Deep Learning for Single Image Super-Resolution: A Brief Review

Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue

***Abstract*—Single image super-resolution (SISR) is a notoriously challenging ill-posed problem, which aims to obtain a high-resolution (HR) output from one of its low-resolution (LR) versions. To solve the SISR problem, recently powerful deep learning algorithms have been employed and achieved the state-of-the-art performance. In this survey, we review representative deep learning-based SISR methods, and group them into two categories according to their major contributions to two essential aspects of SISR: the exploration of efficient neural network architectures for SISR, and the development of effective optimization objectives for deep SISR learning. For each category, a baseline is firstly established and several critical limitations of the baseline are summarized. Then representative works on overcoming these limitations are presented based on their original contents as well as our critical understandings and analyses, and relevant comparisons are conducted from a variety of perspectives. Finally we conclude this review with some vital current challenges and future trends in SISR leveraging deep learning algorithms.**

***Index Terms*—Single image super-resolution, deep learning, neural networks, objective function**

I. Introduction

DEEP learning (DL) [1] is a branch of machine learning algorithms and aims at learning the hierarchical representation of data. Deep learning has shown prominent superiority over other machine learning algorithms in many artificial intelligence domains, such as computer vision [2], speech recognition [3] and nature language processing [4]. Generally speaking, DL is endowed with the strong capacity of handling substantial unstructured data owing to two main contributors: the development of efficient computing hardware and the advancement of sophisticated algorithms.

Single image super-resolution (SISR) is a notoriously challenging ill-posed problem, because a specific low-resolution (LR) input can correspond to a crop of possible high-resolution (HR) images, and the HR space (in most instances it refers to the nature image space) that we intend to map the LR input to is usually intractable [5]. Previous methods for SISR mainly have two drawbacks: one is the unclear definition of the mapping that we aim to develop between the LR space and the HR space, and the other is the inefficiency of establishing a complex high-dimensional mapping given massive raw data. Benefiting from the strong capacity of extracting effective high-level abstractions which bridge the LR space and HR space, recent DL-based SISR methods have achieved significant improvements, both quantitatively and qualitatively.

In this survey, we attempt to give an overall review of recent DL-based SISR algorithms. Our main focus is on two areas: efficient neural network architectures designed for SISR and effective optimization objectives for DL-based SISR learning. The reason for this taxonomy is that when we apply DL algorithms to tackle a specified task, it is best for us to consider both the universal DL strategies and the specific domain knowledge. From the perspective of DL, although many other techniques such as data preprocessing [6] and model training techniques are also quite important [7], [8], the combination of DL and domain knowledge in SISR is usually the key of success and is often reflected in the innovations of neural network architectures and optimization objectives for SISR. In each of these two focused areas, we firstly introduce a benchmark, and then discuss several representative researches about their original contributions and experimental results, as well as our comments and views.

The rest of the paper is arranged as follows. In Section II, we present relevant background concepts of SISR and DL. In Section III, we survey the literatures on exploring efficient neural network architectures for various SISR tasks. In Section IV we survey the researches on proposing effective objective functions for different purposes. In Section V, we summarize some trends and challenges for DL-based SISR. We conclude this survey in Section VI.

II. Background

A. Single Image Super-Resolution

Super-resolution (SR) [9] refers to the task of restoring high-resolution images from one or more low-resolution observations of the same scene. According to the number of input LR images, the SR can be classified into single image super-resolution (SISR) and multi-image super-resolution (MISR). Compared with MISR, SISR is much more popular because of its high efficiency. Since an HR image with high perceptual quality has more valuable details, it is widely useful in many areas, such as medical imaging, satellite imaging and security imaging. It is well known that SISR is an extremely ill-posed problem because one LR input may correspond to many possible HR solutions. Up to now, mainstream algorithms of

This work was partly supported by the National Natural Science Foundation of China (No.61471216 and No.61771276) and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No.JCYJ20170307153940960 and No.JCYJ20170817161845824). (Corresponding author: Wenming Yang)

W. Yang, X. Zhang, Y. Tian and W. Wang are with the Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, China (E-mail: {yang.wenming@sz, xc-zhang16@mails, typ14@mails, wangwei17@mails}.tsinghua.edu.cn.

J.-H. Xue is with the Department of Statistical Science, University College London, UK (E-mail: jinghao.xue@ucl.ac.uk).

SISR are mainly divided into three categories: interpolation-based methods, reconstruction-based methods and learning-based methods.

Interpolation-based SISR methods, such as bicubic interpolation [10] and Lanczos resampling [11], are very fast and simple but suffer from the shortage of accuracy. Reconstruction-based SR methods [12], [13], [14], [15] often adopt sophisticated prior knowledge to restrict the possible solution space with an advantage of generating flexible and sharp details. However, the performance of many reconstruction-based methods degrades rapidly when the scale factor increases, and these methods are usually time-consuming.

Learning-based SISR methods, also known as example-based methods, are brought into focus because of their fast computation and outstanding performance. These methods usually utilize machine learning algorithms to analyze statistical relationships between the LR and its corresponding HR counterpart from substantial training examples. Markov Random Field (MRF) [16] was firstly adopted by Freeman *et al.* to exploit the abundant real-world images to synthesize visually pleasing image textures. Neighbor embedding methods [17] proposed by Chang *et al.* took advantage of similar local geometry between LR and HR to restore HR image patches. Inspired by the sparse signal recovery theory [18], researchers applied sparse coding methods [19], [20], [21], [22], [23] to SISR problems. Lately, random forest [24] was also used to achieve improvement in the reconstruction performance. Very recently, DL-based SISR algorithms have demonstrated great superiority over reconstruction-based methods and other learning-based methods.

B. Deep Learning

Deep learning, also called deep structured learning, is a branch of machine learning algorithms based on directly learning diverse representations of data [25]. Opposed to traditional task-specific learning algorithms, which select useful handcraft features with expert knowledge of the domain, deep learning algorithms aim to automatically learn informative hierarchical representations and then leverage them to achieve the final purpose, where the whole learning process can be seen as an entirety [26].

Because of the great approximating capacity and hierarchical property of artificial neural network (ANN), most modern deep learning models are based on ANN [27]. Early ANN could be traced back to perceptron algorithms in 1960s [28]. Then in 1980s, multilayer perceptron could be trained with the backpropagation algorithm [29], and convolutional neural network (CNN) [30] and recurrent neural network (RNN) [31], two representative derivatives of traditional ANN, were introduced to the computer vision and speech recognition fields, respectively. Despite noticeable progress that ANN had made in that period, there were still many deficiencies handicapping ANN from going further [32], [33]. The rebirth of modern ANN was marked by pretraining the deep neural network (DNN) with restricted Boltzmann machine (RBM) proposed by Hinton in 2006 [34]. After that, benefiting from the booming of computing power and the development of advanced algorithms, models based on DNN have achieved remarkable performance in various supervised tasks [35], [36], [2]. Meanwhile, DNN-based unsupervised algorithms such as deep Boltzmann machine (DBM) [37], variational auto-encoder (VAE) [38], [39] and generative adversarial nets (GAN) [40] have attracted much attention owing to their potential to handling tough unlabeled data.

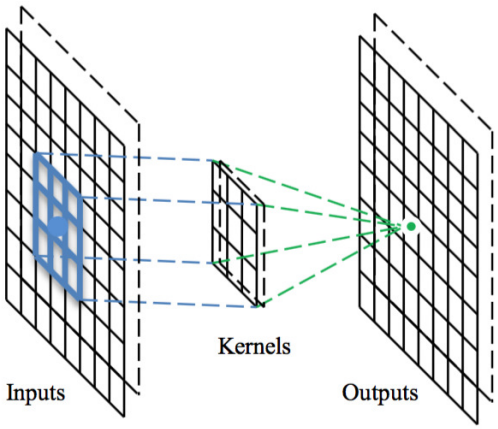


Figure 1: Sketch of a basic convolution operator in the CNN architecture.

Readers can refer to [41] for extensive analysis of DL. As most literatures on DL-based SISR choose CNN as their basic frame, for better discussing these models, here we give a brief description of a basic CNN unit. Other DL algorithms relevant to SISR will be depicted in this paper along with our discussion.

