model trained using the high-resolution DIV2K dataset achieves higher PSNR than the one trained using low-resolution images from RGB91 and BSD200 (37.98 dB vs. 37.91 dB on Set5 with scale $\times 2$). The results indicate that the performance of EA-IDRN model can be further boosted using a higher quality training set (e.g., DIV2K or ImageNet). However, the improvement is not significant. The underlying principle is that our EA-IDRN is a 23-layer model and using a very large training dataset could not significantly boost the performance of this light-weight network. Our experimental results are consistent with the ones reported in [27]. We could stack more modules, increase the filter number of convolution layer (more parameters) and use a larger training dataset to further boost the performance, although it is beyond the scope of this paper. It is worth mentioning that our proposed light-weight EA-IDRN model run significantly faster than EDSR and RDN (EDSR vs. RDN vs. EA-IDRN: 10.0579s vs. 6.3548s vs. 0.0307s), which is more suitable to facilitate image pre-processing tasks.

5. Conclusion

In this paper, a compact but powerful Energy-Aware Improved Deep Residual Network (EA-IDRN) model is proposed for fast and accuracy SISR. We present a number of simple but effective structure modifications to two basic building blocks (residual blocks and skip connections) in deep residual networks. In addition, we propose a novel energy-aware training loss to adaptively adjust how high-frequency and low-frequency image regions are restored. Gradient energy information is utilized to emphasize the restoration of regions with textures or edges to achieve higher SISR accuracy. It is worth mentioning that these improvements can significantly increase SISR accuracy while causing no/ignorable extra computational loads. Extensive experimental results on multiple benchmark datasets demonstrate that our EA-IDRN method achieves more accurate results with faster speed.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 51575486, 51605428, and U1664264).

References

[1] L. Zhang, H. Zhang, H. Shen, P. Li, A super-resolution reconstruction algorithm for surveillance images, Signal Process. 90 (3) (2010) 848–859, doi:10.1016/j.sigpro.2009.09.002.
[2] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[3] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image Super-Resolution Via Sparse Representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[4] R. Timofte, V. De, L. Van Gool, Anchored neighborhood regression for fast example-based super-resolution, in: IEEE Int. Conf. Comput. Vis., 2013, pp. 1920–1927.
[5] R. Timofte, V.D. Smet, L.V. Gool, A+: adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conf. Comput. Vis., 2014.
[6] S. Schulter, C. Leistner, H. Bischof, Fast and accurate image upscaling with super-resolution forests, in: Computer Vision and Pattern Recognition, 2015, pp. 3791–3799.
[7] P. Chen, X. Xu, C. Deng, Deep view-aware metric learning for person re-identification, in: International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 620–626.
[8] J. Xu, L. Luo, C. Deng, H. Huang, Bilevel distance metric learning for robust image recognition, in: Neural Information Processing Systems (NIPS), 2018, pp. 1–10.
[9] Z. He, Y. Cao, Y. Dong, J. Yang, Y. Cao, C.-l. Tisse, Single image based non-uniformity correction of uncooled long-wave infrared detectors: a deep learning approach, Appl. Opt. 57 (18) (2018) D155–D164.
[10] Z. He, S. Tang, J. Yang, Y. Cao, M.Y. Yang, Y. Cao, Cascaded deep networks with multiple receptive fields for infrared image super-resolution, IEEE Trans. Circuits Syst. Video Technol. (2018) Accepted.
[11] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: IEEE Eur. Conf. Comput. Vis., 2014, pp. 184–199, doi:10.1007/978-3-319-10593-2_13.
[12] H. Chang, D.Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2004, pp. 275–282.
[13] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1646–1654. arXiv: 1511.04587, doi: 10.1109/TPAMI.2015.2439281.
[14] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3147–3155, doi:10.1109/CVPR.2017.298.
[15] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: ICCV, 2017, pp. 4799–4807.
[16] Y. Tai, J. Yang, X. Liu, C. Xu, MemNet: a persistent memory network for image restoration, in: ICCV, 2017, pp. 4539–4547. arXiv: 1708.02209.
[17] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: IEEE Eur. Conf. Comput. Vis., 2016. arXiv: 1608.00367, doi: 10.1007/978-3-319-46448-0.
[18] W. Shi, J. Caballero, F. Huszar, J. Totz, A.P. Aitken1, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1874–1883, doi:10.1109/CVPR.2016.207.
[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778. arXiv: 1512.03385, doi: 10.3389/fpsyg.2013.00124.
[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 4681–4690. arXiv: 1609.04802.
[21] J. Kim, J.K. Lee, K.M. Lee, Deeply-recursive convolutional network for image super-resolution, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1637–1645. arXiv: 1511.04491, doi: 10.1109/CVPR.2016.181.
[22] X.-J. Mao, C. Shen, Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, NIPS, 2016.
[23] L. Zhang, L. Zhang, X. Mou, D. Zhang, A comprehensive evaluation of full reference image quality assessment algorithms, ICIP, 2012, doi:10.1109/ICIP.2012.6467150.
[24] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep Laplacian pyramid networks for fast and accurate super-resolution, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 624–632, doi:10.1109/CVPR.2017.618.
[25] Z. Hui, X. Wang, X. Gao, Two-stage convolutional network for image super-resolution, in: ICPR, 2018, pp. 2670–2675.
[26] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Fast and accurate image super-resolution with deep laplacian pyramid networks, IEEE Trans. Pattern Analy. Machine Intell. (2018) Accepted.
[27] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2015) 295–307. arXiv: 1501.00092.
[28] R. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans. Acoust. Speech Signal Process. 29 (6) (1981) 1153–1160.

