applied optics

Inception learning super-resolution

Muhammad Haris,[1,*] M. Rahmat Widyanto,[2] and Hajime Nobuhara[3]

[1]Department of Advanced Science and Technology, Toyota Technological Institute, Nagoya, Japan
[2]Faculty of Computer Science, University of Indonesia, Depok, Indonesia
[3]Department of Intelligent Interaction Technologies, University of Tsukuba, Tsukuba, Japan
*Corresponding author: mharis@toyota-ti.ac.jp

Received 7 April 2017; revised 12 June 2017; accepted 15 June 2017; posted 29 June 2017 (Doc. ID 291961); published 21 July 2017

An efficient network for super-resolution, which we refer to as inception learning super-resolution (ILSR), is proposed. We adopt the inception module from GoogLeNet to exploit multiple features from low-resolution images, yet maintain fast training steps. The proposed ILSR network demonstrates low computation time and fast convergence during the training process. It is divided into three parts: feature extraction, mapping, and reconstruction. In feature extraction, we apply the inception module followed by dimensional reduction. Then, we map features using a simple convolutional layer. Finally, we reconstruct the high-resolution component using the inception module and a 1 × 1 convolutional layer. Experimental results demonstrate that the proposed network can construct sharp edges and clean textures, and reduce computation time by up to three orders of magnitude compared to state-of-the-art methods.

OCIS codes: (100.0100) Image processing; (110.0110) Imaging systems; (150.0150) Machine vision.

https://doi.org/10.1364/AO.56.006043

1. INTRODUCTION

Internet technologies have made various types of images easily available, and access to a wide range of image types has created opportunities for machine learning algorithms to learn image characteristics deeply. These opportunities have been exploited to develop super-resolution (SR) algorithms based on learning approaches [1–6]. The main goal of an SR algorithm is to recover high-frequency information from input low-resolution (LR) images to produce a high-resolution (HR) image.

Currently, learning methods are widely used to map LR to HR patches. SR using sparse representation is popular due to its ability to naturally encode the semantic information of images [1]. By collecting representative samples to create an over-complete dictionary, it is possible to discover the correct basis to correctly encode an input image. Yang *et al.* [5] and Zeyde *et al.* [7] focused on using a single pair of dictionaries; however, that approach can produce many redundancies, which may cause instability during the image reconstruction.

Recently, convolutional neural networks (CNNs) have been used to significantly improve the accuracy of image processing algorithms. Dong *et al.* [2] demonstrated that a CNN can map LR to HR patches called super-resolution convolutional neural networks (SRCNNs). That method employs very simple and lightweight CNNs with two hidden layers and a 3 × 3 filter. Kim *et al.* [8] introduced very deep network for super-resolution (VDSR), which is a very deep CNN with residual learning. VDSR gives accurate results but has critical issues in term of convergence speed. VDSR includes a 20-layer CNN with a 3 × 3 filter.

Fast super-resolution convolutional neural networks (FSRCNN) [9] has demonstrated superior performance compared to SRCNN. FSRCNN focused on improving the current SRCNN and proposed a faster and more accurate algorithm. FSRCNN redesigns a network using three main principles, i.e., deconvolution, dimension shrinking, and a small filter.

In this paper, we propose a computationally inexpensive convolutional network for SR images with fast convergence (Fig. 1). The proposed network is inspired by the inception module and residual learning. GoogLeNet [11] introduced inception, which uses multiple filter sizes in a single stream. This concept was proven in the imagenet large scale visual recognition challenge (ILSVRC) challenge. Residual learning introduced by He *et al.* [12] eases the training of networks and obtains better accuracy.

The reminder of this paper is organized as follows. Section 2 elaborates about the inception module and its characteristics. Section 3 explains the proposed inception learning super-resolution (ILSR) CNN and training strategy. Section 4 discusses the results of our experiments and analysis. Finally, in Section 5, we present our conclusions.

2. INCEPTION MODULE

The Inception module was proposed by GoogLeNet, the winning architecture on image classification competition ILSVRC2014. The inception module is based on a pattern recognition network that mimics the animal visual cortex by observing small details, middle-sized features, or almost whole images. Technically, the inception module consists of multiple convolution (1×1, 3×3, 5×5) and pooling layers. The module basically acts as multiple convolution filter inputs that are processed on the same input. It also does pooling at the same time. Then, all the results are concatenated. This allows the model to take advantage of multi-level feature extraction from each input. For example, it extracts 3×3 and 5×5 features at the same time.

