A Survey of Image Super Resolution Based on CNN

Qianxiong Xu and Yu Zheng(✉)

School of Computer and Software, Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China
yzheng@nuist.edu.cn

Abstract. With the advent of the information age in contemporary society, images are everywhere, no matter in military use or in daily life. Therefore, as a medium for people to obtain information, images have become more and more important. With the fast development of deep convolution neural networks (DCNNs), Single-Image Super-Resolution (SISR) becomes one of the techniques that have made great breakthroughs in recent years. In this paper, we give a brief survey on the task of SISR. In general, we introduce the SR problem, some recent SR methods, public benchmark datasets and evaluation metrics. Finally, we conclude by denoting some points that could be further improved in the future.

Keywords: Single-image · Super-resolution · Deep learning · DCNNs

1 Introduction

As vital image processing class of image processing techniques in image processing and computer vision, Single-Image Super-Resolution (SISR), whose basic goal is to recover high-resolution (HR) images from low-resolution (LR) images, plays an important role in our daily lives. It could be applied to various types of applications, for example, surveillance and security [1–3], video noise removing, medical [4–6] and etc. Besides, it could provide help to other computer vision tasks, because we could make a better dataset with higher quality images. Generally speaking, SISR is quite challenging for there is a one-to-many mapping between the LR images and HR images.

As deep convolution neural networks (DCNNs) appear to be able to handle tasks related to images well, super-resolution (SR) models that based on deep network architectures have been explored and often result in the state-of-the-art (sota) performance on various benchmarks. Date back to 2014, Dong et al. [7, 8] first introduced their model of SRCNN which combined Convolutional Neural Networks (CNN) with the task of SISR and made a huge breakthrough at that time. And as Goodfellow et al. [10] propose the Generative Adversarial Networks (GAN) which contains a theory of adversarial, some great methods, like SRGAN [9] introduce GAN into the field of SR and get satisfying results. In general, different SR algorithms differ from each other mainly in the following major aspects: different types of network architectures [11–13], loss functions [14–16], learning strategies [14, 17, 18], etc.

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 184–199, 2020.
https://doi.org/10.1007/978-3-030-48513-9_15

In this paper, a brief overview of recent methods of SISR with deep learning is presented. While most of the existing surveys focus on traditional methods, our survey will mainly focus on deep learning methods.

Our survey has the following contributions:

1) We give a brief review of SISR techniques based on deep learning, including problem definitions, benchmark datasets, evaluation metrics, etc.
2) We provide a general overview of recent methods with techniques that are based on deep learning hierarchically and explore the pros and cons of each component for an effective SR method.
3) We analyze the challenges and future directions to provide an insightful guidance.

In the following sections, we will cover various aspects of recent advances in SISR with deep learning. Section 2 gives the problem definition, reviews the benchmark datasets and introduce some common evaluation metrics. Section 3 analyzes main components of supervised SR. Section 4 gives an introduction to our experiments and Sect. 5 expresses conclusions and discusses future directions.

2 Problem, Datasets and Evaluation Metrics

2.1 Problem Definitions

Single-Image Super-Resolution problem aims at recovering a high-resolution (HR) image from a single low-resolution (LR) image effectively. We could model the process of the acquisition of LR image I_{LR} with the degradation process as follows:

$$I_{LR} = \mathcal{D}(I_{HR}; \theta_{\mathcal{D}}) \tag{1}$$

Where I_{LR} represents the LR image, $\mathcal{D}$ is a degradation mapping function, I_{HR} denotes the corresponding ground-truth HR image and $\theta_{\mathcal{D}}$ corresponds to the parameters of the degradation process, like some noise factors or scaling factors. The degradation process is quite simple, however, in most situations, the details of the degradation process is unknown and only LR images are provided. Therefore, a requirement of recovering a HR image $\hat{I}_{HR}$ from the provided LR image I_{LR} is raised up, so that $\hat{I}_{HR}$ should be identical to the ground-truth HR image I_{HR} by the following formula:

$$\hat{I}_{HR} = \mathcal{F}(I_{LR}; \theta_{\mathcal{F}}) \tag{2}$$

Where $\mathcal{F}$ is the SR model and $\theta_{\mathcal{F}}$ denotes the parameters of the model.

Most works directly model the degradation as a single downsampling operation as follows:

$$\mathcal{D}(I_{HR}; \theta_{\mathcal{D}}) = (I_{HR}) \downarrow_s, s \in \theta_{\mathcal{D}} \tag{3}$$

Where $\downarrow_s$ is a downsampling operation with the scaling factor s and bicubic interpolation with antialiasing is the most commonly used downsampling operation.

Finally, the objective of SR is given as follow:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}\left(\hat{I}_{HR}, I_{HR}\right) + \lambda\phi(\theta) \tag{4}$$

Where $\mathcal{L}\left(\hat{I}_{HR}, I_{HR}\right)$ represents the loss function between the generated HR image $\hat{I}_{HR}$ and the ground-truth image I_{HR}, $\phi(\theta)$ is the regularization term and λ is a trade-off parameter.