Like [42], we use the stride convolution instead of pooling operator. To guarantee the nonlinearity, usually there is an element-wise nonlinear operator in every unit. Fig.1 shows a basic convolution operator in the CNN unit, with the input feature as $X(i,j,k)$, $0 \le i < H_{in}$, $0 \le j < W_{in}$, $0 \le k < C_{in}$, where H_{in}, W_{in} and C_{in} denote the height, width and channel number of input, respectively. The kernel tensor is $K(l,k,m,n)$, $0 \le m < M$, $0 \le n < N$, where M and N is the height and width of convolutional kernel, l and k denote the channels of output and input, respectively. Similarly, the corresponding output feature is $Y(i,j,l)$, $0 \le i < H_{out}$, $0 \le j < W_{out}$, $0 \le l < C_{out}$. If we sample the input at intervals of s steps in each direction and pad the input feature to maintain its size, then the convolution operator can be formulated as

$$Y(i,j,l) = \sum_{m,n,k} K(l,k,m,n)X((i-1)s+m,(j-1)s+n,k). \tag{1}$$

When $s = 1$, it can be simplified as

$$Y(i,j,l) = \sum_{m,n,k} K(l,k,m,n)X(i+m-1,j+n-1,k). \tag{2}$$

III. Deep Architectures for SISR

In this section, we mainly discuss the efficient architectures proposed for SISR in recent years. Firstly, we set the network

architecture of super-resolution CNN (SRCNN) [43], [44] as the benchmark. When we discuss each related architecture in detail, we focus on their universal parts that can be applicable to other tasks and their specific parts that take the property of SISR into consideration. When it comes to the comparison among different models, for the sake of fairness, we will illustrate the importance of the training dataset and try to compare models with the same training dataset.

A. Benchmark of Deep Architecture for SISR

We select the SRCNN architecture as the benchmark in this section. The overall architecture of SRCNN is shown in Fig.2. As many traditional methods, SRCNN only takes the luminance components for training for simplicity. SRCNN is a three-layer CNN, where its filter size of each layer, following the manner mentioned in Section II-B, is $64 \times 1 \times 9 \times 9$, $32 \times 64 \times 5 \times 5$ and $1 \times 32 \times 5 \times 5$, respectively. The functions of these three nonlinear transformations are patch extraction, nonlinear mapping and reconstruction. The loss function for optimizing SRCNN is the mean square error (MSE), which will be discussed in next section.

The formulation of SRCNN is relatively simple, which can be seen as an ordinary CNN approximating the complex mapping between the LR and HR spaces in an end-to-end manner. SRCNN is reported to demonstrate great superiority over traditional methods at that time, we argue that it is owing to the CNN's strong capability to learn effective representations from big data in an end-to-end manners.

Despite the success of SRCNN, the following problems have inspired more effective architectures:

1) The input of SRCNN is the bicubic of LR, which is an approximation of HR. However, these interpolated inputs have three drawbacks: (a) detail-smoothing effects brought by these inputs may lead to further wrong estimation of image structure; (b) taking interpolated versions as input is very time-consuming; (c) when downsampling kernel is unknown, one specific interpolated input as raw estimation is not reasonable. Therefore, the first question is emerging: can we design CNN architectures directly taking LR as input to tackle these problems? [1]

2) The SRCNN is just three-layer; can more complex (with different depth, width and topology) CNN architectures achieve better results? If yes, then how can we design such more complex models?

3) The prior terms in the loss function which reflect properties of HR images are trivial; can we integrate any property of the SISR process into the design of CNN frame or other parts in the algorithms for SISR? If yes, then can these deep architectures with SISR properties be more effective in handling some difficult SISR problems, such as the big scale factor SISR and the unknown downsampling SISR?

Based on some solutions to these three questions, recent researches on deep architectures for SISR will be discussed in sections III-B1, III-B2 and III-B3, respectively.

[1]Generally speaking, the first problem can be grouped into the third problem below. Because the solutions to this problem are the basic of many other models, it is necessary to introduce this problem separately in the first place.

B. State-of-the-Art Deep SISR Networks

1) Learning Effective Upsampling with CNN: One solution to the first question is to design a module in the CNN architecture which adaptively increases the resolution. Convolution with pooling and stride convolution are the common downsampling operators in the basic CNN architecture. Naturally people can implement upsampling operation which is known as deconvolution [46] or transposed convolution [47]. Given the upsampling factor, the deconvolution layer is made up of an arbitrary interpolation operator (usually we choose the nearest neighbor interpolation for simplicity) and a following convolution operator with stride 1, as shown in 3. Readers should be aware that such deconvolution may not totally recover the information missing from convolution with pooling or stride convolution. Such a deconvolution layer has been successfully adopted in the context of network visualization [48], semantic segmentation [49] and generative modeling [50]. For a more detailed illustration of the deconvolution layer, readers can refer to [51]. To the best of our knowledge, FSRCNN [52] is the first work using this normal deconvolution layer to reconstruct HR images from LR feature maps. As mentioned before, the usage of deconvolution layer has two main advantages: one is the reduction of computation because we just need to increase resolution at the end of network; the other is when the downsampling kernel is unknown, many literatures [53] have shown that inputting an inaccurate estimation has side effects on the final performance.

Although the normal deconvolution layer, that has already been involved in popular open-source packages such as Caffe [54] and Tensorflow [55], offers a quite good solution to the first question, there is still an underlying problem: when we use the nearest neighbor interpolation, the points in the upsampled features are repeated several times in each direction. This configuration of the upsampled pixels is redundant. To deal with this problem, Shi *et al.* proposed an efficient sub-pixel convolution layer in [45], known as ESPCN and the structure of ESPCN is shown in Fig.4. Rather than increasing resolution by explicitly enlarging feature maps as the deconvolution layer does, ESPCN expands the channels of the output features for storing the extra points to increase resolution, and then rearranges these points to get the HR output through a certain mapping criterion. As the expansion is carried out in the channel dimension, so a smaller kernel size is enough. [51] further shows that when the common but redundant nearest neighbor interpolation is replaced with the interpolation that pads the sub-pixels with zero, the deconvolution layer can be simplified into the sub-pixel convolution in ESPCN. Obviously, compared with the nearest neighbor interpolation, this interpolation is more efficient, which can also verify the effectiveness of ESPCN.

2) The Deeper, The Better: In the DL community, there are theoretical literatures [56] showing that the solution space of a DNN can be expanded by increasing its depth or its width. In some situations, to get more hierarchical representations more effectively, many works mainly focus on improvement brought by depth increasing. Recently, various DL-based applications have also demonstrated the great power of very deep neural

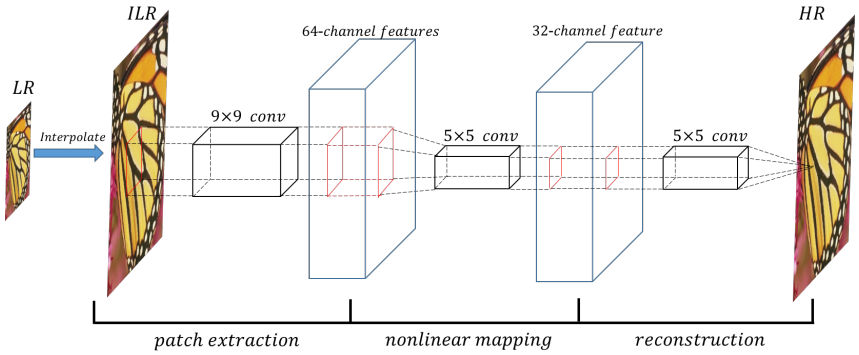


Figure 2: Sketch of the SRCNN architecture.

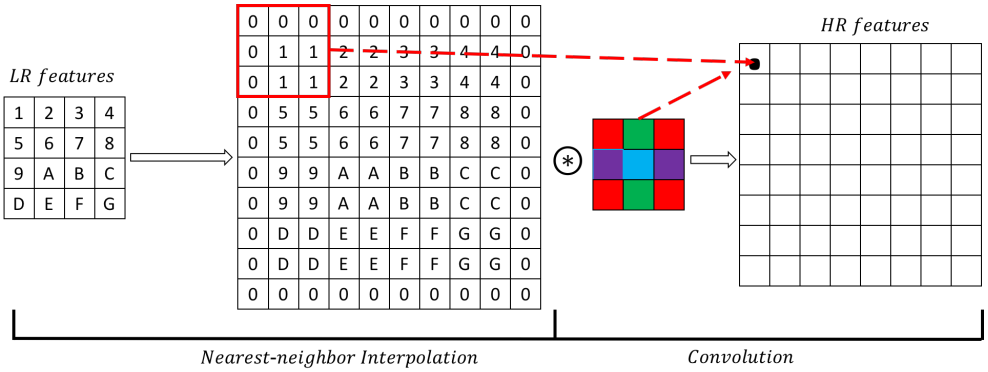


Figure 3: Sketch of the deconvolution layer used in FSRCNN [44], where ⊛ denotes the convolution operator.