The inception concept has been proven in the field of image classification. However, there is no clear evidence in the SR technique. Therefore, the ability to capture multiple features from input images is utilized by our proposed network. We investigate the trades from the use of multiple features in increasing quality of images and burdening the computational time. Originally, the inception module contains a pooling layer. However, we avoid using the pooling layer because it is used to reduce the dimension of the input image, which contradicts the goal of SR.

3. PROPOSED NETWORK

In this section, we describe in detail our proposed network, the training strategy, and how we achieve fast convergence in a computationally inexpensive network. In addition, we elaborate the use of the inception module for SR images. We then describe the initialization strategy and discuss tuning using an image dataset.

A. Proposed Network

The proposed network is illustrated in Fig. 1. The network is composed of only a convolutional layer, as we show in Code File 1, Ref. [13]. The proposed network can be divided into three parts, i.e., feature extraction, mapping, and reconstruction. Here, let conv(f_t, n_t) be the convolutional layer, where f_t is the filter size and n_t is the number of filters. In total, we use 10 convolutional layers.

The HR image is formulated with height M and width N, where the total number of pixels is $M \times N$. Let $X_H = \{x_m: m = 0, 1, 2, \ldots, M-1\}$ and $Y_H = \{y_n: n = 0, 1, 2, \ldots, N-1\}$ be finite sets that determine the number of pixels. The HR image is defined as the function $f_H: X_H \times Y_H \to I$ where $I = \{0, 1, 2, \ldots, 255\}$ is the value of each pixel. Then, down-sampled grid $(M' \times N')$ applies the following condition, $M' < M$ and $N' < N$. Let $X_L = \{x_m: m = 0, 1, 2, \ldots, M' - 1\}$ and $Y_L = \{y_n: n = 0, 1, 2, \ldots, N' - 1\}$ be finite sets that determine the number of pixels in the LR image. The LR image is defined as the function $f_L: X_L \times Y_L \to I$. The middle resolution (MR) image $I_m \in f_H$ is then obtained by up-sampling this LR image and has the same size as the HR images.

Theoretically, the SR algorithm has the ability to map the LR features to the appropriate HR feature spaces. The process starts by up-sampling an LR image into an MR image using a conventional interpolation, e.g., bicubic. In the feature extraction, we extract various features from MR space using our inception modules. Then, we map the MR features to the HR space using a 3×3 convolutional layer. Finally, we construct the high-frequency components by reconstructing the features obtained from the MR image.

Feature extraction and reconstruction exploit the benefits of the inception module. The inception module is used due to the greedy assumption. The inception model was implemented due to confusion resulting from using different filter sizes. Rather than selecting an optimal filter size, it uses multiple filter sizes and combines the results. This is very suitable for the SR image concept, where we can obtain various image features. Our inception module is illustrated in Fig. 2.

Another useful layer is the 1×1 convolutional layer, which reduces the dimensionality of the layers. We use a 1×1 convolutional layer to select the best features in feature extraction and reconstruction. In the feature extraction step, we extract 24 features from the inception module. Then, the 1×1 convolutional layer selects the best eight features for this step. Aligned with the feature extraction step, in the reconstruction step, the 1×1 convolutional layer is used to select the best candidate of HR features from the inception module.

Computationally inexpensive networks are required for real-time applications. The proposed network optimizes the number of filters. We analyze the optimum number of filters and