[29] C.E. Duchon, Lanczos Filtering in One and Two Dimensions, J. Appl. Meteorol. 18 (8) (1979) 1016–1022.
[30] J. Sun, J. Sun, Z. Xu, H.Y. Shum, Image super-resolution using gradient profile prior, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2008, doi:10.1109/CVPR.2008.4587659.
[31] K. Zhang, X. Gao, D. Tao, X. Li, Single image super-resolution with non-local means and steering kernel regression., IEEE Trans. Image Process. 21 (11) (2012) 4544–4556. arXiv: 1211.0290, doi: 10.1109/TIP.2012.2208977.
[32] R. Timofte, R. Rothe, L.V. Gool, Seven ways to improve example-based single image super resolution, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1865–1873.
[33] C. Deng, J. Xu, K. Zhang, D. Tao, X. Gao, X. Li, Similarity constraints-based structured output regression machine : an approach to image super-resolution, IEEE Trans. Neural Netw. Learn.Syst. 27 (12) (2016) 2472–2485, doi:10.1109/TNNLS.2015.2468069.
[34] K. Zhang, J. Li, H. Wang, X. Liu, X. Gao, Learning local dictionaries and similarity structures for single image super-resolution, Signal Process. 142 (2018) 231–243, doi:10.1016/j.sigpro.2017.07.020.
[35] X. Fan, Y. Yang, C. Deng, J. Xu, X. Gao, Compressed multi-scale feature fusion network for single image super-resolution, Signal Process. 146 (2018) 50–60, doi:10.1016/j.sigpro.2017.12.017.
[36] H.S. Hou, H.C. Andrews, Cubic splines for image interpolation and digital filtering, IEEE Trans. Acoust. Speech. Signal Process. 26 (6) (1978) 508–517, doi:10.1109/TASSP.1978.1163154.
[37] X. Li, M.T. Orchard, New edge-directed interpolation, IEEE Trans. Image Process. 10 (10) (2001) 1521–1527, doi:10.1109/83.951537.
[38] S. Baker, T. Kanade, Limits on super-resolution and how to break them, IEEE Trans. Pattern Anal. Mach.Intell. 24 (9) (2002) 1167–1183.
[39] Z. Lin, H.Y. Shum, Fundamental limits of reconstruction-based superresolution algorithms under local translation, IEEE Trans. Pattern Anal. Mach.Intell. 26 (1) (2004) 83–97.
[40] W.T. Freeman, E.C. Pasztor, T. Owen, Y. Carmichael, Learning low-level vision, Int. J. Comput. Vision 40 (2000) 1–43.
[41] W.T. Freeman, T.R. Jones, E.C. Pasztor, Example-based super-resolution, IEEE Comput. Gr. Appl. 22 (March) (2002) 56–65.
[42] M. Bevilacqua, A. Roumy, C. Guillemot, A. Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Br. Mach. Vis. Conf., 2012.
[43] J.B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2015.
[44] G. Freedman, R. Fattal, Image and video upscaling from local self-examples, Acm Trans. Gr. 30 (2) (2011) 12.
[45] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, Enhanced deep residual networks for single image super-resolution, CVPR workshop, 2017. arXiv: 1707.02921, doi: 10.1109/CVPRW.2017.151.
[46] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten, Densely connected convolutional networks, in: CVPR, 2017, pp. 4700–4708. arXiv: 1608.06993, doi: 10.1109/CVPR.2017.243.
[47] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018. arXiv: 1802.08797.
[48] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: ECCV, 2018, p. InPress. arXiv: 1807.02758.
[49] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, et al., Ntire 2017 challenge on single image super-resolution: methods and results, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.
[50] E. Agustsson, R. Timofte, NTIRE 2017 challenge on single image super-resolution: dataset and study, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2017, doi:10.1109/CVPRW.2017.150.
[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vision 115 (3) (2015) 211–252, doi:10.1007/s11263-015-0816-y.
[52] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in: Advances in Neural Information Processing Systems, 2016, pp. 658–666.
[53] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: IEEE Eur. Conf. Comput. Vis., 2016. arXiv: 1603.08155, doi: 10.1007/978-3-319-46475-6_43.
[54] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, S. Yan, Deep edge guided recurrent residual learning for image super-resolution, IEEE Trans. Image Process. 26 (12) (2017) 5895–5907.
[55] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2528–2535. arXiv: 1302.1700, doi: 10.1109/CVPR.2010.5539957.
[56] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, 2016, pp. 630–645.
[57] C.R.R. Molina, O.P. Vila, Solving internal covariate shift in deep learning with linked neurons, arXiv:1712.02609 (2017).
[58] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for image restoration with neural networks, IEEE Trans. Comput. Imaging 3 (1) (2017) 47–57. arXiv: 1511.08861, doi: 10.1109/TCI.2016.2644865.
[59] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, ICCV, 2001.
[60] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: International Conference on Curves and Surfaces, 2010, pp. 711–730.
[61] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, K. Aizawa, Manga109 dataset and creation of metadata, in: Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding, ACM, 2016, p. 2.
[62] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, doi:10.1109/TIP.2003.819861.
[63] H.R. Sheikh, A.C. Bovik, G. de Veciana, An information fidelity criterion for image quality assessment using natural scene statistics., IEEE Trans.Image Process. 14 (12) (2005) 2117–2128, doi:10.1109/TIP.2005.859389.
[64] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, arXiv:1408.5093 (2014).
[65] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: Int. Conf. Learn. Represent., 2015. arXiv: 1412.6980.
[66] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: IEEE Int. Conf. Comput. Vis., 2015. arXiv: 1502.01852, doi: 10.1109/ICCV.2015.123.
[67] Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 723–731. arXiv: 1803.09454v1.