2.2 Datasets

Currently, there are some popular used benchmarks for testing the performance of SR models including Set5 [19], Set14 [20], BSD100 [21], Urban100 [22], DIV2K [23] and Manga109 [24]. More details of these datasets are presented in Table 1 and some images from these datasets are shown in Fig. 1.

Table 1. Public image datasets for SR benchmarks

Dataset	Amount	Format	Categories
Set5 [19]	5	PNG	Baby, bird, butterfly, head, woman
Set14 [20]	14	PNG	Humans, animals, insects, flowers, vegetables, comic, slides, etc.
BSD100 [21]	100	JPG	Animal, building, food, landscape, people, plant, etc.
Urban100 [22]	100	PNG	Architecture, city, structure, urban, etc.
Manga109 [23]	109	PNG	Manga volume
DIV2K [24]	1000	PNG	Environment, flora, fauna, handmade object, scenery, etc.

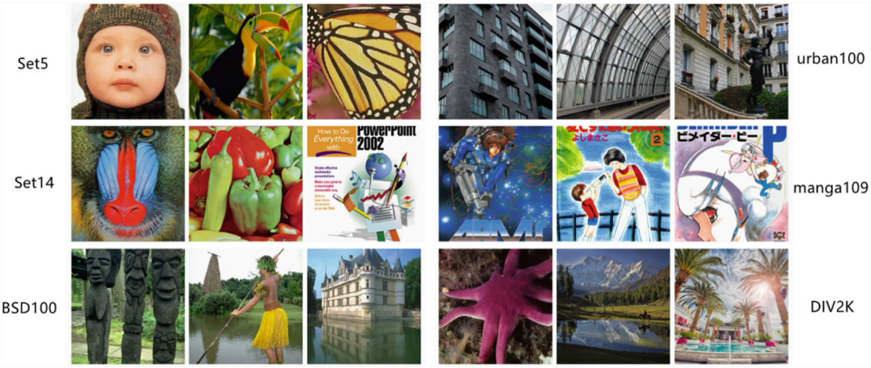

Fig. 1. Image samples from benchmark datasets

Set5 [19] is a classical dataset and only contains five test images of a baby, bird, butterfly, head, and a woman. Set14 [20] consists of more categories compared to Set5. However, the number of images are still low, with only 14 test images. BSD100 [21] is another classical dataset having 100 test images proposed by Martin et al. The dataset is composed of a large variety of images ranging from natural images to object-specific such as plants, people, food etc. Urban100 [22] is a relatively more recent dataset introduced by Huang et al. The number of images is the same as BSD100. However, the composition is entirely different. The focus of the photographs is on human-made structures, such as urban scenes. Manga109 [23] is the latest addition for evaluating super-resolution algorithms. The dataset is a collection of 109 test images of a manga volume. The manga was professionally drawn by Japanese artists and were available only for commercial use between the 1970s and 2010s. DIV2K [24] is a dataset used for NITRE challenge. The image quality is of 2K resolution and is composed of 800 images for training while 100 images each for testing and validation. As the test set is not publicly available, the results are only reported on validation images for all the algorithms.

2.3 Evaluation Metrics

In the task of SR, evaluation metrics are used to assess the quality of the recovered HR image, not only refers to the differences between the recovered pixel and the corresponded pixel in the ground-truth HR image, but also focus on the perceptual assessments of human viewers. In this section, we'll introduce two types of the most commonly used Evaluation metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM).

2.3.1 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is one of the most commonly used evaluation metrics in SR. Its main goal is to measure the reconstruction quality of lossy transformation. The MSE and the PSNR between the ground-truth image I and the generated image $\hat{I}$ are defined as follows:

$$\mathrm{MSE} = \frac{1}{HWC}\sum\nolimits_{i}^{W}\sum\nolimits_{j}^{H}\sum\nolimits_{k}^{C}\left(I^{i,j,k} - \hat{I}^{i,j,k}\right)^2 \tag{5}$$

$$\mathrm{PSNR} = 10 * log_{10}\left(\frac{L^2}{MSE}\right) \tag{6}$$

Where H, W, C denote the height, width and channels of the image, respectively, $I^{i,j,k}$ denotes the pixel in the ground-truth HR image with the coordinates of (i, j, k) in the dimensions of width, height and channels, respectively, $\hat{I}^{i,j,k}$ is defined similarly, L is the maximum possible pixel value(usually 255 for 8-bit image). As L is always fixed, MSE becomes the only factor influencing PSNR, only caring about the differences between the pixel values at the same positions instead of human visual perception. In this way, the generated image might be much better in the perspective of pixel values, but can't be considered as good by human visual systems (HVS). However, due to the necessity to compare performance with literature works and the lack of completely accurate perceptual metrics, PSNR is currently the most widely used evaluation metric for SR models.

