Image Super-Resolution via Generative Adversarial Network Using an Orthogonal Projection

Hiroya Yamamoto, Daichi Kitahara, and Akira Hirabayashi
Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan

***Abstract*—In this paper, we propose a simple but powerful idea to improve super-resolution (SR) methods based on convolutional neural networks (CNNs). We consider a linear manifold, which is the set of all SR images whose downsampling results are the same as the input image, and apply the orthogonal projection onto this linear manifold in the output layers of the CNNs. The proposed method can guarantee the consistency between the SR image and the input image and reduce the mean square error. The proposed method is especially effective for SR methods based on generative adversarial networks (GANs), composed of one *generator* and one *discriminator*, since the generator can learn high-frequency components while maintaining low-frequency ones. Experiments show the effectiveness of the proposed technique for a GAN-based SR method. Finally we introduce an idea of extension to noisy images.**

***Index Terms*—Single image super-resolution, generative adversarial network, orthogonal projection, constrained learning.**

I. Introduction

Super-resolution (SR) is a reconstruction problem of high-resolution (HR) images including various high-frequency components from low-resolution (LR) images including only low-frequency components [1]–[13]. In SR, it is important not only to increase the number of pixels but also to recover the original high-frequency components. In this paper, we focus on single image SR, which is an under-determined inverse problem since we have to recover an HR image from a single LR image having a smaller number of pixels. The simplest ways to increase the number of pixels are *algebraic interpolations*, e.g., the nearest-neighbor, bilinear, and bicubic interpolations. Although these algebraic methods are very fast, they cannot recover the high-frequency components at all. Therefore, SR results, called SR images in this paper, of the algebraic methods are very blurred.

To accurately recover the high-frequency components, most SR methods learn the transform from LR images to HR images by using training data. SR methods based on dictionary learning [2], [3] were studied before, but recently SR methods based on convolutional neural networks (CNNs) [4]–[11] are mainly used in terms of both reconstruction accuracy and processing time. Dong *et al.* proposed the first end-to-end CNN for SR, named SRCNN [4]. SRCNN transforms *interpolated LR images*, which are enlarged to the HR image size by the bicubic interpolation, into SR images through three convolution layers. There exist many improved versions of SRCNN [5]–[11]. For example, VDSR [6] increased the number of the convolution layers by introducing the *residual learning* to avoid the gradient vanishing. ESPCN [7] proposed the sub-pixel convolution layer called *pixel shuffler*, which enlarges LR images at various magnification ratios and removes the bicubic interpolation utilized in the input layer of SRCNN. In each SR method [5]–[11], a single CNN, *generator*, is trained by minimizing mainly the mean square error (MSE) or the mean absolute error (MAE) between the true HR images and the generated SR ones.

It is well-known that SR images generated from the CNNs based on MSE or MAE are over-smoothed yet, i.e., recovery of high-frequency components is still insufficient, since the high-frequency components hardly contribute to the values of MSE and MAE. Therefore, Ledig *et al.* proposed SRGAN [12] that uses a generative adversarial network (GAN) [14]. SRGAN is composed of two CNNs, i.e., a *generator* and a *discriminator*. The discriminator is trained to judge whether the input image is a true HR image or a generated SR one. Since the generator tries to induce the discriminator's misjudgment, SR images including realistic high-frequency components can be generated. However, the generated SR images have some artifacts which do not exist in true HR images, and values of PSNR decrease.

In this paper, we consider the set of all SR images whose downsampling results correspond to the input LR image. This set becomes a linear manifold, and the orthogonal projection onto this set can be easily computed as in [15] under a simple downsampling model in (1) (see Sect. II.A). We propose to use the orthogonal projection in the output layer of the generator. The proposed method can be applied to any generator and SR images having the perfect consistency with the input images are always generated. Therefore, the generator can learn high-frequency components while keeping true low-frequency ones. Numerical experiments show that the proposed method significantly reduces the artifacts of SR images generated by a GAN-based method while reconstructing high-frequency components at high accuracy. Finally, we conclude this paper and introduce an extension of the proposed method to noisy LR images.