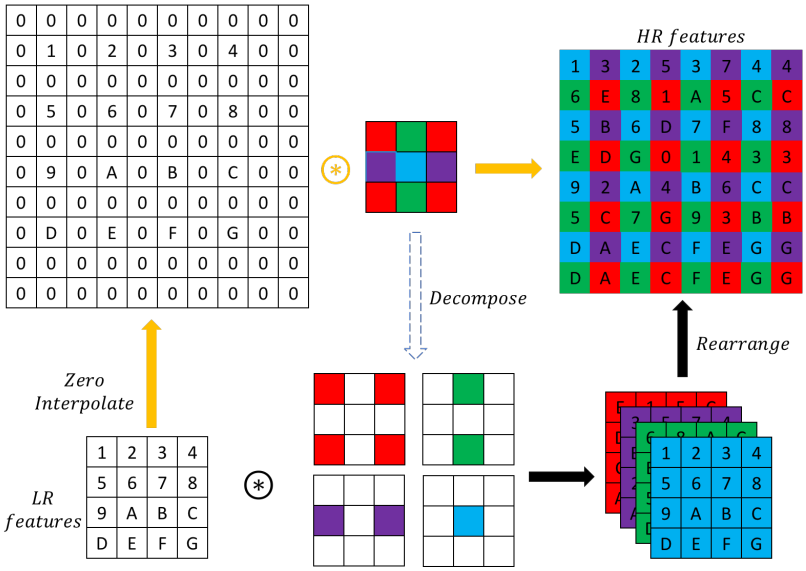


Figure 4: Detailed sketch of ESPCN [45]. The top process with yellow arrow depicts the ESPCN from the view of zero interpolation, while the bottom process with black arrow is the original ESPCN; ⊛ denotes the convolution operator.

networks despite many training difficulties. VDSR [57] is the first very deep model used in SISR. As shown in Fig.5(a), VDSR is a 20-layer VGG-net [58]. The VGG architecture sets all kernel size as 3×3 (the kernel size is usually an odd, and taken the increasing of receptive field into account, 3×3 is the smallest kernel size). To train this deep model, they used a relatively big initial learning rate to accelerate convergence and used gradient clipping to prevent annoying gradient explosion.

Besides the novel architecture, VDSR has made two other contributions. The first one is that a single model is used for multiple scales based on the fact that the SISR processes with different scale factors have strong relationship with each other.

This fact is the basic of many traditional SISR methods. Like SRCNN, VDSR takes the bicubic of LR as input. During training, VDSR put the bicubics of LR of different scale factors together for training. For larger scale factors ($\times 3, \times 4$), the mapping for a smaller scale factor ($\times 2$) may be also informative.

The second contribution is the residual learning. Unlike the direct mapping from the bicubic version to HR, VDSR uses deep CNN mediately to learn the mapping from the bicubic to the residual between the bicubic and HR. They argued that residual learning can improve performance and accelerate convergence.

The convolution kernels in the nonlinear mapping part of VDSR are very similar, in order to reduce parameters, Kim *et al.* further proposed DRCN [59], which utilizes the same convolution kernel in the nonlinear mapping part for 16 times, as shown in Fig.5(b). To overcome the difficulties of training deep recursive CNN, a multi-supervised strategy is applied and the final result can be regarded as the fusion of 16 intermediate results. The coefficients for fusion are a list of trainable positive scalars whose summation is 1. As they showed, DRCN and VDSR have the quite similar performance.

Here we believe it is necessary to emphasize the importance of the multi-supervised training in DRCN. This strategy not only creates short paths through which the gradients can flow more smoothly during back-propagation, but also guides all the intermediate representation to reconstruct raw HR outputs. Finally fusing all these raw HR outputs produces a wonderful result. However, as for fusion, this strategy has two flaws: 1) once the weight scalars are determined in the training process, it will not change with different inputs; 2) using single scalars to weight HR outputs does not take pixel-wise differences into consideration, that is to say, it would be better to weight different parts distinguishingly in an adaptive way.

A plain architecture like VGG-net is hard to go deeper. Various deep models based on skip-connection can be extremely deep and have achieved the state-of-the-art performance in many tasks. Among them, ResNet [60], [61] proposed by He *et al.* is the most representative one. Readers can refer to [62], [63] for further discussions on why ResNet works well. In [64], the authors proposed SRResNet made up of 16 residual units (a residual unit consists of two nonlinear convolutions with residual learning). In each unit, batch normalization (BN) [65] is used for stabilizing the training process. The overall architecture of SRResNet is shown in Fig.5(c). Based on the original residual unit in [61], Tai *et al.* proposed DRRN [66], in which basic residual units are rearranged in a recursive topology to form a recursive block, as shown in Fig.5(d). Then for the sake of parameter reduction, each block shares the same parameters and is reused recursively, just like the single recursive convolution kernel in DRCN.

EDSR [67] proposed by Lee *et al.* has achieved the state-of-the-art performance up to now. EDSR mainly has made three improvements on the overall frame: 1) Compared with the residual unit used in previous work, EDSR removes the usage of BN, as shown in Fig.5(e). The original ResNet with BN was designed for classification, where inner representations are highly abstract and these representations can be insensitive to the shift brought by BN. As for image-to-image tasks like SISR, since the input and output are strongly related, if the convergence of the network is of no problem, then such shift may harm the final performance. 2) Except for regular depth increasing, EDSR also increases the number of output features of each layer on a large scale. To release the difficulties of training such wide ResNet, the residual scaling trick proposed in [68] is employed. 3) Also inspired by the fact that the SISR processes with different scale factors have strong relationship with each other, when training the models for $\times 3$ and $\times 4$ scales, the authors of [67] initialized the parameters with the pre-trained $\times 2$ network. This pre-training strategy accelerates the training and improves the final performance.

The effectiveness of the pre-training strategy in EDSR implies that models for different scales may share many intermediate representations. To explore this idea further, like building a multi-scale architecture as VDSR does on the condition of bicubic input, the authors of EDSR proposed MDSR to achieve the multi-scale architecture, as shown in Fig.5(g). In MDSR, the convolution kernels for nonlinear mapping are shared across different scales, only the front convolution kernels for extracting features and the final sub-pixel upsampling convolution are different. At each update during training MDSR, minibatches for $\times 2$, $\times 3$ and $\times 4$ are randomly chosen and only the corresponding parts of MDSR are updated.

Besides ResNet, DenseNet [69] is another effective architecture based on skip connections. In DenseNet, each layer is connected with all the preceding representations, and the bottleneck layers are used in units and blocks to reduce the parameter number. In [70], the authors pointed out that ResNet enables feature re-usage while DenseNet enables new features exploration. Based on the basic DenseNet, SR-DenseNet [71],as shown in Fig.5(f) further concatenates all the features from different blocks before the deconvolution layer, which is shown to be effective on improving performance. Memnet [72] proposed by Tai *et al.* uses residual unit recursively to replace the normal convolution in the block of the basic DenseNet and add dense connections among different blocks, as shown in Fig.5(h). They explained that the local connections in the same block resemble the short-term memory and the connections with previous blocks resemble the long-term memory [73]. Recently, RDN [74] proposed by Zhang *et al.* uses the similar structure. In an RDN block, basic convolution units are densely connected like DenseNet, and at the end of an RDN block, a bottleneck layer is used following with the residual learning across the whole block. Before entering the reconstruction part, features from all previous blocks are fused by dense connection and residual learning.

3) Combining Properties of SISR Process with the Design of CNN Frame: In this sub-section, we mainly discuss some deep frames taking certain properties of SISR into consideration. Here we illustrate some representative methods which combine these two aspects innovatively.

The sparse prior in nature images and the relationships between the HR and LR spaces rooted from this prior were widely used for its great performance and theoretical support.

