Neural Networks 132 (2020) 394–404

Contents lists available at ScienceDirect

Neural Networks

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

Deconvolutional neural network for image super-resolution

Feilong Cao [a,*], Kaixuan Yao [b], Jiye Liang [b]

[a] *Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, Zhejiang, China*
[b] *Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education,*
School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

ARTICLE INFO

Article history:
Received 11 April 2020
Received in revised form 12 September 2020
Accepted 15 September 2020
Available online 23 September 2020

Keywords:
Deep learning
Single image super-resolution (SISR)
Convolutional neural networks (CNNs)
Deconvolutional neural networks

ABSTRACT

This study builds a fully deconvolutional neural network (FDNN) and addresses the problem of single image super-resolution (SISR) by using the FDNN. Although SISR using deep neural networks has been a major research focus, the problem of reconstructing a high resolution (HR) image with an FDNN has received little attention. A few recent approaches toward SISR are to embed deconvolution operations into multilayer feedforward neural networks. This paper constructs a deep FDNN for SISR that possesses two remarkable advantages compared to existing SISR approaches. The first improves the network performance without increasing the depth of the network or embedding complex structures. The second replaces all convolution operations with deconvolution operations to implement an effective reconstruction. That is, the proposed FDNN only contains deconvolution layers and learns an end-to-end mapping from low resolution (LR) to HR images. Furthermore, to avoid the oversmoothness of the mean squared error loss, the trained image is treated as a probability distribution, and the Kullback–Leibler divergence is introduced into the final loss function to achieve enhanced recovery. Although the proposed FDNN only has 10 layers, it is successfully evaluated through extensive experiments. Compared with other state-of-the-art methods and deep convolution neural networks with 20 or 30 layers, the proposed FDNN achieves better performance for SISR.

1. Introduction

Image super-resolution has been an important research field in general image processing and machine learning (Wang & Huang, 2009; Wang, Huang, & Xu, 2010; Zhang, Tian, Kong, Zhong, & Fu, 2020; Zhao, Glotin, Xie, Gao, & Wu, 2012), attempts have been made to enhance image resolution over the years. Single image super-resolution (SISR) aims to generate a high-resolution (HR) image from a given low-resolution (LR) image, and has become an active area in computer vision because of the increased demand for HR images in various fields.

Such developed resolution-enhancing technologies can be grouped into three main categories: (a) Interpolation-based methods (Dodgson, 1997; Parker, Kenyon, & Troxel, 1983; Zhang, Fan, Bao, Liu, & Zhang, 2018); (b) Reconstruction based methods (Jian, Xu, & Shum, 2008; Tai, Liu, Brown, & Lin, 2010; Zhang, Yang, Zhang, & Huang, 2010); and (c) Learning-based methods (He & Siu, 2011; Kui, Xiaogang, & Xiaoou, 2013; Kwang In & Younghee, 2010; Ni & Nguyen, 2007; Timofte, De and Gool, 2014; Timofte, Smet and Gool, 2014; Wang, Huang, Gong, & Pan, 2017; Yang, Wang, Lin, Cohen, & Huang, 2012; Yang, Wright, Huang, & Ma, 2010; Zhang, Wang, Li, Gao, & Xiong, 2019). Interpolation-based methods estimate the unknown pixels in the HR grid from a given LR image based on conventional interpolation approaches such as bilinear or bicubic interpolation. These interpolation-based methods are simple and intuitive, but tend to blur the detail of an image and fail to guarantee precision of the estimation. Recently, Zhang et al. (2018) constructed a rational fractal interpolation model for SISR, which achieved finer details and sharper edges. Reconstruction-based methods use some specific prior knowledge to restore the details of HR images, such as the gradient profile prior, the edge prior, or the nonlocal means prior. The third method, the learning-based method, is based on learning the relationship between LR and HR image patches from an external image pairs dataset. So far, various types of learning-based methods have been proposed, such as kernel regression (Kwang In & Younghee, 2010), support vector regression (Ni & Nguyen, 2007), and Gaussian process regression (He & Siu, 2011).

This paper continues to address the third approach. In the last few years, the sparse-coding-based method (SC) (Romano, Protter, & Elad, 2014; Yang et al., 2012, 2010), which is one of the representative learning-based methods, has been used for SISR with promising results. In fact, SC is used to learn two compact dictionaries by joint training of LR and HR image patch pairs, and restores an HR image from an LR image by assuming that the

[*] Corresponding author.
E-mail addresses: feilongcao@gmail.com (F. Cao), yaokx2@gmail.com (K. Yao), ljy@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.neunet.2020.09.017
0893-6080/© 2020 Elsevier Ltd. All rights reserved.

HR and LR image patches have the same sparse representation coefficients. Li, Dong, Xie, Shi, Wu, and Li (2018) used a novel hybrid approach toward image SR by combining model-based and learning-based approaches, which learned a parametric sparse prior of HR images from the training set (external source) and the input LR image (internal source). Zhang et al. (2019) developed a novel example regression-based SR algorithm based on a set of the learned multi-round residual regressors in a coarse-to-fine scheme and achieved promising SR results. Presently, a deep leaning method called convolutional neural networks for SISR (SRCNN), which directly learns an end-to-end mapping between LR and HR images based on a universal approximation property of feedforward neural networks has become highly popular.

In fact, deep convolutional neural networks (CNNs) have become one of the hottest technologies in recent years owing to their great success in computer vision (He, Zhang, Ren, & Sun, 2016; Krizhevsky, Sutskever, & Hinton, 2012). The important components of CNNs are the convolution operation, pooling operation, and rectified linear unit (ReLU) activation function. As a main key in CNNs, the convolution operation extracts features, and the extracted features become increasingly abstract as the number of network layers deepens (Erhan, Bengio, Courville, & Vincent, 2009; Zeiler, Taylor, & Fergus, 2011). Many studies (Yosinski, Clune, Nguyen, Fuchs, & Lipson, 2015; Zeiler & Fergus, 2014) have shown that CNNs represent a good tool for image classification tasks owing to the excellent performance of the convolution operation mentioned above.

Recently, many studies have adopted CNNs for SISR and have achieved promising results. Dong, Loy, He and Tang (2016) first introduced use of a CNN in SISR. They proposed a convolutional neural network for SISR (SRCNN) that directly learns an end-to-end mapping between LR and HR images. Since then, additional deep CNN-based SISR methods (Dong, Loy and Tang, 2016; Kim, Lee, & Lee, 2016a, 2016b; Liu et al., 2019; Mao, Shen, & Yang, 2016; Shi, Caballero, Huszar, Totz, Aitken, Bishop, Rueckert, & Wang, 2016; Yang et al., 2017) have been proposed, and significantly outperform classic non-deep learning SISR methods. These deep CNN-based SISR methods, however, mainly rely on increasing the depth of the network (Dong, Loy and Tang, 2016; Kim et al., 2016a) or embedding more complex structures (Kim et al., 2016b; Liu et al., 2019; Mao et al., 2016; Song, Chowdhury, Yang, & Dutta, 2020; Yang et al., 2017) to improve the reconstruction performance of SRCNN, such as use of a deeply-recursive convolutional network (Kim et al., 2016b), and encoder–decoder network (Liu et al., 2019; Mao et al., 2016), recurrent residual network (Yang et al., 2017). Here we omit a comprehensive review of the existing models. Interested readers can refer to survey papers (Wang, Chen, & Hoi, 2020) for more details.