II. Image Super-Resolution by Neural Networks

A. Formulation of Downsampling

Let $Y := (Y_{i,j,c}) \in \mathbb{R}^{I \times J \times 3}$ be a low-resolution (LR) RGB color image to be enlarged, and $Y_{i,j,c} \in \mathbb{R}$ be the (i,j)th pixel value ($i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$) of color channel c ($c = \mathrm{R}, \mathrm{G}, \mathrm{B}$). Suppose that Y is the downsampling result of a high-resolution (HR) color image $X := (X_{i,j,c}) \in \mathbb{R}^{IK \times JL \times 3}$ with slight anti-aliasing (i.e., non-overlapped slight blurring) as

$$Y_{i,j,c} = \sum_{k=1}^{K} \sum_{l=1}^{L} w_{k,l} X_{(i-1)K+k,(j-1)L+l,c}, \qquad (1)$$

This work was supported in part by JSPS Grants-in-Aid Grant Number for Early-Career Scientists JP19K20361 (e-mail: d-kita@fc.ritsumei.ac.jp).

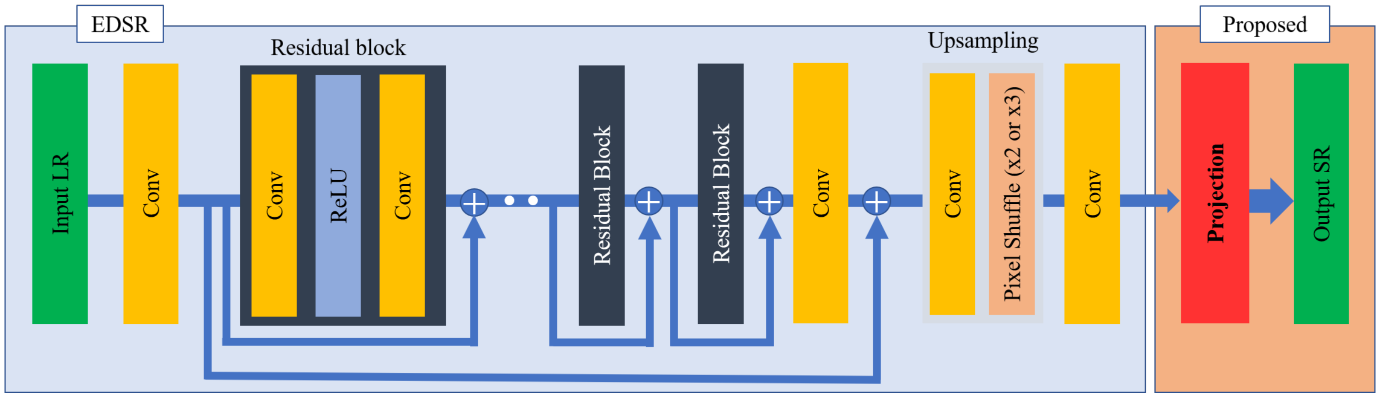


Fig. 1. The proposed generator model based on EDSR. The inputs are LR images and the outputs are SR images projected onto the linear manifold $\mathcal{A}$ in (8).

where K and L are supposed to be integers lager than 1, and $w_{k,l} \in \mathbb{R}$ are downsampling weights[1] s.t. $\sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l} = 1$. Define $\boldsymbol{y} \in \mathbb{R}^{3IJ}$ and $\boldsymbol{x} \in \mathbb{R}^{3IJKL}$ as the vectorized versions of the LR image Y and the HR image X, respectively. Then by using a *block-diagonal-like* matrix $A \in \mathbb{R}^{3IJ\times 3IJKL}$, the downsampling model in (1) is expressed as a matrix form[2] by

$$\boldsymbol{y} = A\boldsymbol{x}. \tag{2}$$

B. Super-Resolution via Single Convolutional Neural Network

Image super-resolution (SR) based on a single convolutional neural network (CNN) has been studied by many researchers [4]–[11], after the work of SRCNN [4]. Especially, SRResNet proposed by Ledig *et al.* as the *generator* of SRGAN [12] (see also Sect. II.C below) is often adopted as a baseline. SRResNet utilizes the architecture of ResNet [17], that is developed originally for the image recognition, and has many *residual blocks* composed of two convolution layers, two batch normalization layers, and one rectified linear unit (ReLU). At the end of each residual block, the input of the residual block is added for the *residual learning* which enables CNNs to avoid the gradient vanishing. Lim *et al.* proposed EDSR [8] as an improved version of SRResNet. In EDSR, the batch normalization layers are removed because they decrease the flexibility of CNNs and use a lot of memory. In the blue box of Fig. 1, the architecture of EDSR is shown. In experiments of Sect. IV, we adopt EDSR, instead of SRResNet, as a baseline and a *generator* of a GAN. Besides the above methods, EUSR [9], RCAN [10], and SAN [11] are proposed as single-CNN-based SR methods.