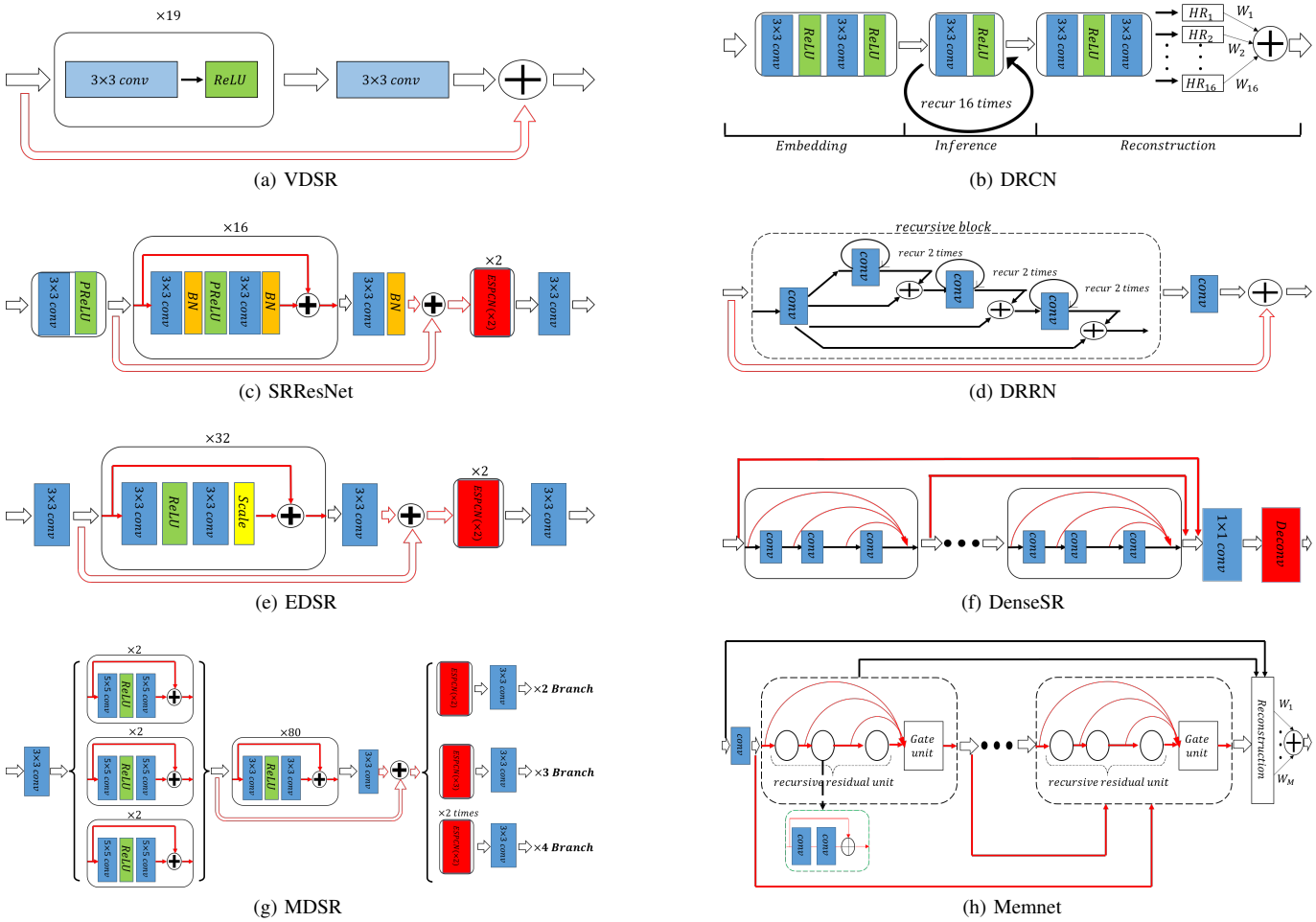


Figure 5: Sketch of several deep architectures for SISR.

To combine the benefits of sparse prior and deep learning, SCN [75] proposed by Wang *et al.* uses the learned iterative shrinkage and thresholding algorithm (LISTA) [76], an algorithm on producing approximate estimation of sparse coding based on NN, to solve the time-consuming inference in traditional sparse coding SISR. They further introduced a cascaded version (CSCN) [77] which employs multiple SCNs. Previous works such as SRCNN tried to explain general CNN architectures with the sparse coding theory, which from today's view, may be a bit unconvincing. SCN combines these two important concepts innovatively and gains both quantitative and qualitative improvements.

Different models specialize in different image patterns of SISR. From the perspective of ensemble learning, a better result can be acquired by adaptively fusing various models with different purposes at the pixel level. Motivated by this idea, MSCN proposed by Liu *et al.* develops an extra module in the form of CNN, taking the LR as input and outputting several tensors with the same shape as the HR. These tensors can be viewed as adaptive element-wise weights for each raw HR output. By selecting NN as the raw SR inference modules, the raw estimating parts and the fusing part can be optimized jointly. The two mentioned deficiencies of the multi-supervised training in DRCN can be mitigated by the fusion module of MSCN. However, in MSCN, the summation of coefficients at each pixel is not 1, which may be a little incongruous.

Compared with other deep architectures, ResNet is intriguing for its progressive properties. Taking SRResNet for example, one can observe that directly sending the representations produced by intermediate residual blocks to the final reconstruction part will also yield a quite good raw HR estimator. The deeper these representations are, the better the results can be obtained. A similar phenomenon of ResNet applied in recognition is reported in [62]. DEGREE [78] proposed by Yang *et al.* combines this progressive property of ResNet with the sub-band reconstruction of traditional SR methods [79]. The residues learned in each residual block can be used to reconstruct high-frequency details, resembling the signals from certain high-frequency band. To simulate sub-band reconstruction, a recursive residual block is used. Compared with the traditional supervised sub-band recovery methods which need to obtain sub-band ground truth by diverse filters, this simulation with recursive ResNet avoids explicitly estimating intermediate sub-band components benefiting from the end-to-end representation learning. Besides this innovation, DEGREE also adds extra supervision on the high-frequency components

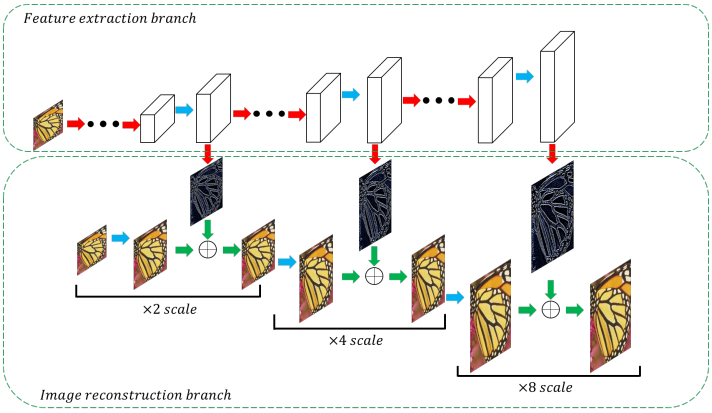


Figure 6: LapSRN architecture. Red arrows indicate convolutional layer; blue arrows indicate transposed convolutions (upsampling); green arrows denote element-wise addition operators.

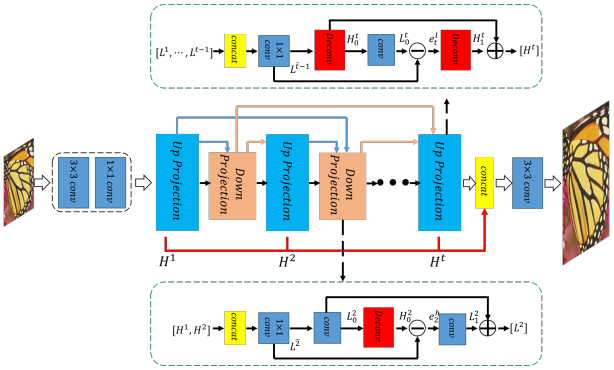


Figure 7: Sketch of the DBPN architecture.

of HR.

As mentioned above, models for small scale factors can be used for a raw estimator of big scale SISR. In the SISR community, SISR under big scale factors (*e.g.*$\times 8$) has been a challenging problem for a long time. In such situations, plausible priors are imposed to restrict the solution space. One simple way to tackle this is to gradually increase resolution by adding extra supervision on the auxiliary SISR process of small scale. Based on this heuristic prior, LapSRN [80] proposed by Lai *et al.* uses the Laplacian pyramid structure to reconstruct HR outputs. LapSRN has two branches: the feature extraction branch and the image reconstruction branch, as shown in Fig.6. At each scale, the image reconstruction branch estimates a raw HR output of the present stage, and the feature extraction branch outputs a residue between the raw estimator and the corresponding ground truth as well as extracts useful representations for next stage. Notably, all deconvolution layers are initialized with a bilinear filter and two branches are jointly trained.