No matter how deep or complex the CNN structure proposed, the main constituent elements of these deep CNN-based SISR methods remain the convolution operations. Clearly, these deep CNN-based SISR methods are mainly transplanted and imitated based on CNNs used for image classification. Generally, in deep CNNs, convolution operations can extract principal features and become increasingly abstract as the depth of the network increases, which are very effective for image classification. As mentioned above, successive multiple convolution operations can extract abstract features; however, the convolution operation may also lose some image detail information during the process, which is, in fact, a process of information from more to less. That is, it is a process in which subordinate features are successively discarded during successive multiple convolution operations. Thus, multiple convolution operations in CNNs are highly suitable for image classification.

However, image reconstruction is a process of information from less to more. In SISR, all LR information should be exploited to recover more detailed information. Contrary to a convolution operation, a deconvolution operation can obtain more abundant information from a few features. We will provide several comparison experiments between convolution and deconvolution operations in Section 4 to support our statements. Thus, using deconvolution operations in multilayered feedforward neural networks (MFNNs) for SISR should be the more appropriate choice.

Inspired by this, this paper proposes a deep fully deconvolutional neural network (FDNN) for SISR. That is, the proposed network only contains deconvolutional layers and directly learns an end-to-end mapping from LR to HR images. The main reason is that a deconvolution operation can restore more detailed information from LR pixels. Unlike most deep CNN-based SISR methods, the proposed FDNN does not require more depth or embedded complex structures in the network; it uses only 10 deconvolution layers. Although the proposed FDNN has only 10 layers, its structure is very simple and its performance for SISR is better than deep CNN-based SISR networks containing 20 or 30 layers, such as VDSR (Kim et al., 2016a), DEGREE (Yang et al., 2017), and RED30 (Mao et al., 2016). An overview on the proposed architecture is shown in Fig. 1.

Furthermore, we will improve both the ReLU activation function and mean squared error (MSE) loss function in the proposed FDNN. Usually, ReLU is the most popular activation function in deep CNNs because it can alleviate the problem of gradient vanishing. However, it can lead to the "Dying ReLU" problem when CNNs are trained with stochastic gradient descent because ReLU is non-negative. Therefore, we will adopt an exponential linear unit (ELU) (Clevert, Unterthiner, & Hochreiter, 2016) as the activation function in the proposed FDNN. Moreover, although most deep CNN-based SISR methods have adopted the MSE loss function (as most image classification CNNs have), the MSE may blur the details of the reconstructed image. To overcome this oversmoothness caused by MSE loss function, we add the Kullback–Leibler divergence into the loss function.

The main contributions of this paper can be briefly described as follows.

- A fully deconvolutional neural network (FDNN) is proposed for SISR, which only contains deconvolutional layers and directly learns an end-to-end mapping from LR to HR images. Although the proposed FDNN has only 10 deconvolution layers, it outperforms deeper, more complex existing CNNs for SISR.
- To our knowledge, it is the first time to add the Kullback–Leibler divergence into the loss function to achieve a better reconstruction effect and avoid the oversmoothness caused by MSE loss.
- In the first layer of the proposed network, an 1 × 1 deconvolution layer with ELU activation is used as a non-linear enhancement module to regenerate additional non-linear information from the low-resolution pixels.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 introduces the proposed FDNN in detail. In Section 4, several experiments are presented to evaluate the performance of the proposed FDNN. The conclusion and future work are presented in Section 5.

2. Related work

The main purpose of SISR is to determine the relationship between LR and HR images. Generally, this relationship can be formulated as

$$y = \psi(x * k + n), \tag{1}$$

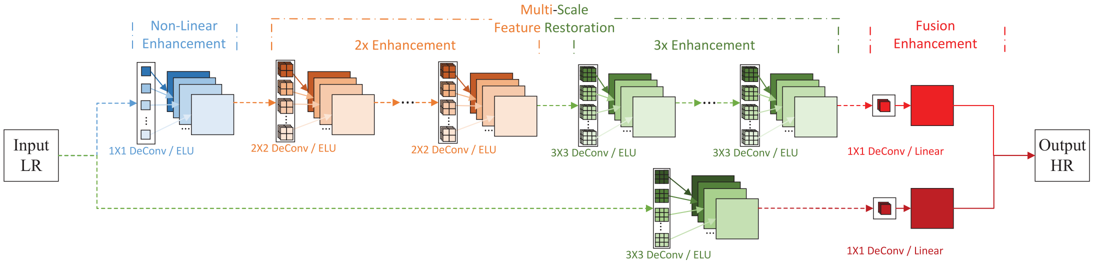


Fig. 1. Network structure of the proposed FDNN for a scale factor of 3. The model is mainly composed of three parts: non-linear enhancement, multi-scale feature restoration, and fusion enhancement. The lower channel can be regarded as the residual learning part, which is composed of one 3 × 3 deconvolution layer and one 1 × 1 deconvolution layer to make the residual feature size the same as the output.

where $\psi(\cdot)$ denotes a nonlinear compression operator, k is the convolution kernel, n is the noise, y is the LR image, and x denotes the ground truth HR image. The Eq. (1) could be further simplify as

$$y = Hx, \tag{2}$$

where H is a degradation matrix that represents a down-sampling operator. Owing to the insufficient condition of the ill-posed inverse problem represented by SISR, we cannot restore x easily by

$$x = H^{-1}y. \tag{3}$$

Recently, various deep CNNs have been developed to learn an end-to-end mapping that restores the desired HR image from an LR image based on

$$\hat{x} = F(y), \tag{4}$$

where $\hat{x}$ denotes the estimate of the ground truth HR image x. These deep CNNs aim to train a network $F(\cdot)$ that minimizes

$$\frac{1}{N}\sum_{i=1}^{N}(F(y_i) - x_i)^2, \tag{5}$$

where N is the number of the training image pairs.

Dong, Loy, He et al. (2016) firstly proposed a CNN to solve SISR (SRCNN), which consists of three convolution layers corresponding to three operations: patch extraction and representation, non-linear mapping, and reconstruction. Therefore, similar to Eq. (4) SRCNN is a composition of three functions:

$$\hat{x} = F_{\mathrm{SRCNN}}(y) = F_3(F_2(F_1(y))), \tag{6}$$

where F_i denotes the ith convolution layer, which is expressed as a mapping operation.

To achieve a better restoration performance, Kim et al. (2016a, 2016b) developed two very deep CNNs for SISR: a very deep super-resolution (VDSR) convolutional network (Kim et al., 2016a) and a deeply-recursive convolutional network (DRCN) (Kim et al., 2016b), each comprising 20 convolution layers. These depth-increased convolution networks can be formulated as

$$\hat{x} = F_d(\cdots F_3(F_2(F_1(y)))), \tag{7}$$

where d denotes the depth of the deep CNN.

VDSR adopted residual learning and adjustable gradient clipping to accelerate the convergence speed of the network. To reduce the number of parameters of the network and ease the difficulty of training, DRCN used recursive-supervision and skip-connection techniques and achieved promising results. Although VDSR and DRCN achieved impressive performances, both are composed solely of convolution layers. The size of the input LR images is reduced during the convolution operations. Thus, SRCNN, VDSR, and DRCN must increase the size of the input LR images before training, which increases the computational complexity. Inspired by this, Mao et al. (2016) proposed a residual encoder–decoder network with 30 layers (RED30) for SISR, which consists of a chain of symmetric convolution layers and deconvolution layers. Owing to its symmetry, RED30 also has to increase the size of LR images before the first layer to keep the size of the input image consistent with the output image. Liu et al. (2019) proposed an end-to-end multi-scale deep encoder–decoder with edge map guidance for SISR. In their model, image and the corresponding edge maps were simultaneously fed into the pipeline. And along the multiple streams, convolution–deconvolution responses with different scales were concatenated to generate the final reconstructed image. Dong, Loy and Tang (2016) proposed a 9-layer fast super-resolution convolutional neural network (FSRCNN), which contains 8 convolution layers and a final deconvolution layer. Shi et al. (2016) proposed an efficient sub-pixel convolutional neural network (ESPCN) and used an efficient sub-pixel convolution layer as the last layer of their network. Both FSRCNN and ESPCN added an upscaling operation at the end of the network as part of the training process.