Let $\{(\boldsymbol{y}_n, \boldsymbol{x}_n)\}_{n=1}^{N}$ be training data composed of N pairs of LR and HR color images. Let $\hat{\boldsymbol{x}}_n \in \mathbb{R}^{3IJKL}$ be the outputs, called SR images, of a certain SR network for the inputs $\boldsymbol{y}_n \in \mathbb{R}^{3IJ}$. As a loss function to be minimized for training of the SR network, the mean square error (MSE)

$$l_{\mathrm{MSE}} = \frac{1}{3IJKLN}\sum_{n=1}^{N}\|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_2^2 \tag{3}$$

is usually adopted, where $\|\cdot\|_2$ denotes the ℓ_2 norm of a vector. Some papers claim that the mean absolute error (MAE)

$$l_{\mathrm{MAE}} = \frac{1}{3IJKLN}\sum_{n=1}^{N}\|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_1 \tag{4}$$

leads to slightly better results, where $\|\cdot\|_1$ is the ℓ_1 norm. The difference between (3) and (4) has little effect on human eyes.

C. Super-Resolution via Generative Adversarial Network

Ledig *et al.* proposed an image SR method using a generative adversarial network (GAN) [14] for the first time, and it was named SRGAN [12]. Conventional methods [4]–[11] train a single CNN by minimizing (3) or (4), but they cannot sufficiently recover high-frequency components of the HR images. On the other hand, SRGAN is composed of two CNNs, i.e., a *generator* and a *discriminator*, and can create high-frequency components *although they might differ from the original ones*.

In SRGAN, the generator (SRResNet) $G : \mathbb{R}^{3IJ} \to \mathbb{R}^{3IJKL}$ and the discriminator $D : \mathbb{R}^{3IJKL} \to (0, 1)$ are alternately updated. The discriminator D judges whether the input image is a true HR image or a generated SR one by maximizing

$$l_{\mathrm{D}} = \frac{1}{N}\sum_{n=1}^{N}\log(D(\boldsymbol{x}_n)) + \frac{1}{N}\sum_{n=1}^{N}\log(1 - D(\hat{\boldsymbol{x}}_n)), \tag{5}$$

where $\hat{\boldsymbol{x}}_n = G(\boldsymbol{y}_n)$ is the SR image generated from the input LR image $\boldsymbol{y}_n$ by the current generator G. In GAN-based techniques, the second term $\frac{1}{N}\sum_{n=1}^{N}\log(1-D(\hat{\boldsymbol{x}}_n))$ in (5) is usually used as a loss function for G, but in the context of SR,

$$l_{\mathrm{A}} = -\frac{1}{N}\sum_{n=1}^{N}\log(D(\hat{\boldsymbol{x}}_n)) \tag{6}$$

is often used as the *adversarial loss* due to its better gradient behavior [12]. As a result, the total loss function for G can be expressed as

$$l = l_{\mathrm{C}} + \kappa l_{\mathrm{A}}, \tag{7}$$

[1] If downsampling is the nearest-neighbor interpolation or the bilinear interpolation, then (1) is always satisfied. If downsampling is the bicubic *polynomial* interpolation and both K and L are equal to or larger than 4, then (1) is also satisfied, but some weights are negative. However, if downsampling is the bicubic *polynomial* interpolation and K or L is equal to or smaller than 3, then (1) is not satisfied. If downsampling is the bicubic *spline* interpolation, then (1) is never satisfied. If standard anti-aliasing, i.e., overlapped blurring as in [16], is done right before interpolation, then (1) becomes an approximation model.

[2] Although most downsampling methods, including the bicubic *polynomial* and *spline* interpolations, are expressed as linear operators A in (2), the condition in (1) is important to easily compute the orthogonal projection as in (10).

where l_{C} is the *content loss* evaluating the consistency between $\hat{\boldsymbol{x}}_n$ and $\boldsymbol{x}_n$, and $\kappa > 0$ is a weight for the adversarial loss l_{A} in (6). In the simplest cases, the content loss l_{C} is defined as l_{MSE} in (3) or l_{MAE} in (4). In more complicated cases [12], [13], the content loss l_{C} is defined, for example, as MSE of the VGG feature maps [18] or MAE of the differential images.