Iterative back-projection [81] is an early SR algorithm which iteratively computes the reconstruction error then feeds it back to tune the HR results. Recently, DBPN [82] proposed by Haris *et al.* uses deep architectures to simulate iterative back-projection and further improves performance with dense connections [69], which is shown to achieve wonderful performance in $\times 8$ scale. As shown in Fig.7, dense connection and 1×1 convolution for reducing dimension is first applied across different up-projection (down-projection) units, then in the tth up-projection unit, the current LR feature input $\widetilde{L^{t-1}}$ is firstly deconvoluted to get a raw HR feature H_0^t, and H_0^t is back projected to the LR feature L_0^t. The residue between two LR features $e_t^l = L^{t-1} - L_0^{t-1}$ is then deconvoluted and added to H_0^t to get a finer HR feature H^t. The down-projection unit is defined very similarly in an inverse way.

For small scale factors, algorithms of recovering lost details can handle SISR effectively. When faced with big scale factors with severe loss of necessary details, some researchers suggest that synthesizing rational details can have better results. In this situation, deep generative models to be mentioned in next sections could be good choices. Compared with the traditional independent point estimation of the lost information, conditional autoregressive generative models using conditional maximum likelihood estimation in directional graphical models gradually generate high resolution images based on the previous generated pixels[2]. PixelRNN [83] and PixelCNN [84] are recent representative autoregressive generative models. The current pixel in PixelRNN and PixelCNN is explicitly dependent on the left and top pixels which have been already generated. To implement such operations, novel network architectures are elaborated. [85] proposed by Dahl *et al.* first applies conditional PixelCNN to SISR. The overall architecture of [85] is shown in Fig.8. The conditioning CNN taking LR as input, which provides LR-conditional information to the whole model, and the PixelCNN part is the auto-regressive inference part. The current pixel is determined by these two parts together using the current softmax probability:

$$P(y_i|x, y_{<i}) = \text{softmax}(A_i(x) + B_i(y_{<i})), \tag{3}$$

where x is the LR input, y_i is the current HR pixel to be generated and $y_{<i}$ are the generated pixels, $A_i(\cdot)$ denotes the conditioning network predicting a vector of logit values corresponding to the possible values, and $B_i(\cdot)$ denotes the prior network predicting a vector of logit value of the ith output pixel. Pixels with the highest probability are taken as the final output pixel. Similarly, the whole network is optimized by minimizing cross-entropy loss (maximizing the corresponding log-likelihood) between the model's prediction and the discrete ground-truth labels $y_i^* \in \{1, 2, \cdots, 256\}$ as follows:

$$\max \sum_{(x,y^*)\in D} \sum_{i=1}^{M} (I[y_i^*]^T (A_i(x) + B_i(y_{<i}^*)) - Ise(A_i(x) + B_i(y_{<i}^*))), \tag{4}$$

where D is the training set, M is the number of the pixels, $I[k]$ is the 256-dimensional one-hot indicator vector with its kth element set to 1, and $Ise(\cdot)$ is the log-sum-exp operator corresponding to the logarithm of the denominator of a softmax.

[2]We illustrate autoregressive models in this section mainly because we believe, when these models are used for synthesis, their novel network architecture for conditional generation is the main innovation.

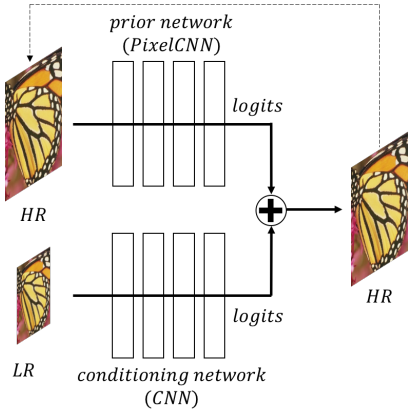


Figure 8: Sketch of the pixel recursive SR architecture.

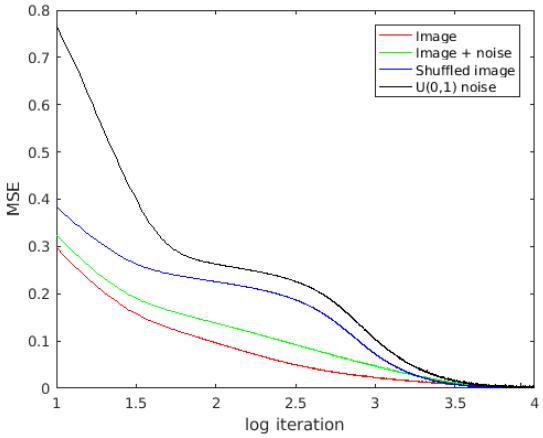


Figure 9: Learning curves for the reconstruction of different kinds of images. We re-implement the experiment in [89] with the image 'butterfly' in Set5.

Internal-example SISR algorithms are based on the recurrence of small pieces of information across different scales of a single image, which are shown to be better at dealing with specific details rarely existing in other external images. ZSSR [86] proposed by Shocher *et al.* is the first literature combining deep architectures with the internal-example learning. In ZSSR, besides the image for testing, no extra images are needed and all the patches for training are taken from different degraded pairs of the test image. As demonstrated in [87], the visual entropy inside a single image is much smaller than the vast training dataset collected from wide ranges, so unlike external-example SISR algorithms, a very small CNN is sufficient. As we mentioned before in VDSR, the training data for a small-scale model can also be useful for training big-scale models. Also based on this trick, ZSSR can be more robust by collecting more internal training pairs with small scale factors for training big-scale models. However, it will increase runtime immensely. Notably, when the downsampling kernel is unknown, by evaluating the kernel directly with the algorithm mentioned in [88] and then feeding it to ZSSR, ZSSR will outperform other algorithms by a large margin.

Recently, Ulyanov *et al.* showed in [89] that the structure of deep neural networks can capture a great deal of low-level image statistical prior. They reported that when neural networks are used to fit images of different statistical properties, the convergence speed for different kinds of images can be also different. As shown in Fig.9, naturally-looking images, whose different parts are highly relevant, will converge much faster. On the contrary, images such as noises and shuffled images, which have little inner relationship, tend to converge more slowly. As many inverse problems such as denoising and super-resolution are modeled as the pixel-wise summation of the original image and the independent additive noises. Based on the observed prior, when used to fit these degraded images, the neural networks tend to fit the naturally-looking images first, which can be used to retain the naturally-looking parts as well as filter the noisy ones. To illustrate the effectiveness of the proposed prior for SISR, only given the LR x_0, they took a fixed random vector z as input to fit the HR x with a randomly-initialized DNN f_θ by optimizing

$$\min_{\theta} ||d(f_\theta(z)) - x_0||_2^2, \tag{5}$$

where $d(\cdot)$ is common differentiable downsampling operator. The optimization is terminated in advance for only filtering noisy parts. Although this novel totally unsupervised methods are outperformed by other supervised learning methods, it does considerably better than some other naive methods.

C. Comparisons among Different Models and Discussion

In this section, we compare the depth, the number of parameters and the quantitative results of the models discussed in Section III. It has been shown in [44] and [45] that the training datasets have great influences on the final performance, and usually more abundant training data will lead to better results. Generally, these models are trained via three main datasets: 1) 91 images from [19] and 200 images from [90], called the 291 dataset (some models only use 91 images); 2) images derived from ImageNet [91] randomly; and 3) the newly published DIV2K dataset [92]. Beside the different number of images each dataset contains, the quality of images in each dataset is also different. Images in the 291 dataset are usually small (on average 150×150), images in ImageNet are much bigger, while images in DIV2K are of very high quality. Because of the restricted resolution of the images in the 291 dataset, models on this set have difficulties in getting big patches with big receptive fields. Therefore, models based on the 291 datasets usually take the bicubic of LR as input, which is quite time-consuming. Table I compares different models on the depth, the number of parameters and the quantitative results of $\times 4$ scale with the Set5 test set.