With the increasing depth of CNNs, the ReLU activation function has been widely used in deep neural networks to eliminate the gradient vanishing problem. However, training based on stochastic gradient descent may suffer from the dying ReLU problem, where a unit dies (it only outputs 0 for any given input). Dong, Loy and Tang (2016) used the parametric rectified linear unit (PReLU) (He, Zhang, Ren, & Sun, 2015) as the activation function in FSRCNN instead of the commonly-used ReLU. Unlike ReLU, the coefficient of the PReLU negative part is not zero and is adaptively learned.

Most deep CNN-based SISR methods have adopted the MSE as the loss function. However, the MSE can result in the loss of high frequency details because of its oversmoothness. To overcome this drawback, Yang et al. (2017) proposed a deep edge guided recurrent residual network (DEGREE) to recover HR images with sharp high frequency details by modeling the edge priors.

A deconvolution network was first introduced by Zeiler, Krishnan, Taylor, and Fergus (2010), and has been successfully applied to visualize neural network layers by generating representative images in feature space (Zeiler & Fergus, 2014; Zeiler et al., 2011). In FSRCNN, a deconvolution operation was first introduced into SISR. However, the deconvolution operation is only used in the final layer, while the remaining layers are all convolution layers. The main mechanism of FSRCNN is thus the convolution operation, and the deconvolution operation is only applied as a pixel amplifier.

3. Fully deconvolutional neural network

3.1. Model structure

Our primary objective is to improve the performance of the neural network for SISR by choosing an appropriate internal algorithm instead of simply increasing the network depth or embedding complex structures into the network. Our proposed FDNN

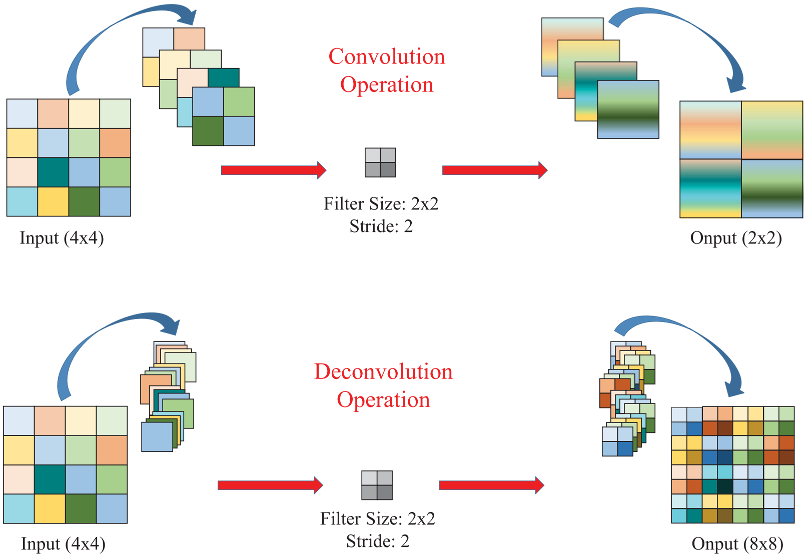


Fig. 2. Illustration of convolution and deconvolution operations.

model, as shown in Fig. 1, primarily comprises three parts: non-linear enhancement, multi-scale feature restoration, and fusion enhancement. FDNN is a simple forward-type network model and consists of only 10 deconvolution layers.

3.1.1. *Non-linear enhancement*

The non-linear enhancement module contains one deconvolution layer whose filter size is 1 × 1, and the 1 × 1 deconvolution operation is equivalent to an 1 × 1 convolution operation. The one by one convolution operation was first introduced by Lin et al. (Lin, Chen, & Yan, 2014) and was further developed by Szegedy et al. in GoogLeNet (Szegedy et al., 2015). When the stride is equal to 1, the one by one convolution or deconvolution operation can be formulated as

$$\hat{H}_i = w_i H, \tag{8}$$

where w_i is a real number and denotes the ith filter weight, H denotes one input feature map of the previous layer, and the size of H_i is equal to that of H.

Suppose the shape of the input tensor is (l_w, l_h, l), where (l_w, l_h) represents the spatial dimensions of a feature map, and l is the number of feature maps. After the (l_w, l_h, l) tensor is fed into an 1 × 1 convolution or deconvolution layer with $\hat{l}$ filters, the output tensor of the 1 × 1 layer will have the shape $(l_w, l_h, \hat{l})$. Thus, the 1 × 1 convolution or deconvolution layer can be used to change the dimensionality in filter space. If $\hat{l} > l$, the 1 × 1 filter increases the dimensionality, but if $\hat{l} < l$, it reduces dimensionality. For GoogLeNet (Szegedy et al., 2015) and Network in Network (Lin et al., 2014), the 1 × 1 convolution operation was used to reduce the channel dimensionality, while the 1 × 1 deconvolution operation in our FDNN is used to increase dimensionality.

From Eq. (8), the 1 × 1 convolution or deconvolution operation is strictly linear, but we add a non-linear ELU activation layer after the 1 × 1 deconvolution layer. In the non-linear enhancement block, the input LR image is first fed into an 1 × 1 deconvolution layer with 64 filters to generate more features from the original image. Obviously, the dimensionality of the channel (feature map) increases from 1 to 64, which could capture more information from the original input image. Then, these features are mapped by a non-linear ELU activation function to increase nonlinearity.

3.1.2. *Multi-scale feature restoration*

Presently, the convolution operation has been widely used in deep learning because of its great success in image classification, and many studies have transplanted CNNs into SISR. However, these CNN-based SR methods rarely analyze if the convolution operation is reasonable for the mechanism of SISR. Most of them just simply transferred the CNN model to SISR from image classification tasks. Therefore, it is necessary to analyze the mechanism of SISR and the difference between convolution operation and deconvolution operation in SISR.

Supposing that a $w \times w$ image is processed by a $k \times k$ convolution, then the size of the output feature map can be computed by Eq. (9):

$$\hat{w} = (w - k)/s + 1, \tag{9}$$

where s denotes the convolution stride, $\hat{w}$ denotes the width of the feature map after convolution, and $\hat{w} \leq w$. In essence, a convolution operation is a process for feature compression and extraction, and it only extracts the principal features and may lose some detailed information in the compression process. Conversely, a deconvolution operation can reconstruct additional information from few features. The size of the feature map recovered by deconvolution can be computed by Eq. (10):

$$\hat{w}' = (w - 1)s + k, \tag{10}$$

where $\hat{w}'$ is the width of the feature map after deconvolution, and it is clear that $\hat{w}' \geq w$.

A detailed illustration of the convolution and deconvolution operations is shown in Fig. 2, where the top panel shows the convolution operation and the bottom panel shows the deconvolution operation. In both panels, the input image is 4 × 4, and the size of the convolution and deconvolution filters are 2 × 2. The size of the convolution output results in a 2 × 2 image, while deconvolution results in an 8 × 8 image. Obviously, the convolution operation is not consistent with the mechanism of SISR which aims to recover the high resolution images from low resolution images. Although the padding operation before convolution could increase the size of output image, while extra white pixels would be generated. On the contrary, the deconvolution operation could increase the size of image which is more consistent with the mechanism of SISR.