2.3.2 Structural Similarity

HVS is more likely to extract the structural information from the viewing field [25], therefore, an evaluation metric named structural similarity index (SSIM) [26] is proposed to measure the structural similarity between images, and there are three relatively independent comparisons: luminance, contrast, and structure comparisons. For an image I with the shape H∗W∗C, its mean and the standard deviation value are given as follows:

$$\mu_I = \frac{1}{HWC}\sum\nolimits_i^W \sum\nolimits_j^H \sum\nolimits_k^C I^{i,j,k} \tag{7}$$

$$\sigma_I = (\frac{1}{HWC-1}\sum\nolimits_i^W \sum\nolimits_j^H \sum\nolimits_k^C (I^{i,j,k} - \mu_I)^2)^{\frac{1}{2}} \tag{8}$$

Where μ_I indicates the mean value of image I, σ_I denotes the standard deviation of the image intensity. And the comparison functions on luminance and contrast, denoted as $\mathcal{C}_l\left(I, \hat{I}\right)$ and $\mathcal{C}_c\left(I, \hat{I}\right)$, respectively, are given as follows:

$$\mathcal{C}_l\left(I, \hat{I}\right) = \frac{2\mu_I\mu_{\hat{I}} + C_1}{\mu_I^2 + \mu_{\hat{I}}^2 + C_1} \tag{9}$$

$$\mathcal{C}_c\left(I, \hat{I}\right) = \frac{2\sigma_I\sigma_{\hat{I}} + C_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2} \tag{10}$$

Where $C_1 = (k_1L)^2$ and $C_2 = (k_2L)^2$ are constants for avoiding instability, in which $k_1 \ll 1$ and $k_2 \ll 1$ are small constants.

The image structure is represented by the normalized pixel values (i.e., $\frac{I-\mu_I}{\sigma_I}$), whose correlations (i.e., inner product) measure the structural similarity. Then, the structure comparison function $\mathcal{C}_s\left(I, \hat{I}\right)$ is defined as follows:

$$\sigma_{I\hat{I}} = \frac{1}{HWC-1}\sum\nolimits_i^W \sum\nolimits_j^H \sum\nolimits_k^C (I^{i,j,k} - \mu_I)(\hat{I}^{i,j,k} - \mu_{\hat{I}}) \tag{11}$$

$$\mathcal{C}_s\left(I, \hat{I}\right) = \frac{\sigma_{I\hat{I}} + C_3}{\sigma_I\sigma_{\hat{I}} + C_3} \tag{12}$$

Where $\sigma_{I\hat{I}}$ is the covariance between I and $\hat{I}$, C_3 is a constant to assure stability.

At last, the formula of calculating SSIM is given by:

$$\mathrm{SSIM}\left(\mathrm{I}, \hat{I}\right) = \left[\mathcal{C}_l\left(I, \hat{I}\right)\right]^{\alpha}\left[\mathcal{C}_c\left(I, \hat{I}\right)\right]^{\beta}\left[\mathcal{C}_s\left(I, \hat{I}\right)\right]^{\gamma} \tag{13}$$

Where α, β, γ are constants for adjusting the relative importance. In practice, researcher often set $\alpha = \beta = \gamma = 1$ and $C_3 = \frac{C_2}{2}$, then SSIM is calculated as:

$$\mathrm{SSIM}\left(\mathrm{I}, \hat{I}\right) = \frac{\left(2\mu_I\mu_{\hat{I}} + C_1\right)\left(\sigma_{I\hat{I}} + C_2\right)}{\left(\mu_I^2 + \mu_{\hat{I}}^2 + C_1\right)\left(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2\right)} \tag{14}$$

As its aim shows, the SSIM evaluates the quality of the generated images from the perspective of the HVS, it better meets the requirements of perceptual assessment [27, 28] compared to PSNR. Therefore, it is also widely used by researchers.

3 Supervised Super-Resolution

Supervised SR models are trained with both LR images and the corresponding ground-truth HR images. The essential components of SR models include: model frameworks, upsampling methods, network architecture and strategies for learning. Therefore, although these models differ from each other greatly, they are all exactly a combination of the components above. In this section, we will focus on the basic components of SR models, analyze their pros and cons, and in Sect. 4, we will choose some classical models to do the experiments.

3.1 Super-Resolution Frameworks

A key problem of SISR is the way of performing upsampling. Although the network architectures of SR models vary greatly, they can be corresponded to four categories: pre-upsampling SR, post-upsampling SR, progressive upsampling SR and iterative up-and-down sampling SR, as Fig. 2 shows.

3.1.1 Pre-upsampling Super-Resolution

As Dong et al. [7, 8] show in their work SRCNN, they first introduce a straightforward method that upsamples the LR images using a traditional method, then refine them using DCNNs in an end-to-end way. This framework (Fig. 2a) is considered the pre-upsampling SR framework. More specifically, the network first uses traditional upsampling method, like bicubic interpolation, to upsample the LR images to coarse HR images, then DCNNs are applied to construct concrete details.

Since this framework does the upsampling with traditional algorithms first, CNNs only need to refine the coarse HR images, therefore, one of the advantages is that the learning difficulty is reduced. Then, this framework appears to be more flexible because it could take arbitrary images and scale factors as input and gives output with the same model [11]. The main differences between models with this framework are the design of the network model and the learning strategies. However, there are also some drawbacks. Firstly, the traditional upsampling methods like bicubic interpolation, would often cause something like noise, blurring, etc. Further, compared with models using other frameworks, the temporal and spatial cost is always much higher [29, 30].