III. Orthogonal Projection in the Output Layer

In the conventional loss function in (7), the content loss l_{C} mainly evaluates the differences between SR images $\hat{\boldsymbol{x}}_n$ and true HR images $\boldsymbol{x}_n$. However, it is not considered whether re-downsampling results $A\hat{\boldsymbol{x}}_n$ are close to given LR images $\boldsymbol{y}_n$ or not. On the other hand, the true HR images $\boldsymbol{x}_n$ always satisfy $A\boldsymbol{x}_n = \boldsymbol{y}_n$. To make the most of training data, we should also consider the re-downsampling results $A\hat{\boldsymbol{x}}_n$. In this paper, we propose a modification technique which enables *any generator* to generate SR images $\check{\boldsymbol{x}}_n$ satisfying $A\check{\boldsymbol{x}}_n = \boldsymbol{y}_n$. The proposed technique is expected to be effective especially for GAN-based SR methods because the generator learns high-frequency components while maintaining the original low-frequency components, i.e., information on the input LR images.

First, we define the set of all SR images matching the input LR image by

$$\begin{aligned}\mathcal{A}_n &:= \{\boldsymbol{x} \in \mathbb{R}^{3IJKL} \mid A\boldsymbol{x} = \boldsymbol{y}_n\} \\ &= \{\boldsymbol{x}_n + \boldsymbol{z} \in \mathbb{R}^{3IJKL} \mid A\boldsymbol{z} = \boldsymbol{0}\} = \boldsymbol{x}_n + \mathcal{N}(A), \quad (8)\end{aligned}$$

where $\mathcal{N}(A)$ denotes the null space of A. From (8), it is found that the set $\mathcal{A}_n$ is a *linear manifold* and the true HR image $\boldsymbol{x}_n$ always belongs to $\mathcal{A}_n$. By applying the orthogonal projection $P_{\mathcal{A}_n}$ onto $\mathcal{A}_n$, in the output layer (see the red box of Fig. 1), to the conventional SR image $\hat{\boldsymbol{x}}_n$, we obtain the proposed SR image $\check{\boldsymbol{x}}_n := P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)$ which matches the input LR image $\boldsymbol{y}_n$. The proposed SR image $\check{\boldsymbol{x}}_n$ is concretely expressed as

$$\begin{aligned}\check{\boldsymbol{x}}_n = P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) &= \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{A}_n} \|\hat{\boldsymbol{x}}_n - \boldsymbol{x}\|_2 \\ &= \hat{\boldsymbol{x}}_n - A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}(A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n). \quad (9)\end{aligned}$$

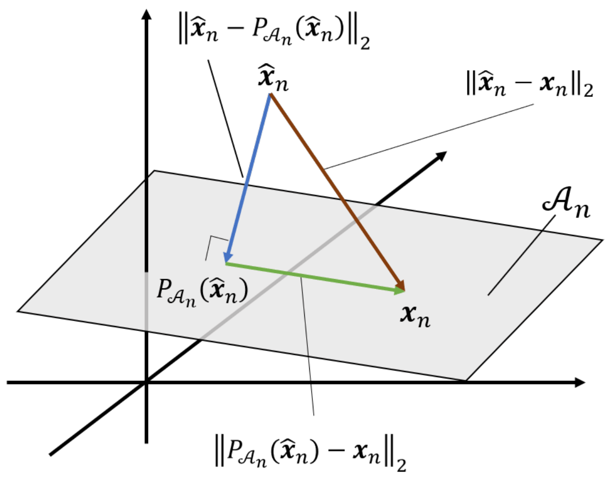
In general, the exact computation of $(AA^{\mathrm{T}})^{-1}$ is difficult when the image size becomes huge. In this paper, since we assumed the blockwise downsampling model as in (1), AA^{T} is always a *diagonal matrix* and hence (9) is easily computed by

$$\check{\boldsymbol{x}}_n = P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) = \hat{\boldsymbol{x}}_n - \frac{1}{\sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l}^2} A^{\mathrm{T}}(A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n). \quad (10)$$