For the relationship between LR and HR images Eq. (1), we consider a simple linear degradation model (Xu, Ren, Liu, & Jia, 2014)

$$y = x * k. \tag{11}$$

According to the convolution theorem (Bracewell, 2002), the spatial convolution can be transformed to a frequency domain multiplication

$$\mathcal{F}(y) = \mathcal{F}(x) \cdot \mathcal{F}(k), \tag{12}$$

where $\mathcal{F}(\cdot)$ is the Fourier transform, and $\cdot$ denotes the element-wise multiplication. Then, in the Fourier domain, x could be expressed as

$$x = \mathcal{F}^{-1}(\mathcal{F}(y)/\mathcal{F}(k)) = \mathcal{F}^{-1}(1/\mathcal{F}(k)) * y, \tag{13}$$

where $\mathcal{F}^{-1}(\cdot)$ denotes the inverse Fourier transform, and $*$ denotes the convolution operation. Thus, the ground truth image could be recovered from low resolution image y by a pseudo inverse convolution kernel

$$x = k^{\dagger} * y, \tag{14}$$

where "$^{\dagger}*$" is the deconvolution operation, and $k^{\dagger} = \mathcal{F}^{-1}(1/\mathcal{F}(k))$ denotes the deconvolution kernel.

Generally, the deconvolution kernel $k^{\dagger}$ is difficult to obtain. Thus, we build a multi-layers deconvolutional neural network FDNN to learn these deconvolution kernels. Unlike most deep CNN-based SISR methods, we use only deconvolution layers in our FDNN for image super-resolution. As shown in Fig. 1, the multi-scale feature restoration block is composed of several deconvolution layers, which is primarily responsible for progressive recovery of image details in multiple steps. Furthermore, an image contains various features, while a single scale of deconvolution operation cannot significantly restore all types of features. Thus, the multi-scale feature restoration block contains different scales of deconvolution operations to restore different types of high-resolution features. For an up-scale factor of N ($N \geq 2$), the multi-scale feature restoration block would consist of $N - 1$ sub-blocks to restore different up-scale level pixel features. As shown in Fig. 1, for an up-scale factor of 3, the multi-scale feature enhancement block contains two sub-blocks: 2× enhancement and 3× enhancement.

3.1.3. Fusion enhancement

As described above, the multi-scale feature enhancement block learns high-resolution features with different scale deconvolutional kernels and each kernel learns one type of features. To fuse all types of high-resolution features, in the fusion enhancement block, we use an 1×1 deconvolution layer containing one kernel to achieve image feature fusion. Here, the 1×1 deconvolution kernel is used to keep the scale-invariance of the last layer in the multi-scale feature enhancement block. First, the HR features restored by the multi-scale feature restoration block are fused by the 1×1 deconvolution operation:

$$x^j = \sum_{i=1}^{n^j} c_i^j z_i^j, \tag{15}$$

where z_i^j is the ith feature map restored by the multi-scale feature restoration block, $j = 1$ denotes the top channel, $j = 2$ denotes the bottom channel (or residual channel) as shown in Fig. 1, c_i^j denotes the 1×1 deconvolution filter weight, n^j is the number of feature maps to be restored by the multi-scale feature restoration block, and x^j denotes the jth channel fusion feature.

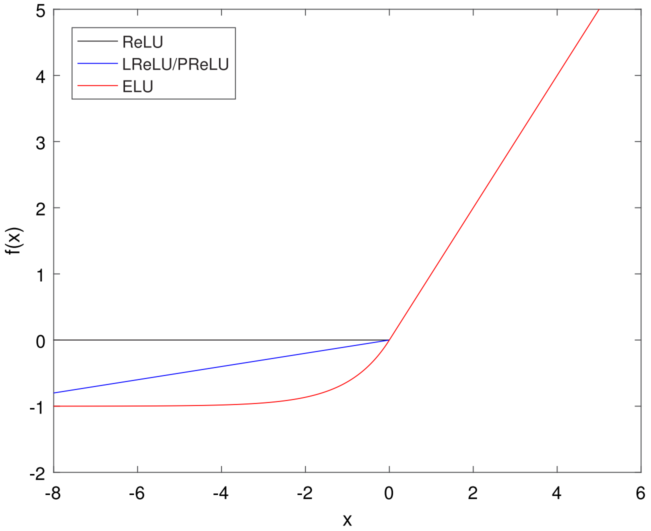


Fig. 3. Shapes of different activation functions: ReLU, LReLU/PReLU (LReLU is an exception of PReLU when the coefficient of the negative part is fixed at 0.1), and ELU for $\alpha = 1.0$.

Next, the top channel reconstructed HR feature map and the residual channel reconstructed HR feature map are combined into the final reconstructed HR image as follows:

$$\hat{x} = \sum_{i=1}^{2} x_i. \tag{16}$$

3.2. Exponential linear unit activation function

The commonly-used activation function in DNN-based SISR models is rectified linear unit (ReLU) (Nair & Hinton, 2010), however, the ReLU activation may generate "dead features" (Zeiler & Fergus, 2014) caused by zero gradients. Several improved ReLU methods such as leaky ReLU (LReLU), PReLU, and the exponential linear unit (ELU) (Clevert et al., 2016) are provided to address this problem, which use the identity for positive values to avoid the gradient vanishing problem and zero gradients. Owing to their negative parts, LReLU and PReLU are represented as a slope, and they do not ensure a noise-robust deactivation state, while ELU can decrease the propagated variation and information by saturating to a negative value with smaller input values. Fig. 3 shows the shapes of these activation functions. Thus, we use the ELU to replace the commonly-used ReLU. Formally, ELU is defined as

$$f(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha(\exp(x) - 1) & \text{if } x < 0. \end{cases} \tag{17}$$

Then

$$f'(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ f(x) + \alpha & \text{if } x < 0. \end{cases} \tag{18}$$

where the α is a scalar to control the slope of negative section and it is depicted as 1.0 in our experiments.

3.3. Kullback–Leibler divergence loss

Most deep CNN-based SISR methods use the MSE as the loss function, but the MSE can lead to the loss of information on image details due to its oversmoothness. To avoid this shortcoming, we add Kullback–Leibler divergence (KL divergence) into the final loss function. KL divergence is a non-symmetric measure of the difference between two probability distributions. It can calculate exactly how much information is lost when one distribution approximates another. Formally, let p be the original distribution (original HR image) and q be the approximating distribution (reconstructed HR image), then the KL divergence can be written as

$$D_{\mathrm{KL}}(p|q)=\sum_{i=1}^{M}p(x_i)\cdot\log\frac{p(x_i)}{q(x_i)}, \tag{19}$$

where M denotes the total number of pixels.