3.1.2 Post-upsampling Super-Resolution

As using pre-upsampling framework would result in much efficiency cost, the post-upsampling framework is then proposed by researchers. Similar to the pre-upsampling framework and just as its name illustrates, the post-framework does the complex mappings in the low-dimensional space and after that, it performs a learnable upsampling at the end (Fig. 2b).

It's obvious that this framework cost less because the operations of convolutions are performed in the low-dimensional space and this could also provide with faster speed. As a result, this framework also occupies one position in the SR field [9, 16]. However, there are also some shortcomings. The first one is that the upsampling is only performed in one

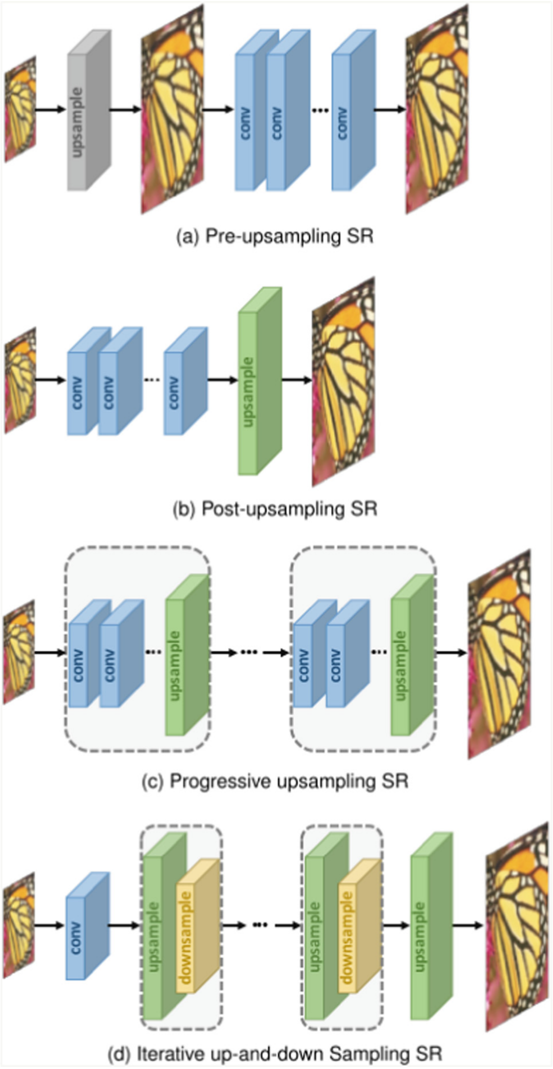

Fig. 2. SR model frameworks based on deep learning. The blue boxes indicate convolutional layers, the gray boxes represent predefined upsampling operations, the green and yellow boxes denote learnable upsampling and downsampling layers, respectively. (Color figure online)

step and this might fail to learn when the scale factors are large. Another disadvantage is that it lacks some flexibility, for it can't handle the work using a single model well when the scale factors vary.

Generally speaking, models using this type of framework differ to each other mainly in aspects of network design, learnable upsample layers and strategies for learning.

3.1.3 Progressive Upsampling Super-Resolution

To address the drawbacks of post-upsampling framework, a progressive upsampling SR framework is come into use (Fig. 2c). A typical example of this framework is the Laplacian pyramid SR network (LapSRN) [12]. It is based on a cascade of CNNs and progressively reconstruct the HR images. At each stage, the images are upsampled to higher resolution and refined by CNNs.

The main feature of progress upsampling framework is that it decomposes a difficult task into several simple tasks, models with this framework could both become much easier to learn to obtain better performance and could handle the conditions of different scale factors well without much extra cost. Furthermore, this kind of framework requires a multi-stage design, so, some strategies for learning can be further considered to reduce the learning difficulty to enhance the performance. However, one problem is that the network designing task is quite difficult and therefore, we need more guidance and instructions.

3.1.4 Iterative Up-and-Down Sampling Super-Resolution

Iterative up-and-down sampling framework is proposed to make the relationship between the LR and HR images pairs become more tightly. Back-projection is a new efficient iterative procedure within this framework, it is used to better refine the relationship between the LR and HR images [31]. Haris et al. [26] propose a deep back-projection network (DBPN) using blocks of connected upsampling and downsampling layers and reconstruct the final HR image by using the concatenation of all the reconstructed HR feature maps during the process of forward propagation. Combine with other techniques, like dense connections [32], DBPN became the champion algorithm in the competition of NTIRE 2018 [33].

The models under this framework can better mine the deep relationships between LR-HR image pairs and thus provide higher-quality reconstruction results. Nevertheless, the design criteria of the back-projection modules are still unclear. In fact, the back-projection units used in DBPN have a very complicated structure and require heavy manual design. Since this mechanism has just been introduced into super-resolution based on deep learning, the framework has great potential and needs further exploration.

3.2 Upsampling Methods

As the above section shows, there are mainly four frameworks to deal with the upsampling layers. Besides, it's also important to know how to implement the upsampling operations. Although there has already been various of traditional upsampling algorithms, like nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, etc. Using CNNs to learn upsampling operators has become more and more popular. In this section, we'll discuss about some classical interpolation-based algorithms and upsampling layers that are based on deep learning.