From Table I we can see that generally as the depth and the number of parameters grow, the performance also improves. However, the growth rate of performance levels off. Recently some works on parameter economic models [93], [94] aiming at achieving relatively good performance with less computation were proposed, which are believed to be very meaningful

Models	PSNR/SSIM(×4)	Train data	Parameters	Layers
SRCNN_EX	30.49/0.8628	ImageNet subset	57K	3
ESPCN	30.90/-	ImageNet subset	20K	3
FSRCNN	30.71/0.8657	G100+Yang91	12K	5
VDSR	31.35/0.8838	G200+Yang91	665K	20
DRCN	31.53/0.8838	Yang91	1.77M(recursive)	20
DRRN	31.68/0.8888	G200+Yang91	297K(recursive)	54
LapSRN	31.54/0.8855	G200+Yang91	812K	24
SRResNet	32.05/0.9019	ImageNet subset	1.5M	37
Memnet	31.74/0.8893	G200+Yang91	677K(recursive)	80
RDN	32.61/0.9003	DIV2K	21.9M	149
EDSR	32.62/0.8984	DIV2K	43M	69
MDSR	32.60/0.8982	DIV2K	8M	166
DBPN	32.47/0.898	DIV2K+Flickr+ ImageNetb subset	10M	46

Table I: Comparisons among some representative deep models.

in practice. What's more, from the descriptions of various deep architectures above, we can see that many models only focus on alleviating training difficulties to improve performance and accuracy. There often lack of analyses on what representation the deep architectures have learned to improve performance. Also, despite great improvement in recent years, there still lack good solutions to some long-standing problems, such as the SISR with unknown corruption and the unsupervised SISR. To tackle this problem, we think only the usage of deep learning algorithms is not enough. Effective solutions can be achieved by combining the power of the deep learning and the properties of SISR.

IV. Optimization Objectives for DL-based SISR

A. Benchmark of Optimization Objectives for DL-based SISR

We select the MSE loss used in SRCNN as the benchmark. It is known that using MSE favors a high PSNR, and PSNR is a widely-used metric for quantitatively evaluating image restoration quality. Optimizing MSE can be viewed as a regression problem, leading to a point estimation of θ as

$$\min_{\theta} \sum_{i} ||F(x_i;\theta) - y_i||^2, \tag{6}$$

where (x_i, y_i) are the ith training examples and $F(x;\theta)$ is a CNN parameterized by θ. Here (6) can be interpreted in a probabilistic way by assuming Gaussian white noise ($\mathcal{N}(\epsilon; 0, \sigma^2 I)$) independent of image in the regression model, and then the conditional probability of y given x becomes a Gaussian distribution with mean $F(x;\theta)$ and diagonal covariance matrix $\sigma^2 I$, where I is the identity matrix:

$$p(y|x) = \mathcal{N}(y; F(x;\theta), \sigma^2 I). \tag{7}$$

Then using maximum likelihood estimation (MLE) on the training examples with (7) will lead to (6).

The Kullback-Leibler divergence (KLD) between the conditional empirical distribution P_{data} and the conditional model distribution P_{model} is defined as

$$D_{KL}(P_{data}||P_{model}) = E_{x\sim P_{data}}[\log P_{data}(x) - \log P_{model}(x)]. \tag{8}$$

We call (8) the forward KLD, where $E_{x\sim P_{data}}[\log P_{data}(x)]$ is an intrinsic term determined by the training data regardless of the parameter θ of the model (or say the model distribution $P_{model}(x;\theta)$). Hence when we use the training samples to estimate parameter θ, minimizing this KLD is equivalent to MLE.

Here we have demonstrated that MSE is a special case of MLE, and furthermore MLE is a special case of KLD. However, we may wonder what if the assumptions underlying these specialization are violated. This has led to some emerging objective functions from three perspectives:

1) Translating MLE into MSE can be achieved by assuming Gaussian white noise. Despite the Gaussian model is the most widely-used model for its simplicity and theoretical support, what if this independent Gaussian noise assumption is violated in a complicated scene like SISR?

2) To use MLE, we need to assume the parametric form the data distribution. What if the parametric form is mis-specified?

3) Apart from KLD in (8), are there any other distances between probability measures which we can use as the optimization objectives for SISR?

Based on some solutions to these three questions, recent researches on optimization objectives for DL-based SISR will be discussed in sections IV-B, IV-C and IV-D, respectively.

B. Objective Functions Based on non-Gaussian Additive Noises

The poor perceptual quality of the SISR images obtained by optimizing MSE directly demonstrates a fact: using Gaussian additive noise in the HR space is not good enough. To tackle

this problem, solutions are proposed from two aspects: use other distributions for this additive noise, or transfer the HR space to some space where the Gaussian noise is reasonable.

1) Denote Additive Noise with Other Probability Distribution: In [95], Zhao *et al.* investigated the difference between mean absolute error (MAE) and MSE used to optimize NN in image processing. Like (6), MAE can be written as

$$\min_{\theta} \sum_{i} ||F(x_i;\theta) - y_i||_1. \tag{9}$$

From the perspective of probability, (9) can be interpreted as introducing Laplacian white noise, and like (7), the conditional probability becomes

$$p(y|x) = Laplace(y; F(x;\theta), bI). \tag{10}$$

Compared with MSE in regression, MAE is believed to be more robust against outliers. As reported in [95], when MAE is used to optimize an NN, the NN tends to converge faster and produce better results. They argued that this might be because MAE could guide NN to reach a better local minimum. Other similar loss functions in robust statistics can be viewed as modeling additive noises with other probability distributions.

Although these specific forms of distribution often cannot represent unknown additive noise very precisely, their corresponding robust statistical loss functions are used in many DL-based SISR literatures for its conciseness and advantages over MSE.

2) Using MSE in a Transformed Space: Alternatively, we can search for a mapping Ψ to transform the HR space to a certain space where Gaussian white noise can be used reasonably. From this perspective, Bruna *et al.* [96] proposed so-called perceptual loss to leverage deep architectures. In [96], the conditional probability of the residual r between HR and LR given the LR x is stimulated by the Gibbs energy model:

$$p(r|x) = \exp(-||\Phi(x) - \Psi(r)||^2 - \log Z), \tag{11}$$

where Φ and Ψ are two mappings between the original spaces and the transformed ones, and Z is the partition function. The features produced by sophisticated supervised deep architectures have been shown to be perceptually stable and discriminative, denoted by $\Psi(r)$[3]. Then Ψ is the corresponding deep architectures. In contrast, Φ is the mapping between the LR space and the manifold represented by $\Psi(r)$, trained by minimizing the Euclidean distance as

$$\min_{\Phi} ||\Phi(x) - \Psi(r)||^2. \tag{12}$$

After Φ is obtained, the final result r can be inferred with SGD by solving

$$\min_{r} ||\Phi(x) - \Psi(r)||^2. \tag{13}$$

For further improvement, [96] also proposed a fine-tuning algorithm in which Φ and Ψ can be fine-tuned to the data. Like the alternating updating in GAN, Φ and Ψ are fine-tuned with SGD based on current r. However, this fine-tuning will involve calculating gradient of partition function Z, which is a well-known difficult decomposition into the positive phase and the negative phase of learning. Hence to avoid sampling within inner loops, a biased estimator of this gradient is chosen for simplicity.

The inference algorithm in [96] is extremely time-consuming. To improve efficiency, Johnson *et al.* utilized this perceptual loss in an end-to-end training manner [97]. In [97], the SISR network is directly optimized with SGD by minimizing the MSE in the feature manifold produced by VGG-16 as follows:

$$\min_{\theta} ||\Psi(F(x;\theta)) - \Psi(y)||^2, \tag{14}$$

where Ψ denotes the mapping represented by VGG-16, $F(x;\theta)$ denotes the SISR network, and y is the corresponding ground truth. Compared with [96], [97] replaces the nonlinear mapping Φ and the expensive inference with an end-to-end trained CNN, and results reported show that this change does not affect the restoration quality, but does accelerate the whole process.

Perceptual loss really avoids blurry and leads to more visually-pleasing results compared with directly optimizing MSE in HR space. However, there is still no theoretical support for the reason why it works. In [96], the author generally concluded that successful supervised networks used for high-level tasks can produce very compact and stable features. In these feature spaces, small pixel-level variation and many other trivial information can be omitted, making these feature maps mainly focusing on human-interested pixels. At the same time, with the deep architectures, the most specific and discriminative information of input are shown to be retained in feature spaces because of the great performance of the models applied in various high-level tasks. From this perspective, using MSE in these feature spaces will focus more on the parts which are attractive to human observers with little loss of original contents, so perceptually pleasing results can be obtained.