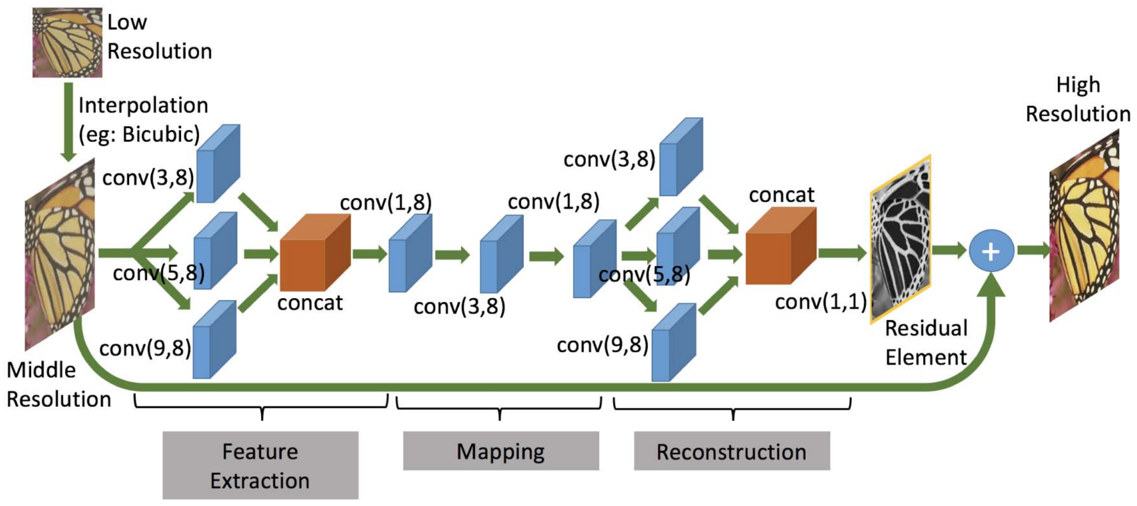

Fig. 1. Proposed network (butterfly image taken from Set-5 [10]).

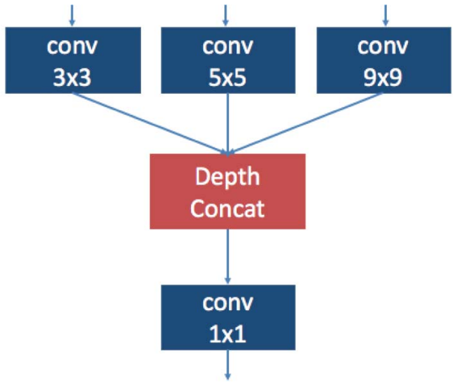

Fig. 2. Proposed inception module.

filter size based on an experiment. The use of the inception module can result in long computational time; however, we can achieve fast computation by reducing the number of filters.

We aim to achieve faster convergence during training compared to state-of-the-art methods. To achieve this, we construct the network using residual learning and gradient clipping. Residual learning [12] can ease the training of the networks. This assumption is motivated by the general SR problem where reconstructed HR images lose HR components during the enlargement process. The HR component can be substituted by a residual component from the proposed network.

In the proposed network, we use a high learning rate to achieve quick convergence. However, the high learning rate can affect infinity loss during the training process. Therefore, we use gradient clipping to avoid the infinity error. Gradient clipping is suitable for residual learning because it can limit the individual gradient to a predefined range. By using gradient clipping, we can avoid infinity error and ensure fast convergence.

B. Training Strategy

The learning step is very crucial in constructing optimal convolutional networks. Many studies have investigated this issue. During initialization, the std value for each layer should be determined. In the proposed network, we calculate the std value based on [14]. Here std is computed by $(\sqrt{2/n_l})$ where $n_l = k_l^2 d_l$, k_l is the filter size, and d_l is the number of filters. For example, with $k_l = 3$ and $d_l = 8$, the std is 0.111.

The use of various training images with clear edges and textures results in better performance for the proposed network. We adopt the training images used by Dong *et al.* [9]. First, we train the proposed network using 91 images. In the first training step, we use a 0.1 learning rate for each convolutional layer. Then, after the network is saturated, we use the combination of the 91 images and 100 general images for fine-tuning. In the tuning step, we increase the learning rate to 1 for each convolutional layer.

In the fine-tuning step, training images are generated using the image augmentation process proposed by Wang *et al.* [15]. We downscale the image into several scales and rotate each image in multiple degrees.

4. EXPERIMENTAL RESULTS

To confirm the efficiency of the proposed network, we conducted several quantitative and qualitative experiments. All experiments were conducted using Caffe and MATLAB R2015b under Windows 8 64-bit with an Intel Core i7 (3.2 GHz), 32 GB RAM, and an NVIDIA GTX 780 graphics processing unit (GPU). The images used in the experiment were taken from Yang *et al.* [5] and Dong *et al.* [9].

In the proposed network, we use two datasets for training (Fig. 3). We use a set of 41 × 41 pixel sub-images taken from the 91-image dataset (total size 267 MB) during the first training session. During this session, we use a 0.1 learning rate for each convolutional layer. Then, we tune the network using a set of 41 × 41 pixel sub-images taken from the 100-general-image dataset (total size 5.9 GB) with a learning rate equal to 1 for each convolutional layer.