3.2.1 Interpolation-Based Upsampling

Traditional interpolation methods include nearest-neighbor, bilinear, bicubic interpolation and etc. Although upsampling layers that are based on deep learning perform quite well, some of the traditional interpolation-based upsampling methods are still in use in some networks.

Nearest-Neighbor Interpolation. The nearest-neighbor interpolation is simple. Its basic idea is to select the value of the nearest pixel for each interpolating position. On the one hand, this method is very fast to execute. On the other hand, it would usually produce blocky results.

Bilinear Interpolation. Just as the name denotes, bilinear interpolation would first do linear interpolation once on one axis, after that, it would do it again on another axis. Compared with nearest-neighbor interpolation, bilinear interpolation not only results in better performance, but also run fast.

Bicubic Interpolation. Similar to bilinear interpolation, the bicubic interpolation does cubic interpolation once on each of the two dimensions of the image. The bicubic interpolation could generate smoother results with fewer interpolation artefacts and lower speed compared to bilinear interpolation. As a matter of fact, the bicubic interpolation with anti-aliasing is now widely used to degrade the HR image to generate the corresponding LR image to make a dataset and is also widely accepted by researchers to use a pre-upsampling framework.

The interpolation-based upsampling methods don't provide any new information, just focus on its content and as a result, there would always be some side effects.

3.2.2 Learning-Based Upsampling

CNNs could handle the task of "understanding" the images well and therefore, researchers tried to use CNNs to force the network to understand the image and do a better upsampling. Two popular methods of learning-based upsampling are transposed convolution layer and sub-pixel layer.

Transposed Convolution Layer. While a normal convolutional operator with a stride greater than one, the output of the operation would result in smaller width and height, a transposed convolution layer, also known as deconvolution layer [34], behaves just as the opposite, it tries to get bigger width and height, so they could be used to do the upsampling task [26, 35]. More concretely, it could enlarge the resolution of images by inserting zero values and then doing convolution.

Although transposed convolution layer can be used in the field of SR to perform learnable upsampling, it could also cause "uneven overlapping" on each axis [36] and would easily generate chessboard-like patterns to reduce the SR performance.

Sub-pixel Layer. Another learnable upsampling layer is the sub-pixel layer [30], it performs upsampling by generating feature maps with the shape of $\mathrm{H} * \mathrm{W} * s^2C$ by convolution and then reshaping them into a shape of $\mathrm{sH} * \mathrm{sW} * \mathrm{C}$, where s is the upscale factor.

One of the advantages of using sub-pixel layer is that we could obtain larger receptive field, which could provide more contextual information to generate better HR images. However, blocky regions actually share the same receptive field and therefore it may result in some artefacts near the boundaries of different blocks.

These two learnable upsampling layers are widely used in post-upsampling framework and are always set in the final upsampling stage.

3.3 Network Architecture

The network design is currently one of the most important part in deep learning, researchers would always use some technologies, like residual learning, dense connection, etc. to improve their design.

3.3.1 Residual Learning

He et al. [37] propose a kind of DCNN named ResNet in 2016 and from then on, residual blocks are widely used in the design of networks. But before the proposal of ResNet, in the field of SR, researchers have already employed the technique of residual learning to their SR models. The residual learning can be roughly divided into two types: global residual learning and local residual learning.

Global Residual Learning. First of all, global residual learning is widely used especially in the pre-upsampling framework, because it only learns the residuals between the coarse HR image and the ground-truth HR image. Instead of learning the complicate information an image need, global residual learning only learns a residual map to restore the missing high-frequency details, and therefore reduce the learning difficulty and complexity.

Local Residual Learning. Both of the local residual learning and residual blocks in ResNet are used to improve the problem of degradation and gradient vanishing due to the learning difficulty caused by the network depths.

With the structure of shortcut connection and element-wise addition operations, global residual learning directly connects the input and output images, while local residual learning usually sets several this kind of structure between the layers.

3.3.2 Dense Connection

Huang et al. [32] come up with a network named DenseNet in CVPR 2017, the main components of this network are dense blocks, then more and more people use dense blocks to design their networks. Inside the dense blocks, the inputs consist of all former layers, which results in C_l^2 connections in a dense block with l layers. Similar to residual learning, the dense connections could effectively help avoid gradient vanishing, enhance signal propagation and encourage feature reuse. Besides it could also substantially reduce the number of parameters by reducing the number of channels in dense blocks and squeezing channels after concatenation.

Dense connections are widely used, some famous networks like, ESRGAN [38] and DBPN [26], adopt dense connections and get good results.

3.4 Strategies for Learning

There are various of strategies that are useful to promote the performance, like running time, the quality of the generated HR images, etc. The most commonly used strategies can be roughly divided into three categories: the loss functions, batch normalization and others.