As shown in Fig. 2, MSE between $\hat{\boldsymbol{x}}_n$ and $\boldsymbol{x}_n$ in (3) (brown line) can be divided into vertical components to $\mathcal{A}_n$ (blue line) and horizontal components to $\mathcal{A}_n$ (green line), and we have

$$\begin{aligned}\|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_2^2 &= \|\hat{\boldsymbol{x}}_n - P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)\|_2^2 + \|P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) - \boldsymbol{x}_n\|_2^2 \\ &\geq \|P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) - \boldsymbol{x}_n\|_2^2 = \|\check{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_2^2. \quad (11)\end{aligned}$$

From (11), if $A\hat{\boldsymbol{x}}_n \neq \boldsymbol{y}_n$, then MSE always becomes smaller, i.e., PSNR always improves, by the orthogonal projection $P_{\mathcal{A}_n}$. Since the output image is changed from $\hat{\boldsymbol{x}}_n$ to $P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)$, the



$\|\hat{\boldsymbol{x}}_n - P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)\|_2$
$\hat{\boldsymbol{x}}_n$
$\|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_2$
$\mathcal{A}_n$
$P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)$
$\boldsymbol{x}_n$
$\|P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) - \boldsymbol{x}_n\|_2$

Fig. 2. Linear manifold $\mathcal{A}_n$ in (8) and errors of horizontal/vertical directions.

MSE content loss is changed to

$$l_{\mathrm{C}'} = \frac{1}{3IJKLN}\sum_{n=1}^{N}\|P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n) - \boldsymbol{x}_n\|_2^2, \quad (12)$$

where we use not MAE but MSE as the content loss for stable training. When we only consider a weighted sum of $l_{\mathrm{C}'}$ in (12) and l_{A} in (6) as the total loss function l for the generator G, actually SR results are *not good* because error information of the vertical components to $\mathcal{A}_n$ is lost. Hence, to generate SR images as close as possible to the linear manifold $\mathcal{A}_n$ before the orthogonal projection, we further propose to consider MSE of the vertical components to $\mathcal{A}_n$ as the *projection loss*

$$l_{\mathrm{P}} = \frac{1}{3IJKLN}\sum_{n=1}^{N}\|\hat{\boldsymbol{x}}_n - P_{\mathcal{A}_n}(\hat{\boldsymbol{x}}_n)\|_2^2. \quad (13)$$

Note that, under the downsampling model in (1), the projection loss l_{P} is also expressed as

$$\begin{aligned}l_{\mathrm{P}} &= \frac{1}{3IJKLN(\sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l}^2)^2}\sum_{n=1}^{N}\|A^{\mathrm{T}}(A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n)\|_2^2 \\ &= \frac{1}{3IJKLN\sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l}^2}\sum_{n=1}^{N}\|A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n\|_2^2. \quad (14)\end{aligned}$$

From (14), the projection loss is essentially equivalent to MSE between the re-downsampling results $A\hat{\boldsymbol{x}}_n$ and the LR images $\boldsymbol{y}_n$.[3] Finally the proposed total loss function for G is defined as

$$l = l_{\mathrm{C}'} + \lambda l_{\mathrm{P}} + \kappa l_{\mathrm{A}}, \quad (15)$$

where $\lambda > 0$ and $\kappa > 0$. Since the content loss $l_{\mathrm{C}'}$ in (12) is more important than the projection loss l_{P} in (13), we recommend to use λ which is smaller than 1. In (15), we evaluate the horizontal and vertical MSEs at a ratio of $1 : \lambda$ while in (7) the conventional methods evaluated them at a ratio of $1 : 1$.

[3] Some downsampling losses similar to (14) have been proposed in [19]–[21]. Differently from [19]–[21], we applied the orthogonal projection in the output layer and clarified the equivalence of the projection and downsampling losses.