Note that we enhance the brightness component (Y) while enhancing other components using bicubic interpolation because human vision is more sensitive to changes in brightness. Then, each resulting image channel is combined to produce a final color image. This procedure is very effective because it can speed up the computational process of the proposed network. Set-5 [10] is used for experimental testing. It consists of five images: baby (512 × 512), bird (288 × 288), butterfly (256 × 256), face (280 × 280), and woman (228 × 344). First, we downscale the original image to one-third of the original size to generate a test image. Then, the test image is used as input for various methods.

Figure 4 shows that the proposed network demonstrates rapid convergence. In the experiment, we train the network using 300,000 epochs. Note that training 100,000 epochs takes 2 h. For fine-tuning, we use another 300,000 epochs (each

Fig. 3. Samples from the 91-image dataset (top) and 100-general-images dataset (bottom) [9].

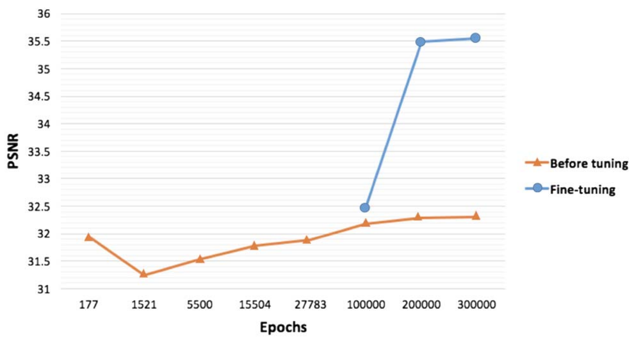

Fig. 4. PSNR value during first training and fine-tuning.

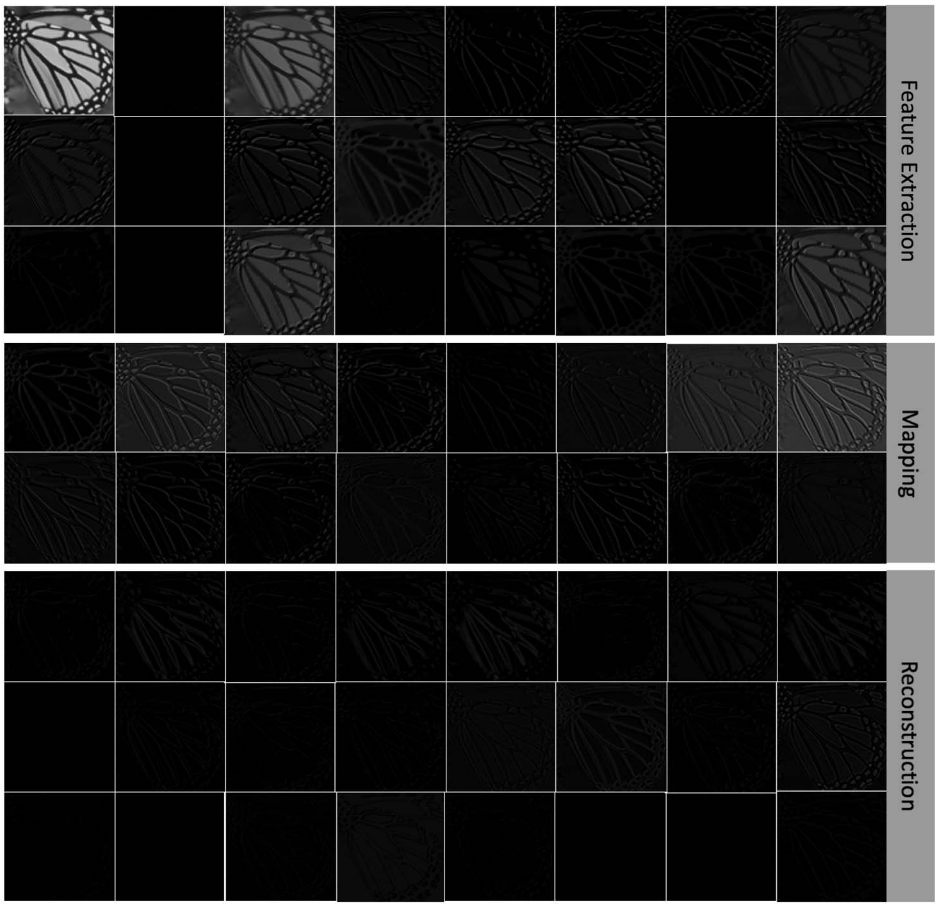

Fig. 5. Result of each step from the proposed network.

100,000 epochs takes 6 h). Thus, in total, we train the network using only 600,000 epochs in 24 h.