Let $\{L_i\}_{i=1}^{N}\in\mathbb{R}^{l_m\times l_n}$ be the LR training image set, and $\{H_i\}_{i=1}^{N}\in\mathbb{R}^{h_m\times h_n}$ be the HR training image set, where N is the total number of training image pairs. Denoting our proposed model by F_{FDNN}, then the reconstructed HR image by the FDNN is expressed as $F_{\mathrm{FDNN}}(L_i,\theta)$, where θ is the network parameter. The final loss function is a combination of MSE and KL divergence, which is given by

$$\begin{aligned}\text{Loss}=&\frac{1}{N}\sum_{i=1}^{N}\|F_{\mathrm{FDNN}}(L_i,\theta)-H_i\|_F\\&+\frac{1}{N}\sum_{i=1}^{N}\|D_{\mathrm{KL}}(H_i)F_{\mathrm{FDNN}}(L_i,\theta)\|_F\ ,\end{aligned} \tag{20}$$

where $\|\cdot\|_F$ is the Frobenius norm. Combining Eq. (19) with Eq. (20), we obtain

$$\begin{aligned}\text{Loss}=&\frac{1}{N}\sum_{i=1}^{N}\|F_{\mathrm{FDNN}}(L_i,\theta)-H_i\|_F\\+&\frac{1}{N}\sum_{i=1}^{N}\left\|\sum_{j=1}^{M}H_i(x_j)\log\frac{H_i(x_j)}{F_{\mathrm{FDNN}}(L_i,\theta)(x_j)}\right\|_F\ .\end{aligned} \tag{21}$$

In our loss function, the MSE loss aims to reduce the error between the predicted image set and ground truth high resolution image set. On the other hand, the KL loss could ensure the pixel distribution of predicted image is as close as possible to the ground truth high resolution image to avoid the distortion phenomenon caused by the oversmoothness of MSE. To our knowledge, it is the first time to introduce the KL divergence into SISR. To evaluate the effectiveness of KL divergence, we compare VDSR under the standard MSE loss with MSE-KL loss. For the sake of fairness, we rebuild VDSR with 10 layers and train it with MSE loss and KL divergence based loss. It is important to note that the implementation details of the VDSR10 under MSE loss are completely consistent with VDSR10 under the MSE-KL loss. They are both trained over 100 epochs with batch size of 64, and the optimization algorithm is Adam (Kingma & Ba, 2015). The training dataset is composed of 291 images without data augmentation. The experimental results are presented in Table 1. Obviously, the VDSR10 under MSE-KL loss achieves better performance than VDSR10 under MSE loss.

4. Experiments

In this section, several experiments are presented to evaluate the performance of our proposed method. The datasets used for training and testing are introduced first. Next, some implementation details are described. Finally, we compare our method with several state-of-the-art SISR methods. Experiments in this paper are carried out using one GPU (GeForce GTX 1050 ti) and an Intel CORE i7 with 16 GB RAM memory system. The training phase is performed with Keras (Chollet et al., 2015) under the Tensorflow framework, CUDA9 (Nickolls, Buck, Garland, & Skadron, 2008), and cuDNN6 (Chetlur et al., 2014). The test phase is performed using Matlab 2016a.

Table 1
The PSNR results of VDSR10 with MSE loss and MSE-KL loss under different datasets.

Dataset	Scale	VDSR10 + MSE	VDSR10 + MSE-KL
Set5	2 ×	36.65	36.96
	3 ×	32.83	32.88
	4 ×	30.40	30.53
Set14	2 ×	32.41	32.60
	3 ×	29.29	29.32
	4 ×	27.40	27.48
B100	2 ×	31.44	31.55
	3 ×	28.49	28.50
	4 ×	26.94	26.96

4.1. Datasets and implementation details

4.1.1. Training datasets

For benchmarking, we use 291 images to train our proposed model as most of the comparison SISR methods have done. The training dataset consists of two parts: the first contains 91 images from Yang et al. (2010), and the other contains 200 images from the Berkeley Segmentation Dataset (Martin, Fowlkes, Tal, & Malik, 2001). It should be noted that data augmentation (rotation and flip) is used in our experiments. For fair comparison, we do not train our models with a larger dataset and all methods use the same training dataset as described above.

4.1.2. Test dataset

For the test dataset, we compare our proposed method with state-of-the-art SISR methods on three popular benchmark datasets: Set5 (Bevilacqua, Roumy, Guillemot, & Alberimorel, 2012), Set14 (Zeyde, Elad, & Protter, 2010), and BSD100 (Martin et al., 2001) with scaling factors of 3 and 4. The three benchmark datasets consist of 5, 14, and 100 images, respectively.

4.1.3. Evaluation metrics

We use the widely used the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) (Wang, Bovik, Sheikh, Simoncelli, et al., 2004) to evaluate the performance of different methods, which are defined as

$$\text{PSNR}=20\log_{10}\frac{255}{\text{RMSE}},\quad \textit{where}\quad \text{RMSE}=\sqrt{\frac{1}{mn}\|\hat{x}-x\|_F^2}, \tag{22}$$

and

$$\text{SSIM}(x,\hat{x})=l(x,\hat{x})c(x,\hat{x})s(x,\hat{x}), \tag{23}$$

where

$$\begin{cases}l(x,\hat{x})=\frac{2\mu_x\mu_{\hat{x}}+C_1}{\mu_x^2+\mu_{\hat{x}}^2+C_1}\\ c(x,\hat{x})=\frac{2\sigma_x\sigma_{\hat{x}}+C_2}{\sigma_x^2+\sigma_{\hat{x}}^2+C_2}\\ s(x,\hat{x})=\frac{\sigma_{x\hat{x}}+C_3}{\sigma_x\sigma_{\hat{x}}+C_3}\end{cases} \tag{24}$$

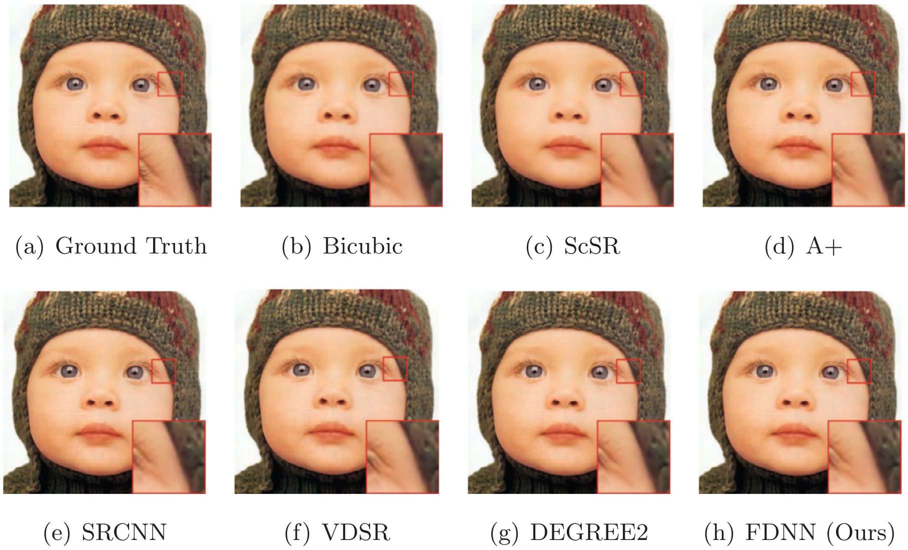
respectively. Here x denotes the ground truth high resolution image, and $\hat{x}$ denotes the predicted image. The positive constants C_1, C_2 and C_2 are used to avoid a null denominator

4.1.4. Training samples

The 291 original ground truth images are first processed by data augmentation (rotation and flip) to obtain the HR training image set $\{H_i\}_{i=1}^{N}\in\mathbb{R}^{h_m\times h_n}$. Then we use the desired scaling factor s to down-sample the HR training image set $\{H_i\}_{i=1}^{N}\in\mathbb{R}^{h_m\times h_n}$ to form the corresponding LR training image set $\{L_i\}_{i=1}^{N}\in\mathbb{R}^{l_m\times l_n}$, where $h_m=sl_m$ and $h_n=sl_n$.