IV. Numerical Experiments

We compare SR results of EDSR, EDSR-Projection, EDSR-GAN, and EDSR-GAN-Projection, where 'Projection' means that the orthogonal projection is added in the output layer of the generator and the content loss $l_{\mathrm{C}} = l_{\mathrm{MSE}}$ is replaced with $l_{\mathrm{C}'} + \lambda l_{\mathrm{P}}$, and 'GAN' means that the discriminator of SRGAN [12] is trained and used in the adversarial loss l_{A}. We set the weights $\lambda = 10^{-3}$ and $\kappa = 10^{-3}$. DIV2K dataset is adopted as training data. We set the size of training HR patches as 96×96 for $\times 2$ scale (i.e., $K = L = 2$) and 144×144 for $\times 3$ scale. For each scale, all LR patches of size 48×48 are given by downsampling of the HR patches with the *arithmetic mean matrix* A. We evaluate the SR performance of each method for four test datasets, Set5, Set14, BSD100 and Urban100. We prepare two kinds of test LR images. One is downsampled with the matrix A, and the other is given with the bicubic spline interpolation.

We utilize Adam [22] as the optimizer, where we set $\alpha = 10^{-4}$ in the training of the CNNs, $\alpha = 10^{-5}$ in the training of the GANs, and $\beta = 0.9$. For stable GAN training, a pre-trained EDSR is used as the initial value of the generator. We set the minibatch size as 96 to speed up training, and use PSNR and FSIM$_C$ [23] as image quality indices for comparison. It takes about 2 days to train the networks with GeForce GTX 1080ti.

Tables I and II summarize the SR results for the two kinds of the LR images. Among the single-CNN-based methods, there are no significant differences, which means that a MSE-based CNN almost satisfies $A\hat{\boldsymbol{x}}_n = \boldsymbol{y}_n$ without the orthogonal projection. On the other hand, among the GAN-based methods, the proposed method improves PSNR by the orthogonal projection in Tables I and II. Moreover, the proposed method improves FSIM$_C$ mainly for $\times 3$ scale in Table I, and for the both scales in Table II (though the downsampling is different from training data), which means that the proposed method has a certain robustness against the different downsampling. Figures 3 and 4 show two SR results. The SR images of the MSE-based CNNs are over-smoothed, and those of EDSR-GAN seem clearer but include some artifacts. On the other hand, the proposed GAN-based SR (EDSR-GAN-Projection) generates high-frequency components accurately while greatly reducing the artifacts.

V. Conclusion and Future Work

In this paper, for image SR using CNNs, we proposed to use an orthogonal projection in the output layer for generating SR images having the perfect consistency with the input LR images. In particular, the proposed method is effective for GAN-based SR methods since generators learn high-frequency components while keeping the original low-frequency ones. Note that the proposed method can be applied to another constrained learning if the projection onto the constraint is available.

As future work, by considering noise and model error of A, we plan to use the set of all images of noise level under $\epsilon > 0$:

$$\mathcal{A}_n^{\epsilon} := \{\boldsymbol{x} \in \mathbb{R}^{3IJKL} \mid \|A\boldsymbol{x} - \boldsymbol{y}_n\|_2 \leq \epsilon\}.$$

Under the condition in (1), the projection onto $\mathcal{A}_n^{\epsilon}$ is given by

$$P_{\mathcal{A}_n^{\epsilon}}(\hat{\boldsymbol{x}}_n) = \hat{\boldsymbol{x}}_n - \frac{\delta_n^{\epsilon}(\hat{\boldsymbol{x}}_n)}{\sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l}^2} A^{\mathrm{T}}(A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n),$$

where $\delta_n^{\epsilon} : \mathbb{R}^{3IJKL} \to \mathbb{R}$ is a *differentiable* function defined as

$$\delta_n^{\epsilon}(\hat{\boldsymbol{x}}_n) := \begin{cases} 0 & \text{if } \|A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n\|_2 \leq \epsilon, \\ \dfrac{\|A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n\|_2 - \epsilon}{\|A\hat{\boldsymbol{x}}_n - \boldsymbol{y}_n\|_2} & \text{otherwise.} \end{cases}$$

References

[1] B. Hauang, W. Wang, M. Bates, and X. Zhuang, "Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy," *Science*, vol. 319, no. 5864, pp. 810–813, 2008.

[2] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[3] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY: Springer, 2010.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016.

[5] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. ECCV*, Amsterdam, the Netherlands, 2016, pp. 391–407.

[6] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE CVPR*, Las Vegas, NV, 2016, pp. 1646–1654.

[7] 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 *Proc. IEEE CVPR*, Las Vegas, NV, 2016, pp. 1874–1883.

[8] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. "Enhanceddeep residual networks for single image super-resolution," in *Proc. IEEE CVPRW*, Honolulu, HI, 2017, pp. 1132–1140.