As can be seen in the Fig. 5, each layer from the proposed network produces specific features to obtain a residual image from the HR image. Feature extraction has the ability to obtain multi-level features from input LR. Then, the features are transformed in the mapping step. Finally, the features are altered to obtain an accurate residual image of the HR image. Here, we can conclude that the network is effective to produce unique features for each filter.

To evaluate the proposed network, we conducted experiments using magnified images (3 ×) to compare the proposed network to four other methods, i.e., bicubic, sparse-based [7], SRCNN-ex [2], and FSRCNN-s [9]. All parameters and training images were provided by the original authors using optimized setting.

A. Qualitative Analysis

Here, we present a qualitative evaluation of the results obtained by the proposed and other methods. The proposed method clearly produces sharper and smoother edges and can clearly construct the details of a scene compared to FSRCNN-s. FSRCNN-s and SRCNN-ex suffer from artifacts, particularly in edge areas, as shown in Fig. 6. Bicubic and sparse-based methods also produce blurring effects in the enlarged image. The best result was produced by SRCNN-ex, which yields

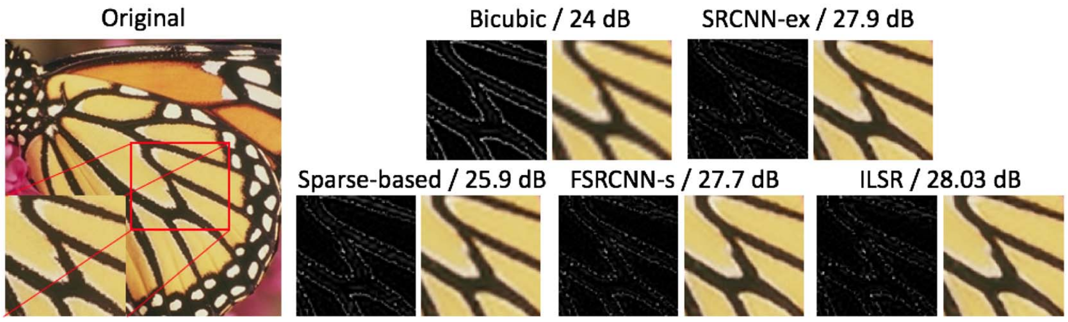

Fig. 6. 3 × magnification of a butterfly image from Set-5 images [10].

sharper edges and fewer artifacts. However, it is computationally expensive.

As can be seen in Fig. 6, the intensity of the differenced images tripled compared to the original values. We compare the proposed network to FSRCNN-s because both methods have similar computational costs. With the butterfly image (Fig. 6), the proposed network demonstrates a greater PSNR value than FSRCNN-s. As can be seen, the wing pattern produced by the proposed network has a clear shape and pattern. In contrast, FSRCNN-s produces halo artifacts.

B. Quantitative Analysis

Peak signal-to-noise ratio (PSNR) [16], structural similarity (SSIM) [17], feature similarity (FSIM) [18], and elapsed time were used to quantitatively evaluate the proposed method. PSNR (dB) represents the similarity between the original image and the upscaled image. SSIM measures the quality of images based on the structural content of the original and magnified images. FSIM is based on the fact that the human visual system processes an image primarily in terms of low-level features. Two features are considered in FSIM computation: the primary feature, i.e., phase congruency, which is a dimensionless measure of a local structure's significance, and the secondary feature, i.e., the image gradient magnitude. FSIM combines these features to characterize the local quality of an image. Note that higher PSNR, SSIM, and FSIM values indicate better quality. MATLAB functions (i.e., tic and toc) were used to measure the elapsed CPU time for a given process. To simplify and objectively calculate errors, all measurements used only the luminance channel (Y).

The results confirm that the proposed network is computationally less expensive than the other methods, except for the bicubic method. Moreover, the proposed method has greater PSNR, SSIM, and FSIM values compared to some methods. For example, compared to FSRCNN-s, which demonstrates comparable computational time, the proposed method demonstrates a greater PSNR value with the B100 dataset [19]. In addition, the proposed network shows greater SSIM and FSIM values for Set-5. Detailed comparisons are shown in Tables 1 and 2.

In Table 1, the proposed network can reduce around one-third of FSRCNN-s computational time and up to three orders of magnitude of SRCNN-ex computational time. In Table 2, the proposed network can reduce around one-fifth of FSRCNN-s computational time and around four orders of magnitude of SRCNN-ex computational time.