Table 2
Comparison results of the proposed FDNN (10 layers) with Bicubic, A+, SRCNN, VDSR (20 layers) DEGREE2 (20 layers), and ESCN for a scale factor 3 on dataset Set5. The bold numbers denote the best performance and the underlined numbers indicate the second-best performance.

Set5	Criterion	Bicubic	A+	SRCNN	VDSR	DEGREE2	ESCN	FDNN
baby	PSNR	33.92	35.17	35.25	35.39	35.34	35.35	**35.41**
	SSIM	0.9033	0.9228	0.9241	0.9269	0.9249	0.9262	**0.9270**
bird	PSNR	32.58	35.76	35.48	36.67	36.37	36.22	**36.75**
	SSIM	0.9264	0.9563	0.9549	**0.9649**	0.9625	0.9617	0.9648
butterfly	PSNR	24.03	27.35	27.95	**29.95**	29.49	28.87	29.87
	SSIM	0.8232	0.9112	0.9098	**0.9428**	0.9369	0.9300	0.9410
head	PSNR	32.88	33.73	33.71	**33.97**	33.88	33.92	33.94
	SSIM	0.7996	0.8271	0.8272	**0.8341**	0.8298	0.8319	0.8335
woman	PSNR	28.56	31.30	31.37	32.35	31.87	32.01	**32.37**
	SSIM	0.8902	0.9290	0.9291	**0.9409**	0.9368	0.9369	0.9403
Average	PSNR	30.39	32.59	32.75	33.66	33.39	33.28	**33.68**
	SSIM	0.8682	0.9088	0.9090	**0.9213**	0.9182	0.9173	**0.9213**


(a) Ground Truth (b) Bicubic (c) ScSR (d) A+

(e) SRCNN (f) VDSR (g) DEGREE2 (h) FDNN (Ours)

Fig. 4. Reconstruction results of "Baby" (Set5) with scale factor 3.

4.1.5. Implementation details

As shown in Fig. 1, our proposed FDNN consists of three parts and all layers are deconvolution layers. The FDNN model is designed with 10 layers, where the nonlinear enhancement and fusion enhancement are both composed of one 1 × 1 deconvolution layer. For a scale factor of 3, the multi-scale feature restoration block consists of three 2 × 2 and five 3 × 3 deconvolution layers. The last layer contains one filter and the remaining layers all contain 64 filters. For a scale factor of 4, the multi-scale feature restoration block consists of two 2 × 2, two 3 × 3 deconvolution layers, and four 4 × 4 deconvolution layers. Analogously, the last layer contains one filter and the remaining layers all contain 64 filters.

For filter weight initialization, we use the Glorot uniform initializer (Glorot & Bengio, 2010) for each deconvolution filter. It randomly extracts samples from a uniform distribution [−limt, limit], where the limit is $\sqrt{6}/\sqrt{n_i + n_{i+1}}$, n_i denotes the number of input units in the weight tensor, and n_{i+1} denotes the number of output units in the weight tensor. The Glorot uniform initializer is formulated as

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}, \frac{\sqrt{6}}{\sqrt{i + n_{i+1}}}\right]. \tag{25}$$

We use the Adam optimization algorithm (Kingma & Ba, 2015) to optimize our proposed model. Stochastic gradient descent (SGD) maintains a single learning rate (termed alpha) for all weights update and the learning rate does not change during training, while Adam estimates the first and second moments of the gradients to compute individual adaptive learning rates for different parameters. The parameters used for Adam follow those provided in the original paper (Kingma & Ba, 2015): learning rate is setting to 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

4.2. Comparisons to state-of-the-art methods

To evaluate the performance of our proposed FDNN, we compare FDNN with several state-of-the-art SISR methods:

- A+ - adjusted anchored neighborhood regression (Timofte, Smet et al., 2014);
- SRCNN — convolutional neural network for SISR (Dong, Loy, He et al., 2016);
- VDSR — very deep CNN for SISR (20 layers) (Kim et al., 2016a);
- DEGREE2 — deep edge guided recurrent residual network (20 layers) (Yang et al., 2017);
- ESCN — ensemble based sparse coding network (Wang et al., 2017);
- SML — parametric sparse model learning (Li et al., 2018);
- MSDEPC - multi-scale deep encoder–decoder with edge map guidance (24 layers) (Liu et al., 2019).

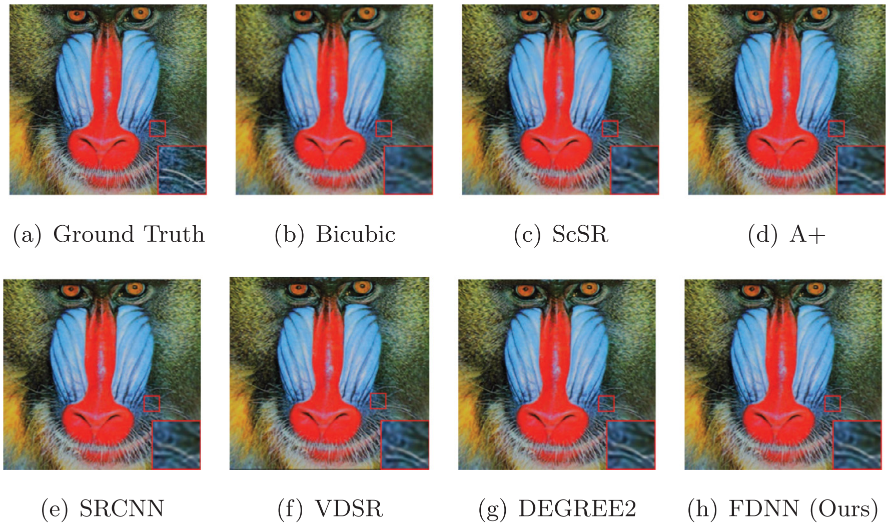

(a) Ground Truth (b) Bicubic (c) ScSR (d) A+
(e) SRCNN (f) VDSR (g) DEGREE2 (h) FDNN (Ours)

Fig. 5. Reconstruction results of “baboon" (Set14) with scale factor 3.

Table 3
Average comparison results of the proposed model (10 layers) with Bicubic, ScSR, A+, SRCNN, VDSR (20 layers), DEGREE2 (20 layers), ESCN, SML, and MSDEPC (24 layers). The bold numbers denote the best performance and the underlined numbers indicate the second-best performance.

Dataset		Set5		Set14		B100	
Method	Criterion	×3	×4	×3	×4	×3	×4
Bicubic	PSNR	30.39	28.42	27.55	26.00	27.21	25.96
	SSIM	0.8682	0.8104	0.7742	0.7027	0.7385	0.6675
A+	PSNR	32.58	30.28	29.13	27.32	28.29	26.82
	SSIM	0.9088	0.8603	0.8188	0.7491	0.7835	0.7087
SRCNN	PSNR	32.75	30.48	29.28	27.49	28.41	26.90
	SSIM	0.9090	0.8628	0.8209	0.7503	0.7863	0.7101
VDSR	PSNR	33.66	31.35	29.77	28.01	28.82	27.29
	SSIM	0.9213	0.8838	0.8314	0.7674	**0.7976**	**0.7251**
DEGREE2	PSNR	33.39	31.03	29.61	27.73	28.63	27.07
	SSIM	0.9182	0.8761	0.8275	0.7597	0.7916	0.7177
ESCN	PSNR	33.28	31.02	29.51	27.75	28.58	27.11
	SSIM	0.9173	0.8774	0.8264	0.7611	0.7917	0.7197
SML	PSNR	33.50	31.26	29.58	27.76	28.55	27.06
	SSIM	0.9175	0.8791	0.8262	0.7593	0.7887	0.7155
MSDEPC	PSNR	**33.70**	**31.41**	29.78	**28.02**	**28.88**	**27.30**
	SSIM	**0.9225**	**0.8836**	0.8319	**0.7679**	0.7974	0.7249
FDNN(Ours)	PSNR	33.68	31.17	**29.80**	27.93	28.80	27.22
	SSIM	0.9213	0.8813	**0.8320**	0.7651	**0.7976**	0.7236

All images are down-sampled by the same bicubic kernel and the results are implemented by publicly available codes from the authors or from their original papers. It should be noted that some compared methods do not recover the image border; these methods need to crop some pixels near image boundaries during test processing. Our proposed method does not have to crop the image border; FDNN can reconstruct the full-sized image. However, for fair comparison, we also crop the boundary pixels during test processing.