3.4.1 Loss Functions

In the area of SR, loss functions are used to measure the difference between ground-truth HR images and the generated HR images it could help to optimize the model greatly. In the early stage of this task, researchers usually used pixel-wise L2 loss for optimization, but found it couldn't measure the reconstruction quality well. Since then, many other loss functions showed up for solving the problem. Despite of the loss functions combined with GAN, e.g., adversarial loss, cycle consistency loss, etc., there are four commonly used loss functions: Pixel Loss, Content Loss, Texture Loss and Total Variation Loss. The formulas are shown in Table 2.

Table 2. Common loss functions

Loss function	Formula
Pixel loss	$\mathcal{L}_{pixel\_l1}\left(I,\hat{I}\right)=\frac{1}{HWC}\sum_{i}^{W}\sum_{j}^{H}\sum_{k}^{C}\left\lvert\hat{I}^{i,j,k}-I^{i,j,k}\right\rvert$ $\mathcal{L}_{pixel_{l2}}\left(I,\hat{I}\right)=\frac{1}{HWC}\sum_{i}^{W}\sum_{j}^{H}\sum_{k}^{C}\left(\hat{I}^{i,j,k}-I^{i,j,k}\right)^{2}$
Content loss	$\mathcal{L}_{content}\left(I,\hat{I};\phi,l\right)=\frac{1}{H_{l}W_{l}C_{l}}\left\{\sum_{i}^{W_{l}}\sum_{j}^{H_{l}}\sum_{k}^{C_{l}}\left[\phi_{(l)}^{i,j,k}\left(\hat{I}\right)-\phi_{(l)}^{i,j,k}(I)\right]^{2}\right\}^{\frac{1}{2}}$
Texture loss	$G_{(l)}^{ij}(\mathrm{I})=\mathrm{vec}\left(\phi_{(l)}^{i}(\mathrm{I})\right)\cdot\mathrm{vec}\left(\phi_{(l)}^{j}(\mathrm{I})\right)$ $\mathcal{L}_{texture}\left(I,\hat{I};\phi,l\right)=\frac{1}{c_{l}^{2}}\left\{\sum_{i}^{W}\sum_{j}^{H}\left[G_{(l)}^{i,j}\left(\hat{I}\right)-G_{(l)}^{i,j}(I)\right]^{2}\right\}^{\frac{1}{2}}$
Total variation loss	$\mathcal{L}_{TV}\left(\hat{I}\right)=$ $\frac{1}{HWC}\sum_{i}^{W}\sum_{j}^{H}\sum_{k}^{C}\left[\left(\hat{I}^{i,j+1,k}-\hat{I}^{i,j,k}\right)^{2}+\left(\hat{I}^{i+1,j,k}-\hat{I}^{i,j,k}\right)^{2}\right]^{\frac{1}{2}}$

Pixel Loss. Pixel loss is used to measure the pixel-wise difference between I and $\hat{I}$ which includes L1 loss (mean absolute error) and L2 loss (mean square error). By using pixel loss as the loss function, it could guide the network to generate $\hat{I}$ to be close to the ground-truth I. L1 loss tends to have better performance and convergence compared to L2 loss [13, 17]. As the definition of PSNR illustrates, it is highly correlated with pixel-wise difference and minimizing pixel loss could directly maximize PSNR, the pixel loss appears to be the most popular choice. But the generated image would often lack high-frequency details and result in perceptually unpleasant results with over smooth textures [9, 15].

Content Loss. In order to solve the perceptual problem in pixel loss, the content loss is then introduced into SR [15]. By extracting feature maps by using a pre-trained image classification network, it could measure the semantic differences between images. Denote this pre-trained network as ϕ and the extracted feature maps on l^{th} layer as $\phi_{(l)}(I)$, the content loss is the Euclidean distance between high-level representations between two images. Content loss encourages the output image $\hat{I}$ to be perceptually similar to the ground-truth image I instead of forcing them to match pixels exactly.

Texture Loss. Inspired by Gatys et al. [39], the style of an image is considered to be an important factor influencing the quality of the generated image, Gram matrix $G_{(l)} \in \mathcal{R}^{c_l * c_l}$ is then introduced into SR, where $G_{(l)}^{ij}$ is the inner product between the vectorized feature maps i and j on layer l.

Total Variation Loss. The total variation loss is introduced into the SR field by Aly et al. [40] to suppress the noise. The sum of the absolute differences between neighboring pixels consists of the total variation loss and it could measure the amount of noise is in the image.

3.4.2 Batch Normalization

Batch normalization (BN) is proposed by Sergey et al. [41] to reduce internal covariate shift of networks. BN enables us to use much higher learning rates and initialization is not a big problem any more. BN results in accelerating the speed of convergence and improve the accuracy, therefore, it is widely adopted by researchers. However, Lim et al. [17] argue that using BN would lose the scale information of each image and the range flexibility. There is a trade-off whether using BN or not.

4 The Experiment Result and Performance Analysis

In this section, we mainly focus on the experiments, we select some of the classical models and some of the benchmark datasets, then apply PSNR and SSIM on them to make some comparisons. For SR models, we choose bicubic, SRCNN, EDSR, SRGAN and ESRGAN to recover HR images with a scale factor of 4. And use all benchmark datasets above to evaluate the PSNR and SSIM values of these models.