C. Optimizing Forward KLD Based on Non-parametric Estimation

Parametric estimation methods such as MLE need to specify in advance the parametric form the distribution of data, which suffer from model misspecification. Different from parametric estimation, non-parametric estimation methods such as kernel distribution estimation (KDE) fit the target distribution totally based on the given data, which is shown to be robust when the real distributional form is unknown. Based on non-parametric estimation, recently the contextual loss [98], [99] was proposed by Mechrez *et al.* to maintain natural image statistics. In the contextual loss, the following Gaussian kernel function is applied:

$$K(x,y) = \exp(-dist(x,y)/h - \log Z), \tag{15}$$

where $dist(x,y)$ can be any symmetric distance between x and y, h is the bandwidth, and the partition function $Z =$

[3]Either scattering network or VGG can be denoted by Ψ. When Ψ is VGG, there is no residual learning and fine-tuning.

$\int \exp(-dist(x,y)/h)dy$. Then P_{data} and P_{model} are

$$\begin{aligned} P_{data}(x) &= \sum_{x_i \sim P_{data}} K(x, x_i), \\ P_{model}(x) &= \sum_{y_i \sim P_{model}} K(x, y_i). \end{aligned} \tag{16}$$

Then (8) can be rewritten as

$$\begin{aligned} &D_{KL}(P_{data}||P_{model}) = \\ &\frac{1}{N}\sum_k [\log \sum_{x_i \sim P_{data}} K(x_k, x_i) - \log \sum_{y_j \sim P_{model}} K(x_k, y_j)]. \end{aligned} \tag{17}$$

The first log term in (17) is a constant with respect to the model parameters. Let us denote the kernel $K(x_k, y_j)$ in the second log term by A_{kj}. Then the optimization objective in (17) can be rewritten as

$$-\frac{1}{N}\sum_k \log \sum_j A_{kj}. \tag{18}$$

With the Jensen inequality we can get a lower bound of (18):

$$-\frac{1}{N}\sum_k \log \sum_j A_{kj} \geq -\log \frac{1}{N}\sum_k \sum_j A_{kj} \geq 0. \tag{19}$$

The first equality holds if and only if $\forall k, k'$, $\sum_j A_{kj} = \sum_j A_{k'j}$. Both equalities hold if and only if $\forall k$, $\sum_j A_{kj} = 0$. When (18) reaches 0, the given lower bound also reaches 0. Therefore, we can take this lower bound as optimization objective alternatively.

Here we can further simplify the lower bound in (19). The lower bound can be rewritten as

$$-\log \frac{1}{N}\sum_j ||A_j||_1, \tag{20}$$

where $A_j = (A_{1j}, \cdots, A_{kj})^T$, and $\|\cdot\|_1$ is the ℓ_1 norm. When the bandwidth $h \to 0$, the affinity A_{kj} will degrade into indicator function, which means if $x_k = y_j$, $A_{kj} \approx 1$, else $A_{kj} \approx 0$. In this case, ℓ_1 norm can be approximated well by ℓ_∞ norm, which returns the maximum element of the vector. Thus (20) can degenerate into the contextual loss in [98], [99]:

$$-\log \frac{1}{N}\sum_j \max_k A_{kj}. \tag{21}$$

From the analysis above we can clearly see the relationship between the forward KLD and the contextual loss. As reported in [98], [99], visually pleasing results can be generated by the contextual loss. However, KDE is generally very time-consuming for practical uses. To avoid this difficulty, several reasonable approximations are applied, especially the usage of lower bound in (19) and ℓ_∞ norm in (21). As we all know, training deep architectures is a complex non-convex optimization problem, the approximations that are useful in a convex problem will not guarantee good performance in complex non-convex situations. Therefore, there is still much difference between the forward KLD and the contextual loss when applied to optimize deep neural networks.

D. Other Distances between Probability Measures Used in SISR

1) Distances between Probabilities in the Form of Generative Adversarial Training: KLD is an asymmetric distance for measuring the distance between two probabilities. Given two distributions P_{data} and P_{model}, besides the forward KLD in (8), a backward KLD is

$$D_{KL}(P_{model}||P_{data}) = E_{x \sim P_{model}}[\log P_{model}(x) - \log P_{data}(x)]. \tag{22}$$

When $P_{model} = P_{data}$, both KLDs come to the minimum value 0. However, when the solution is inadequate, these two KLDs will lead to quite different results. As shown in Fig.10 for a toy example, a single peak Gaussian distribution (red dotted lines) is used to fit a multimodal Gaussian distribution, and we estimate the parameters of this Gaussian distribution by optimizing the two KLDs respectively. As shown in Fig.10(a), optimizing the forward KLD leads to the final results converging to middle areas among the different variates, while as shown in Fig.10(b), the backward KLD may focus on one peak. This average effect of the forward KLD is well-known as the regression-to-the-mean problem, which is very common when solution is inadequate in practice.

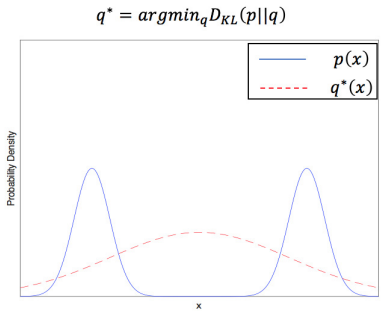
In [100], the authors argued that optimizing the backward KLD helps improve visual quality. It is rational to assume that human observers have learned the natural distribution P_{data}, and P_{data} can be seen as a kind of human prior belief. When we encounter an image, we mainly evaluate it by checking whether it conforms to our prior belief. That is, $E_{x \sim P_{model}(x)} \log P_{data}(x)$ should be minimized. To ensure diversity we can also add the entropy of P_{model}. When we carry out visual quality assessment, the main opinion score (MOS) testing is often adopted. Image quality rated by human observers can be seen as encoding what they see from their own perspective, and usually this encoding process is expressed in the form of cross entropy $-E_{x \sim P_{model}(x)} \log P_{data}(x)$.

Although minimizing the backward KLD will lead to better visual results with respect to non-reference perceptual quality assessment, in most low-level computer vision tasks, P_{data} is an empirical distribution and P_{model} is an intractable distribution. For this reason, the backward KLD is unpractical for optimizing deep architectures. To relieve optimizing difficulties, we replace the asymmetric KLD with the symmetric Jensen-Shannon divergence (JSD) as follows:

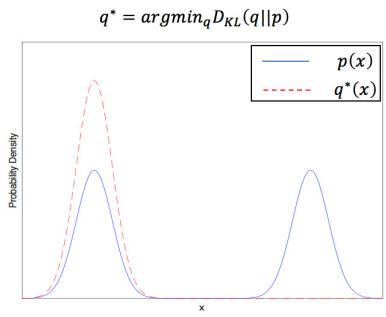
$$\begin{aligned} JS(P_{data}||P_{model}) =& \frac{1}{2}KL[P_{data}||\frac{P_{data}+P_{model}}{2}]+ \\ & \frac{1}{2}KL[P_{model}||\frac{P_{data}+P_{model}}{2}]. \end{aligned} \tag{23}$$

Optimizing (23) explicitly is also very difficult. Generative adversarial nets (GAN) proposed by Goodfellow *et al.* use the objective function below to implicitly handle this problem in a game theory scenario, successfully avoiding the troubling approximate inference and approximation of the partition function gradient:

$$\min_G \max_D [E_{x \sim P_{data}} \log D(x) + E_{x \sim P_{model}} \log(1 - D(x))], \tag{24}$$



(a) forward KLD



(b) backward KLD

Figure 10: A toy example to illustrate the difference between the forward KLD and the backward KLD. Suppose we have a multi-modal distribution $p(x)$ and want to approximate it with $q(x)$ under unideal conditions. (a) Minimizing the forward KLD: in this case, the algorithm tends to select the q that has high frequency where p has high frequency, resulting in blurring the modes together. (b) Minimizing the backward KLD: in this case, the algorithm tends to select the q that has low frequency where p has low frequency, then the backward KLD will be more likely to converge to a certain single mode.

where G is the main part called generator supervised by an auxiliary part D called discriminator. The two parts update alternatively and when the discriminator cannot give useful information to the generator anymore, in other words, the outputs of the generator totally confuse the discriminator, the optimization procedure is completed. For the detailed discussion on GAN, readers can refer to [40]. Recent works have shown that sophisticated architectures and suitable hyperparameters can help GAN perform excellently. The representative works on GAN-based SISR are [64] and [101]. In [64], the generator of GAN is the SRResNet mentioned before, and the discriminator refers to the design criterion of DCGAN [50]. In the context of GAN, a recent work [101] follows the similar path except with a different architecture. Fig.11 shows the results of GAN and MSE with the same architecture; despite the lower PSNR due to artifacts, the visual quality really improves by using GAN for SISR.