As can be seen, the proposed network has the lowest computational time, except when compared with the bicubic method. Table 3 shows the computation time with the Set-5 dataset. Note that all measurements used CPU-based MATLAB methods. Therefore, to achieve real-time application, the computation time can be reduced by implementing a general purpose GPU-based method.

In summary, the proposed method demonstrates competitive quality compared to current state-of-the-art methods, such as SRCNN-ex. It also offers the lowest computation time necessary for real-time application.

Table 1. PSNR, SSIM, FSIM, and Computational Time for 3× Magnification

Methods	PSNR	SSIM	FSIM	Time (s)
Bicubic	30.392	0.868	0.897	0.001
Sparse-based [7]	31.906	0.897	0.924	1.035
SRCNN-ex [2]	32.749	0.909	0.941	3.471
FSRCNN-s [9]	32.604	0.906	0.938	1.274
Proposed	32.565	0.908	0.939	0.917

Table 2. PSNR, SSIM, FSIM, and Computational Time for 3× Magnification

Methods	PSNR	SSIM	FSIM	Time (s)
Bicubic	27.207	0.738	0.827	0.001
Sparse-based [7]	27.875	0.773	0.856	1.565
SRCNN-ex [2]	28.412	0.786	0.876	5.321
FSRCNN-s [9]	28.284	0.783	0.873	1.690
Proposed	28.323	0.783	0.872	1.394

Table 3. Detailed Computational Time on 3× Magnification (in seconds)

Methods	Baby	Bird	Butterfly	Face	Woman
Bicubic	0.003	0.002	0.001	0.001	0.001
Sparse-based [7]	2.40	0.74	0.63	0.67	0.74
SRCNN-ex [2]	9.37	1.90	2.36	1.88	2.1
FSRCNN-s [9]	2.84	1.02	0.75	0.91	0.89
Proposed	2.62	0.48	0.53	0.48	0.51

5. CONCLUSIONS

We have proposed ILSR. The proposed network was inspired by GoogLeNet's inception module to produce multiple features during the feature extraction and reconstruction processes. The proposed strategies ensure that the network demonstrates fast convergence and low computation time. The proposed network was evaluated experimentally. The results show that the proposed network can reduce computation time by up to three orders of magnitude compared to state-of-the-art networks (such as SRCNN-ex). Furthermore, the proposed network successfully exploits the inception module and residual learning in the SR approach.

We aim to obtain a lightweight network while constructing clear and sharp HR images. Note that we did not consider or analyze the advantages of inception modules under various conditions. As a result, it is highly probable that the proposed network will become trapped in a local optimum solution. In the future, we will design a more efficient network to produce better quality images.

Acknowledgment. Muhammad Haris would like to thank the Indonesia Endowment Fund for Education (LPDP) Scholarships from the Ministry of Finance, The Republic of Indonesia, for a doctoral degree scholarship.

REFERENCES

1. M. Elad, "Sparse and redundant representation modeling–what next?" IEEE Signal Process. Lett. **19**, 922–928 (2012).

2. C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell. **38**, 295–307 (2016).
3. 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.
4. J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5197–5206.
5. J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” IEEE Trans. Image Process. **19**, 2861–2873 (2010).
6. M. Haris and H. Nobuhara, “Super-resolution based on edge-aware sparse representation via multiple dictionaries,” in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, Italy, 27–29 February 2016, Vol. **3**, pp. 42–49.
7. R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Curves and Surfaces* (Springer, 2012), pp. 711–730.
8. J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral)* (2016).
9. 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.
10. M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in *British Machine Vision Conference (BMVC)* (2012).
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9.
12. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” arXiv:1512.03385 (2015).
13. M. Haris, M. R. Widyanto, and H. Nobuhara, “Inception learning super-resolution,” Figshare, 2017, https://doi.org/10.6084/m9.figshare.5120053.
14. 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.
15. Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deeply improved sparse coding for image super-resolution,” arXiv preprint arXiv:1507.08905 (2015).
16. M. Irani and S. Peleg, “Motion analysis for image enhancement: resolution, occlusion, and transparency,” J. Visual Commun. Image Represent. **4**, 324–335 (1993).
17. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Trans. Image Process. **13**, 600–612 (2004).
18. L. Zhang, L. Zhang, X. Mou, and D. Zhang, “Fsim: a feature similarity index for image quality assessment,” IEEE Trans. Image Process. **20**, 2378–2386 (2011).
19. 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 *Proceedings. Eighth IEEE International Conference on Computer Vision (ICCV)* (2001), Vol. **2**, pp. 416–423.