We evaluate each SR algorithm selected on the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) on the benchmark datasets in Sect. 2.2. Table 3 presents the results for 4x for the SR algorithms. In Fig. 3, we present the visual comparison between the selected SR algorithms and Fig. 4 shows a detailed comparison between a pair of images of the ground truth image and an image recovered by ESRGAN.

Table 3. Evaluation on PSNR and SSIM on recovered images with a upscale factor of 4

Methods	Set5		Set14		BSD100		Urban100		Manga109		DIV2K	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	28.43	0.8109	26.00	0.7023	25.96	0.6678	23.14	0.6574	25.15	0.789	28.11	0.775
SRCNN	30.48	0.8628	27.50	0.7513	26.90	0.7103	24.52	0.7226	27.66	0.858	29.33	0.809
EDSR	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.914	29.25	0.9017
SRGAN	32.05	0.8910	28.53	0.7804	27.57	0.7351	26.07	0.7839	–	–	28.92	0.896
ESRGAN	32.73	0.9011	28.99	0.7917	27.85	0.7455	27.03	0.8153	31.66	0.9196	–	–

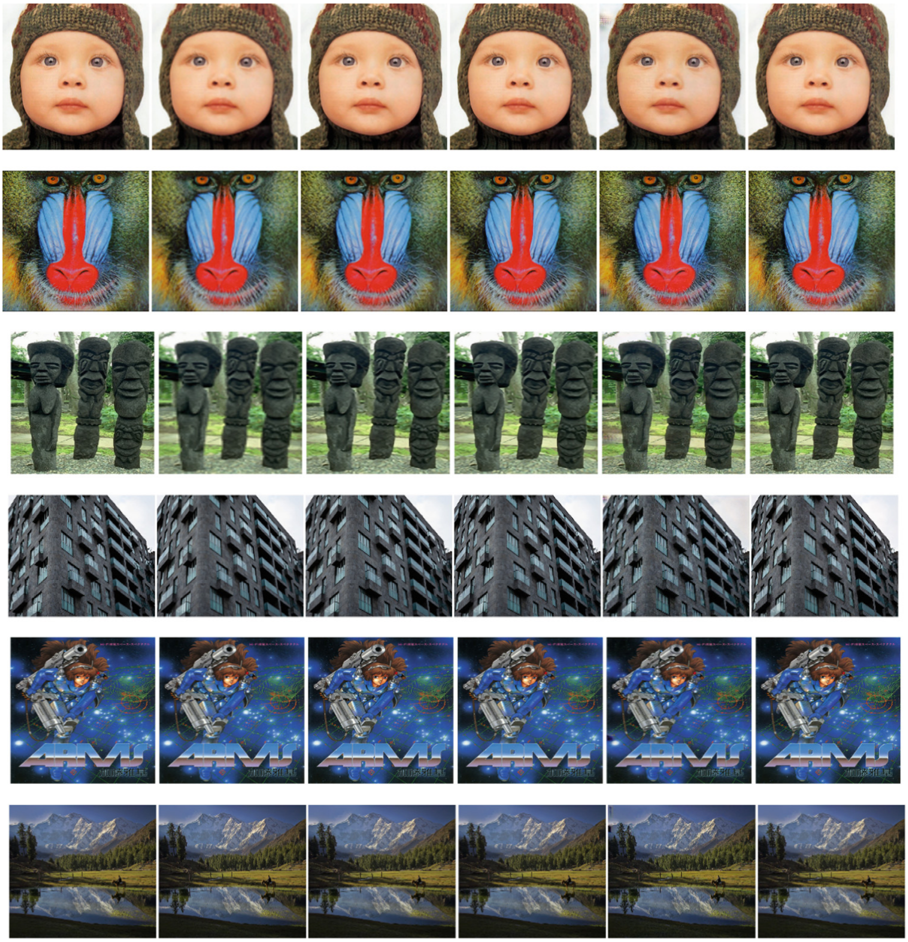

Fig. 3. Comparison between images, the images of each column are the ground-truth image, image recovered by bicubic, image recovered by SRCNN, image recovered by EDSR, image recovered by SRGAN and image recovered by ESRGAN, respectively.

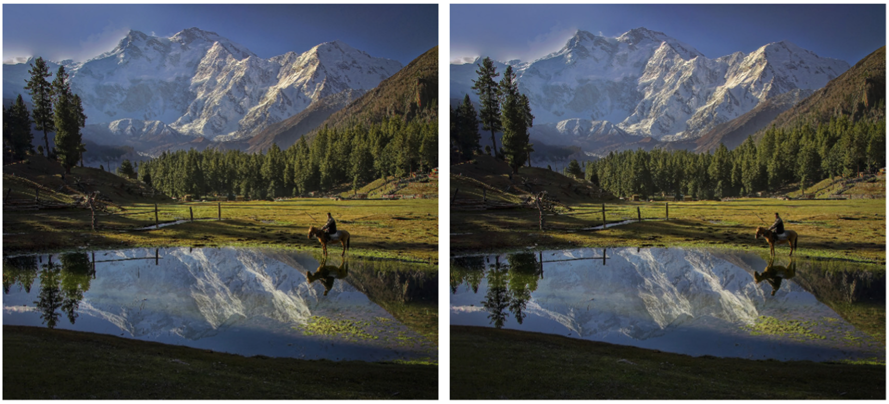

Fig. 4. 0002.jpg of DIV2K, the left one is the ground-truth image and the right one is the image recovered by ESRGAN.