[9] J. H. Kim and J. S. Lee, "Deep residual network with enhanced upscaling module for super-resolution," in *Proc. IEEE/CVF CVPRW*, Salt Lake City, UT, 2018, pp. 800–808.

[10] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, Munich, Germany, 2018, pp. 294–310.

[11] T. Dai, J. Cai, Y. Zhang, S. T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF CVPR*, Long Beach, CA, 2019, pp. 11057–11066.

[12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE CVPR*, Honolulu, HI, 2017, pp. 105–114.

[13] M. Cheon, J. H. Kim, J. H. Choi, and J. S. Lee, "Generative adversarial network-based image super-resolution using perceptual content losses," in *Proc. ECCV*, Munich, Germany, 2018, pp. 51–62.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, Montreal, Canada, 2014, pp. 2672–2680.

[15] L. Condat and A. Montanvert, "A framework for image magnification: Induction revisited," in *Proc. IEEE ICASSP*, Philadelphia, PA, 2005, vol. 2, pp. 845–848.

[16] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Las Vegas, NV, 2016, pp. 770–778.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, 2015, 14 pages.

[19] R. Chen, Y. Qu, K. Zeng, J. Guo, C. Li, and Y. Xie, "Persistent memory residual network for single image super resolution," in *Proc. IEEE/CVF CVPRW*, Salt Lake City, UT, 2018, pp. 922–929.

[20] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE/CVF CVPR*, Salt Lake City, UT, 2018, pp. 9446–9454.

[21] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-supervised photo upsampling via latent space exploration of generative models," in *Proc. IEEE/CVF CVPR*, Seattle, WA, 2020, pp. 2437–2445.

[22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, 2015, 15 pages.

[23] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.

TABLE I

SR RESULTS (PSNR / FSIM$_C$) FOR SET5, SET14, BSD100 AND URBAN100. TEST IMAGES ARE DOWNSAMPLED BY THE MATRIX A USED IN TRAINING.

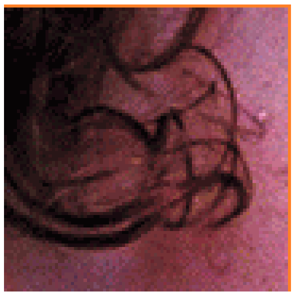
Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	36.31 / 0.9728	36.31 / 0.9726	34.02 / 0.9664	35.01 / 0.9625
	×3	34.27 / 0.9416	34.26 / 0.9376	33.02 / 0.9251	33.33 / 0.9300
Set14	×2	33.91 / 0.9538	33.93 / 0.9540	32.70 / 0.9436	32.98 / 0.9418
	×3	32.59 / 0.9065	32.58 / 0.9059	31.59 / 0.8890	31.88 / 0.8969
BSD100	×2	33.76 / 0.9396	33.76 / 0.9399	32.38 / 0.9275	32.84 / 0.9285
	×3	32.47 / 0.8807	32.47 / 0.8810	31.50 / 0.8685	31.71 / 0.8743
Urban100	×2	33.71 / 0.9468	33.73 / 0.9464	32.50 / 0.9332	32.91 / 0.9325
	×3	32.29 / 0.8854	32.29 / 0.8856	31.30 / 0.8657	31.55 / 0.8729

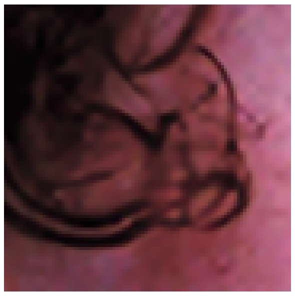
TABLE II

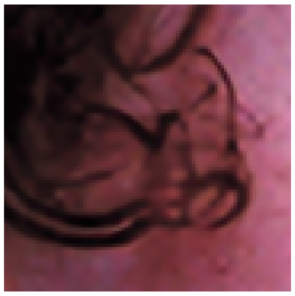
SR RESULTS (PSNR / FSIM$_C$) FOR SET5, SET14, BSD100 AND URBAN100. TEST IMAGES ARE DOWNSAMPLED BY THE BICUBIC SPLINE INTERPOLATION.