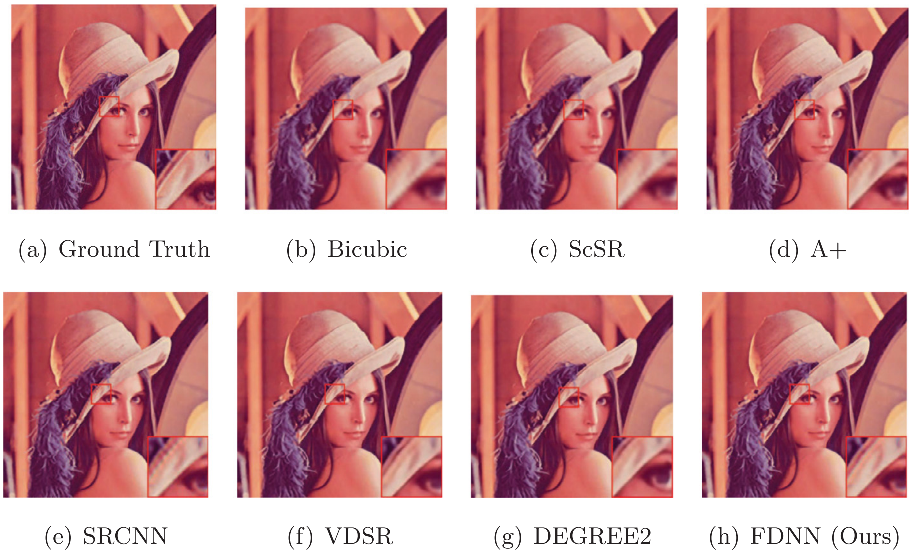
4.2.1. Objective evaluation

In Table 2, we provide the detailed results with a scale factor of 3 on Set5, respectively. Although our proposed FDNN contains only 10 deconvolution layers, its reconstruction performance is better than those CNN-based SISR methods with 20 layers. Table 3 provides a summary of the quantitative evaluations on Set5, Set14, and B100.

Moreover, an ablation study is implemented to further show that the deconvolution operation is more suitable for the mechanism of SISR than convolution operation. We build several mixed models of convolutional and deconvolutional for SISR, the total number of layers of these models are the same as our FDNN. It should be noted that the input image and output image of these mixed models have the same size and the padding form is set to “same" during convolution, as VDSR did. In addition to these, the training dataset and implementation details are also the same as our FDNN for the fair of the comparison experiment. The experimental results are presented in Table 4, which show that the performance increases with the number of deconvolutional layers.

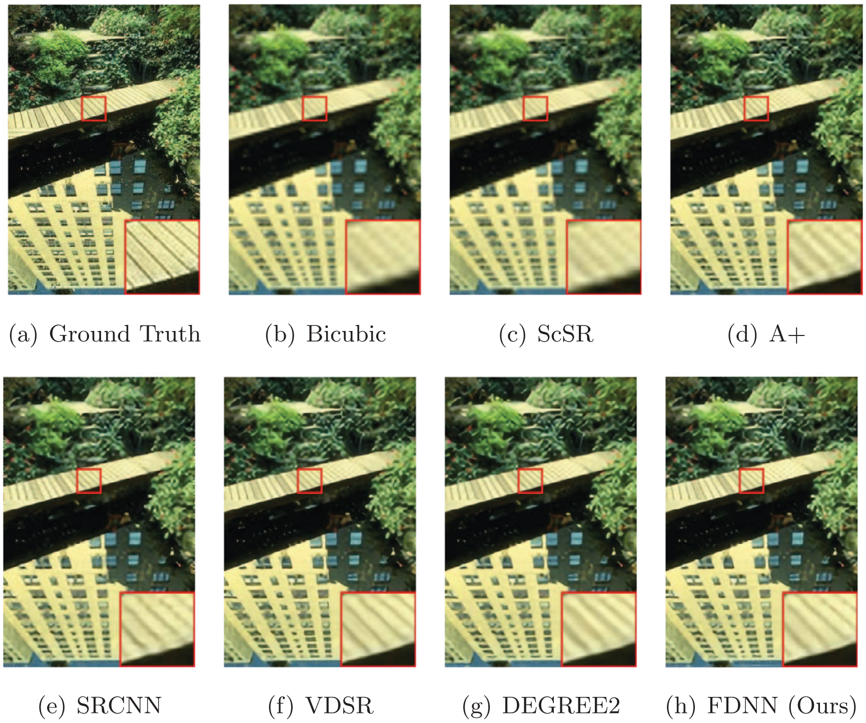
4.2.2. Visual evaluation

In order to investigate the performance of the different methods in terms of visual quality, we present some visual results in Figs. 4, 5, 6, and 7. From these figures, we can observe that the reconstructed results of CNN-based SISR methods are far superior to traditional learning-based SISR methods. For the image “148026" from B100 in Fig. 7, which contains much texture information, the images reconstructed by ScSR and A+ are slightly blurry, while the SRCNN and VDSR generated images are clearer than ScSR and A+.


(a) Ground Truth (b) Bicubic (c) ScSR (d) A+

(e) SRCNN (f) VDSR (g) DEGREE2 (h) FDNN (Ours)

Fig. 6. Reconstruction results of “lenna" (Set14) with scale factor 4.


(a) Ground Truth (b) Bicubic (c) ScSR (d) A+

(e) SRCNN (f) VDSR (g) DEGREE2 (h) FDNN (Ours)

Fig. 7. Reconstruction results of “148026" (B100) with scale factor 4.

However, because of the MSE loss function used in SRCNN and VDSR, the images generated by these two CNN-based methods are too smooth to reconstruct sharp details. By introducing the Kullback–Leibler divergence loss into our proposed FDNN, the image reconstructed by our method produces more natural sharp details.

5. Conclusion

This study proposes a fully deconvolutional neural network (FDNN) and uses FDNN to reconstruct HR images. Deeper CNN-based methods have been introduced into SISR with promising results, and have become a widely used option for SISR in recent years. However, the majority of the previous deep CNN-based SISR models have focused on increasing the network depth or embedding complex structures, which result in deeper and more complex networks.

Our main objective is to improve the performance of the network by using its internal processing mechanisms instead of simply increasing its depth or embedding complex structures. We thus use the more reasonable deconvolution operation instead of the convolution operation for SISR to reconstruct more natural details, and introduce the Kullback–Leibler divergence into the final loss function to avoid the oversmoothness caused by the MSE loss.