5 Conclusion

SISR methods based on deep learning have achieved great success recently. In this paper, we give a brief survey on recent SISR methods and mainly discussed the improvement of supervised SR methods. However, there still exists something that we could improve to get a better result and, in this section, we will talk about this.

Network Design. Current sota SR methods tend to mainly focus on the final results of the recovered HR images while ignoring the complexity of their models and result in low inference speed. With a high-performance GPU, i.e. Titan GTX, current SR methods would take over 10 s for 4x SR per image of DIV2K, which is unacceptable in daily usage, therefore, we need to come up with some lightweight architectures to improve this problem. In addition, an important component of SR is the upsampling layers, the current upsampling methods, i.e. interpolation-based methods would result in expensive computation and couldn't be end-to-end learned, the transposed convolution would probably cause checkerboard artefacts. So, improving the upsampling methods could probably improve the recovering effects and inference time.

Learning Strategies. Loss function plays a critical part in the training of SR models which would build up constraints among LR and HR images and guide the network to optimize. In practice, some loss functions like L1 loss, L2 loss, perceptual loss are widely used. However, if there is any better loss function for SR is still unclear. Another factor is normalization, current sota SR methods prefer not to use normalization for some side effects, so other effective normalization techniques should be studied.

References

1. Lin, F., Fookes, C., Chandran, V., Sridharan, S.: Super-resolved faces for improved face recognition from surveillance video. In: Lee, S.-W., Li, Stan Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 1–10. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74549-5_1

2. Zhang, L., Zhang, H., Shen, H., Li, P.: A super-resolution reconstruction algorithm for surveillance images. Sig. Process. **90**, 848–859 (2010)
3. Rasti, P., Uiboupin, T., Escalera, S., Anbarjafari, G.: Convolutional neural network super resolution for face recognition in surveillance monitoring. In: Perales, F.J.J., Kittler, J. (eds.) AMDO 2016. LNCS, vol. 9756, pp. 175–184. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41778-3_18
4. Greenspan, H.: Super-resolution in medical imaging. Comput. J. **52**, 43–63 (2008)
5. Isaac, J.S., Kulkarni, R.: Super resolution techniques for medical image processing. In: ICTSD (2015)
6. Huang, Y., Shao, L., Frangi, A.F.: Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding. In: CVPR (2017)
7. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13
8. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI **38**, 295–307 (2016)
9. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
10. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
11. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
12. Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H.: Deep Laplacian pyramid networks for fast and accurate superresolution. In: CVPR (2017)
13. Ahn, N., Kang, B., Sohn, K.-A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 256–272. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_16
14. Sajjadi, M.S., Schölkopf, B., Hirsch, M.: EnhanceNet: single image super-resolution through automated texture synthesis. In: ICCV (2017)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
16. Bulat, A., Tzimiropoulos, G.: Super-FAN: integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs. In: CVPR (2018)
17. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW (2017)
18. Wang, Y., Perazzi, F., McWilliams, B., Sorkine-Hornung, A., Sorkine-Hornung, O., Schroers, C.: A fully progressive approach to single-image super-resolution. In: CVPRW (2018)
19. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
20. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Boissonnat, J.-D., et al. (eds.) Curves and Surfaces 2010. LNCS, vol. 6920, pp. 711–730. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27413-8_47
21. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
22. Huang, J.-B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)

23. Fujimoto, A., Ogawa, T., Yamamoto, K., Matsui, Y., Yamasaki, T., Aizawa, K.: Manga109 dataset and creation of metadata. In: International Workshop on coMics ANalysis, Processing and Understanding (2016)
24. Timofte, R., et al.: NTIRE 2017 challenge on single image super-resolution: methods and results. In: CVPRW (2017)
25. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
26. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR (2018)
27. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Trans. Image Process. **15**, 3440–3451 (2006)
28. Wang, Z., Bovik, A.C.: Mean squared error: love it or leave it? A new look at signal fidelity measures. IEEE Sig. Process. Mag. **26**, 98–117 (2009)
29. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 391–407. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_25
30. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR (2016)
31. Timofte, R., Rothe, R., Van Gool, L.: Seven ways to improve example-based single image super resolution. In: CVPR (2016)
32. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
33. Ancuti, C., et al.: NTIRE 2018 challenge on image dehazing: methods and results. In: CVPRW (2018)
34. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
35. Mao, X., Shen, C., Yang, Y.-B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS (2016)
36. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016)
37. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
38. Wang, X., et al.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11133, pp. 63–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11021-5_5
39. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
40. Aly, H.A., Dubois, E.: Image up-sampling using total-variation regularization with a new observation model. IEEE Trans. Image Process. **14**, 1647–1659 (2005)
41. Sergey, I., Christian, S.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)