Furthermore, KLD and JSD are not the only choices and both of them have their own deficiency. GAN offers a methodology can be optimized implicitly in an adversarial training way by deep neural networks. Based on this, more rational but more complicated measures such as Wasserstein distances [102], f-divergence [103] [4] and maximum mean discrepancy (MMD) [104] are taken as alternatives of JSD for training GAN.

E. Discussion

In this section, we have discussed many optimization objectives on the baseline of MSE. Every loss function has its own characters, as shown in [105]. Even some objectives seem more universal and theoretically reasonable, when taken the practical optimization difficulties of these objectives into consideration, their real effects are not always better. For example, in medical imaging area, reconstruction accuracy is of key importance, MSE as a conservative choice is often the best objective. In other areas where perceptual quality may be preferred or scale factor is very big, MSE is often not a suitable choice. Therefore, we should be aware that there is no one-fits-all objective function and we should choose a suitable one to the context of an application.

V. Trends and Challenges

Along with the promising performance the DL algorithms have achieved in SISR, there remains several important challenges as well as inherent trends as follows.

1) Lighter Deep Architectures for Efficient SISR: Although high accuracy of the advanced deep models has been achieved for SISR, it is still difficult to deploy these model to real-world scenarios, mainly due to massive parameters and computation. To tackle this, we need to design light deep models or slim the existed deep models for SISR with less parameters and computation at the expense of little or no performance degradation. Hence in the future, researchers are expected to focus more on reducing the size of NN for speeding up SISR process.

2) More Effective DL Algorithms for Large-scale SISR and SISR with Unknown Corruption: Generally speaking, DL algorithms proposed in recent years have improved the performance of traditional SISR tasks by a large margin. However, the large scale SISR and the SISR with unknown corruption, the two big challenges in the SR community, are still lacking very effective remedies. DL algorithms are thought to be skilled at handling many inference or unsupervised problems, which is of key importance to tackle these two challenges. Therefore, leveraging great power of DL, more effective solutions to these two tough problems are expected.

3) Theoretical Understanding of Deep Models for SISR: The success of deep learning is said to be attributed to learning powerful representations. However, up to present, we still cannot understand these representations very well and the deep architectures are treated like a black box. As for DL-based SISR, the deep architectures are often viewed as a universal approximation, and the learned representations are often omitted for simplicity. This behavior is not beneficial for further exploration. Therefore, we should not only focus on whether a deep model works, but also concentrate on why it works and how it works. That is to say, more theoretical explorations are needed.

[4]Forward KLD, backward KLD and JSD can all be regarded as the special cases of f-divergence.

(a) HR (b) bicubic(21.59dB/0.6423) (c) SRResNet(23.53dB/0.7832) (d) SRGAN(21.15dB/0.6868)

Figure 11: Visual comparisons between the GAN loss and MSE (The authors of [64] released their results.) We can see that the GAN loss leads to a lower PSNR/SSIM, but a better visual quality can be achieved.

4) More Rational Assessment Criterion for SISR in Different Applications: In many applications, we need to design a desired objective function for a specific application. However, in most cases, we cannot give an explicit and precise definition to assess the requirement for the application, which leads to vagueness of the optimization objectives. Many works, although for different purposes, simply employ MSE as the assessment criterion, which has been shown as a poor criterion in many cases. In the future, we think it is of great necessity to make clear definitions for assessments in various applications. Based on these criterion, we can design better targeted optimization objectives and compare algorithms in the same context more rationally.

VI. Conclusion

This paper presents a brief review of recent deep learning algorithms on SISR. It divides the recent works into two categories: the deep architectures for simulating SISR process and the optimization objectives for optimizing the whole process. Despite the promising results reported so far, there are still many underlying problems. We summarize the main challenges into three aspects: the accelerating of deep models, the extensive comprehension of deep models and the criterion for designing and evaluating the objective functions. Along with these challenges, several directions may be further explored in the future.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[5] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *European Conference on Computer Vision*. Springer, 2014, pp. 372–386.

[6] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 1865–1873.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[9] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[10] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[11] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of applied meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[12] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, "Softcuts: a soft edge smoothness prior for color image super-resolution," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 969–981, 2009.

[13] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[14] Q. Yan, Y. Xu, X. Yang, and T. Q. Nguyen, "Single image superresolution based on gradient profile sharpness," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3187–3202, 2015.

[15] A. Marquina and S. J. Osher, "Image super-resolution by tv-regularization and bregman iteration," *Journal of Scientific Computing*, vol. 37, no. 3, pp. 367–382, 2008.

[16] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[17] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–I.

[18] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.

[19] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via

sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[20] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[21] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1920–1927.

[22] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[23] W. Yang, Y. Tian, F. Zhou, Q. Liao, H. Chen, and C. Zheng, "Consistent coding scheme for single-image super-resolution via independent dictionaries," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 313–325, 2016.

[24] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[26] H. A. Song and S.-Y. Lee, "Hierarchical representation using nmf," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 466–473.

[27] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[28] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on information Theory*, vol. 2, no. 3, pp. 80–93, 1956.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[30] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[31] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[34] G. E. Hinton, "Learning multiple layers of representation," *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[35] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1. Barcelona, Spain, 2011, p. 1237.

[36] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural networks*, vol. 32, pp. 333–338, 2012.

[37] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 693–700.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[39] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[41] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[43] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[44] ——, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[45] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[46] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2018–2025.

[47] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[48] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[49] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[50] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[51] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?" *arXiv preprint arXiv:1609.07009*, 2016.

[52] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 391–407.

[53] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, "Accurate blur models vs. image priors in single image super-resolution," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2832–2839.

[54] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[55] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[56] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, 2014, pp. 2924–2932.

[57] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[59] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[61] ——, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[62] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 550–558.

[63] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" *arXiv preprint arXiv:1702.08591*, 2017.

[64] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2017.

[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[66] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 4, 2017.

[67] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, vol. 1, no. 2, 2017, p. 3.

[68] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[69] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, no. 2, 2017, p. 3.

[70] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4470–4478.

[71] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4809–4817.

[72] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4539–4547.

[73] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[74] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[75] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[76] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.

[77] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.

[78] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5895–5907, 2017.

[79] A. Singh and N. Ahuja, "Super-resolution using sub-band self-similarity," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 552–568.

[80] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 624–632.

[81] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.

[82] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep backprojection networks for super-resolution," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[83] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[84] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[85] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," *arXiv preprint arXiv:1702.00783*, 2017.

[86] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," *arXiv preprint arXiv:1712.06087*, 2017.

[87] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 977–984.

[88] T. Michaeli and M. Irani, "Nonparametric blind super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 945–952.

[89] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *arXiv preprint arXiv:1711.10925*, 2017.

[90] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

[91] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[92] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, vol. 3, 2017, p. 2.

[93] Z. Yang, K. Zhang, Y. Liang, and J. Wang, "Single image super-resolution with a parameter economic residual-like convolutional neural network," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 353–364.

[94] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and, lightweight super-resolution with cascading residual network," *arXiv preprint arXiv:1803.08664*, 2018.

[95] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," *Computer Science*, 2015.

[96] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," *arXiv preprint arXiv:1511.05666*, 2015.

[97] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[98] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, "Learning to maintain natural image statistics," *arXiv preprint arXiv:1803.04626*, 2018.

[99] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," *arXiv preprint arXiv:1803.02077*, 2018.

[100] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" *arXiv preprint arXiv:1511.05101*, 2015.

[101] M. S. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4501–4510.

[102] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[103] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[104] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, "Generative models and model criticism via optimized maximum mean discrepancy," *arXiv preprint arXiv:1611.04488*, 2016.

[105] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.