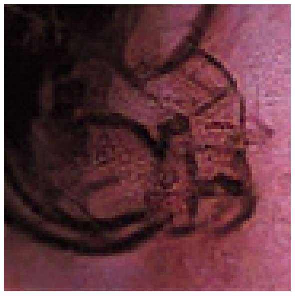
Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	35.97 / 0.9697	35.98 / 0.9697	33.69 / 0.9637	35.15 / 0.9644
	×3	33.88 / 0.9340	33.87 / 0.9339	33.21 / 0.9250	33.36 / 0.9285
Set14	×2	33.45 / 0.9370	33.48 / 0.9378	32.49 / 0.9302	32.98 / 0.9314
	×3	32.40 / 0.8961	32.39 / 0.8957	31.83 / 0.8885	32.02 / 0.8939
BSD100	×2	33.60 / 0.9360	33.59 / 0.9361	32.41 / 0.9238	33.08 / 0.9306
	×3	32.33 / 0.8704	32.33 / 0.8706	31.79 / 0.8671	31.92 / 0.8708
Urban100	×2	33.10 / 0.9336	33.11 / 0.9337	32.25 / 0.9241	32.70 / 0.9272
	×3	31.94 / 0.8636	31.94 / 0.8636	31.48 / 0.8576	31.63 / 0.8624

(a) HR image


(a') HR image


(b) EDSR
(32.38 / 0.9091)


(c) EDSR-Projection
(32.37 / 0.9092)


(d) EDSR-GAN
(31.64 / 0.8969)

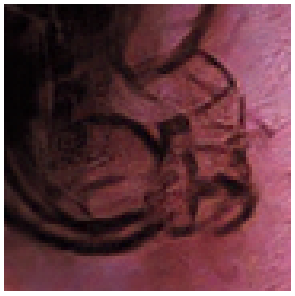
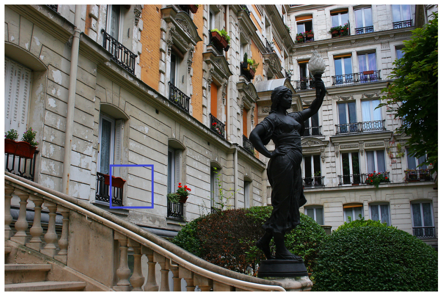
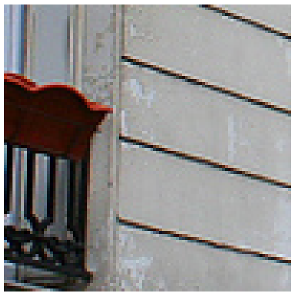
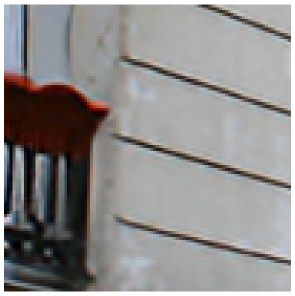
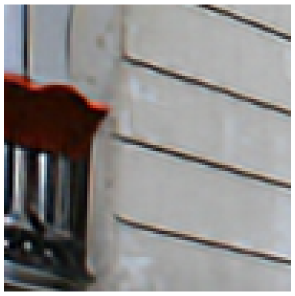

(e) EDSR-GAN-Projection
(31.78 / 0.9017)

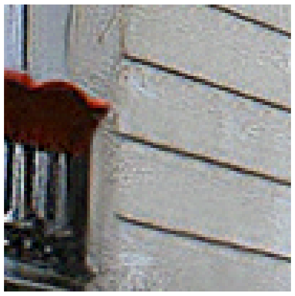
Fig. 3. SR results of 'head' in Set5 for ×3 scale (PSNR / FSIM$_C$).


(a) HR image


(a') HR image


(b) EDSR
(30.55 / 0.8582)


(c) EDSR-Projection
(30.55 / 0.8568)


(d) EDSR-GAN
(29.74 / 0.8349)

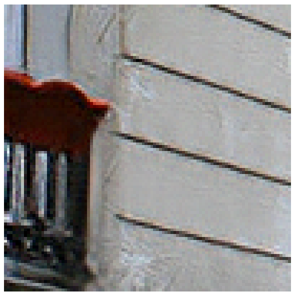

(e) EDSR-GAN-Projection
(29.96 / 0.8452)

Fig. 4. SR results of 'img003' in Urban100 for ×3 scale (PSNR / FSIM$_C$).