Building a deep FDNN for SISR and using KL divergence loss during training represent a preliminary attempt to increase the performance of MFNNs based on their internal processing mechanism. We believe that the improvement and innovation of the

Table 4
Ablation study for a scale factor 3 on three datasets. *CiDj* denotes the mixed model contains *i* convolutional layers and *j* deconvolutional layers. FCNN denotes the model with fully convolutional layers.

Dataset		Set5	Set14	B100
Method	Criterion	×3	×3	×3
FCNN	PSNR	32.15	29.39	28.15
	SSIM	0.8999	0.8375	26.47
C8D2	PSNR	33.65	29.74	28.78
	SSIM	0.9214	0.8313	0.7972
C7D3	PSNR	33.65	29.76	28.78
	SSIM	0.9214	0.8316	0.7973
C6D4	PSNR	33.65	29.75	28.78
	SSIM	0.9214	0.8316	0.7973
C5D5	PSNR	33.66	29.74	28.78
	SSIM	0.9213	0.8314	0.7972
C4D6	PSNR	33.65	29.75	28.78
	SSIM	0.9214	0.8316	0.7972
C3D7	PSNR	33.67	29.77	28.79
	SSIM	0.9216	0.8318	0.7975
C2D8	PSNR	33.66	29.75	28.79
	SSIM	**0.9215**	0.8317	0.7975
FDNN(Ours)	PSNR	**33.68**	**29.80**	**28.80**
	SSIM	0.9213	**0.8320**	**0.7976**

algorithms applied in FDNN are more meaningful than the recombination of different structures for the development of MFNN applications.

Declaration of competing interest

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

Acknowledgments

This study was supported by the National Natural Science Foundation of Zhejiang Province, China under Grant LZ20F030001, and the National Natural Science Foundation of China under Grants 61672477 and 61876103.

References

Bevilacqua, M., Roumy, A., Guillemot, C., & Alberimorel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British machine vision conference* (pp. 1–10).

Bracewell, R. (2002). The fourier transform and its applications. *American Journal of Physics*, *34*.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., et al. (2014). Cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759.

Chollet, F., et al. (2015). *Keras*. GitHub, https://github.com/fchollet/keras.

Clevert, D., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In *International conference on learning representations*.

Dodgson, N. A. (1997). Quadratic interpolation for image resampling. *IEEE Transactions on Image Processing*, *6*(9), 1322–1326.

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307.

Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In *Proceedings of the European conference on computer vision* (pp. 391–407).

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). Visualizing higher-layer features of a deep network. *University of Montreal*, *1341*(3), 1.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

He, H., & Siu, W. C. (2011). Single image super-resolution using Gaussian process regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 449–456).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Jian, S., Xu, Z., & Shum, H. Y. (2008). Image super-resolution using gradient profile prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Kim, J., Lee, J. K., & Lee, K. M. (2016a). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1646–1654).

Kim, J., Lee, J. K., & Lee, K. M. (2016b). Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1637–1645).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations* (pp. 1–13).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the advances in neural information processing systems* (pp. 1097–1105).

Kui, J., Xiaogang, W., & Xiaoou, T. (2013). Image transformation based on learning dictionaries across image spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(2), 367–380.

Kwang In, K., & Younghee, K. (2010). Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(6), 1127.

Li, Y., Dong, W., Xie, X., Shi, G., Wu, J., & Li, X. (2018). Image super-resolution with parametric sparse model learning. *IEEE Transactions on Image Processing*, *27*(9), 4638–4650.

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. In *International conference on learning representations*.

Liu, H., Fu, Z., Han, J., Shao, L., Hou, S., & Chu, Y. (2019). Single image super-resolution using multi-scale deep encoder-decoder with phase congruency edge map guidance. *Information Sciences*, *473*, 44–58.

Mao, X., Shen, C., & Yang, Y. (2016). Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Proceedings of the advances in neural information processing systems* (pp. 2810–2818).

Martin, D. R., Fowlkes, C. C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE international conference on computer vision* (pp. 416–423).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the international conference on machine learning* (pp. 807–814).

Ni, K. S., & Nguyen, T. Q. (2007). Image superresolution using support vector regression. *IEEE Transactions on Image Processing*, *16*(6), 1596–1610.

Nickolls, J. R., Buck, I., Garland, M., & Skadron, K. (2008). Scalable parallel programming with CUDA. In *Proceedings of the international conference on computer graphics and interactive techniques* (pp. 40–53).

Parker, J. A., Kenyon, R. V., & Troxel, D. E. (1983). Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging*, *2*(1), 31–39.

Romano, Y., Protter, M., & Elad, M. (2014). Single image interpolation via adaptive nonlocal sparsity-based modeling. *IEEE Transactions on Image Processing*, *23*(7), 3085–3098.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). 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* (pp. 1874–1883).

Song, T., Chowdhury, S. R., Yang, F., & Dutta, J. (2020). PET image super-resolution using generative adversarial networks. *Neural Networks*, *125*, 83–91.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tai, Y., Liu, S., Brown, M. S., & Lin, S. (2010). Super resolution using edge prior and single image detail synthesis. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 2400–2407).

Timofte, R., De, V., & Gool, L. V. (2014). Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision* (pp. 1920–1927).

Timofte, R., Smet, V. D., & Gool, L. V. (2014). A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Proceedings of the Asian conference on computer vision* (pp. 111–126).

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

Wang, Z., Chen, J., & Hoi, S. C. (2020). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, X., & Huang, D. (2009). A novel density-based clustering framework by using level set method. *IEEE Transactions on Knowledge and Data Engineering*, *21*(11), 1515–1531.

Wang, L., Huang, Z., Gong, Y., & Pan, C. (2017). Ensemble based deep networks for image super-resolution. *Pattern Recognition*, *68*, 191–198.

Wang, X., Huang, D., & Xu, H. (2010). An efficient local Chan-Vese model for image segmentation. *Pattern Recognition*, *43*(3), 603–618.

Xu, L., Ren, J. S., Liu, C., & Jia, J. (2014). Deep convolutional neural network for image deconvolution. In *Proceedings of the advances in neural information processing systems* (pp. 1790–1798).

Yang, W., Feng, J., Yang, J., Zhao, F., Liu, J., Guo, Z., et al. (2017). Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, *26*(12), 5895–5907.

Yang, J., Wang, Z., Lin, Z., Cohen, S. D., & Huang, T. S. (2012). Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, *21*(8), 3467–3478.

Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, *19*(11), 2861–2873.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the European conference on computer vision* (pp. 818–833).

Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*.

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. In *Proceedings of the international conference on computer vision* (pp. 2018–2025).

Zeyde, R., Elad, M., & Protter, M. (2010). On single image scale-up using sparse-representations. In *Proceedings of the international conference on curves and surfaces* (pp. 711–730).

Zhang, Y., Fan, Q., Bao, F., Liu, Y., & Zhang, C. (2018). Single-image super-resolution based on rational fractal interpolation. *IEEE Transactions on Image Processing*, *27*, 3782–3797.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2020). Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, http://dx.doi.org/10.1109/TPAMI.2020.2968521.

Zhang, K., Wang, Z., Li, J., Gao, X., & Xiong, Z. (2019). Learning recurrent residual regressors for single image super-resolution. *Signal Processing*, *154*, 324–337.

Zhang, H., Yang, J., Zhang, Y., & Huang, T. S. (2010). Non-local kernel regression for image and video restoration. In *Proceedings of the European conference on computer vision* (pp. 566–579).

Zhao, Z., Glotin, H., Xie, Z., Gao, J., & Wu, X. (2012). Cooperative sparse representation in two opposite directions for semi-supervised image annotation. *IEEE Transactions on Image Processing*, *21*(9), 4218–4231.