

PAPER

Image Super-Resolution via Generative Adversarial Networks Using Metric Projections onto Consistent Sets for Low-Resolution Inputs^{*}

Hiroya YAMAMOTO[†], *Nonmember*, Daichi KITAHARA^{†a)}, Hiroki KURODA^{†b)},
and Akira HIRABAYASHI^{†c)}, *Members*

SUMMARY This paper addresses single image super-resolution (SR) based on convolutional neural networks (CNNs). It is known that recovery of high-frequency components in output SR images of CNNs trained by the least square errors or least absolute errors is insufficient. To generate realistic high-frequency components, SR methods using generative adversarial networks (GANs), composed of one *generator* and one *discriminator*, are developed. However, when the generator tries to induce the discriminator's misjudgment, not only realistic high-frequency components but also some *artifacts* are generated, and objective indices such as PSNR decrease. To reduce the artifacts in the GAN-based SR methods, we consider the set of all SR images whose square errors between downscaling results and the input image are within a certain range, and propose to apply the *metric projection* onto this *consistent set* in the output layers of the generators. The proposed technique guarantees the consistency between output SR images and input images, and the generators with the proposed projection can generate high-frequency components with few artifacts while keeping low-frequency ones as appropriate for the known noise level. Numerical experiments show that the proposed technique reduces artifacts included in the original SR images of a GAN-based SR method while generating realistic high-frequency components with better PSNR values in both *noise-free* and *noisy* situations. Since the proposed technique can be integrated into various generators if the downscaling process is known, we can give the consistency to existing methods with the input images without degrading other SR performance.

key words: *single image super-resolution, convolutional neural network, generative adversarial network, metric projection, consistent set*

1. Introduction

Super-resolution (SR) is a reconstruction problem of high-resolution (HR) images containing high-frequency components from given low-resolution (LR) images having only low-frequency components [2]–[21]. 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 address single image SR that 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 interpo-*

lations, e.g., the nearest-neighbor, bilinear, bicubic, and natural bicubic spline interpolations. Although these algebraic methods are quick, 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 transformation from LR images to HR images by using training data. Dictionary learning was exploited for single image SR [2], [3] in the past. Recently, SR methods based on convolutional neural networks (CNNs) [4]–[16] are mainly studied 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 generates SR images with three convolution layers from *interpolated LR images* that are enlarged to the HR image size by a bicubic interpolation. There exist many improved versions [5]–[16] of SRCNN. For example, VDSR [6] increased the number of the convolution layers by introducing the *residual learning* to resolve the gradient vanishing. ESPCN [7] proposed a sub-pixel convolution layer, called the *pixel shuffler*, that enlarges LR images at various magnification ratios and removes the bicubic interpolation used in the input layer of SRCNN. In each SR method [4]–[16], a single CNN, called a *generator*, is trained by minimizing mainly the mean square error (MSE) or the mean absolute error (MAE) between true HR images and the output SR images of the generator.

It is well-known that SR images generated from 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. To generate realistic high-frequency components, Ledig et al. proposed SRGAN [17] that uses a generative adversarial network (GAN) [22]. SRGAN is composed of two CNNs used as a *generator* and a *discriminator*. The discriminator is trained to judge whether the input image is a true HR image or a generated SR image. Since the generator tries to induce the discriminator's misjudgment, SR images having realistic high-frequency components can be generated. However, in the process of inducing the discriminator's misjudgment, some *artifacts* that do not exist in true HR images are also generated, and MSE and MAE increase. In addition, although the downscaling results of the true HR images are the input LR images, the re-downscaling results of the SR images tend to be different from the LR images.

To reduce artifacts in SR images of GAN-based methods, this paper proposes to consider the consistency of the SR

Manuscript received April 26, 2021.

Manuscript revised August 14, 2021.

Manuscript publicized September 29, 2021.

[†]The authors are with the Dept. of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525-8577 Japan.

^{*}This paper is an extended version of our previous paper presented at the 28th EUSIPCO, October 2020 [1]. The previous paper only addressed noise-free input images while the current paper extends the proposed projection technique to noisy input images.

a) E-mail: d-kita@media.ritsumei.ac.jp (Corresponding author)

b) E-mail: kuroda@media.ritsumei.ac.jp

c) E-mail: akirahrb@media.ritsumei.ac.jp

DOI: 10.1587/transfun.2021EAP1038

images for given LR inputs in two different situations. One is a situation where the LR images are *noise-free*, and the other is a situation where the LR images are *contaminated by additive Gaussian noise*. In the *noise-free* situation, we consider the set of all SR images whose downscaling results are the same as the input LR image. This set is a linear manifold, and we propose to modify the output into an SR image in this set with the minimum moving distance. This proposed modification is represented as the *orthogonal projection* onto the linear manifold in the output layer of a generator, and easily computed as in [23] under a simple blockwise downscaling model in (1) (see Sect. 3.1). The proposed method can be applied to any generator, and SR images having the perfect consistency with the input LR images are always generated. Since a true HR image has the perfect consistency with the LR image obviously, the orthogonal projection brings the SR image closer to the true HR image (see Fig. 3 in Sect. 4).

On the other hand, in the *noisy* situation, if we directly apply the above orthogonal projection, then noisy SR images are generated because the perfect consistency with the noisy inputs is imposed on them. Instead, based on the observation that the energy of noise will be within a certain range with high probability (see Fig. 5 in Sect. 5), we consider the set of all SR images whose square errors between downscaling results and the noisy LR image are within this range. Though this *consistent set* is nonconvex, we can give the exact *metric projection* onto it under the simple blockwise downscaling model. Thus, we propose to apply this projection in the output layer of a generator. Since a true HR image belongs to the consistent set with high probability, the projection brings the SR image closer to the HR image with high probability.

For both noise-free and noisy situations, the projection can appropriately guarantee the consistency between output SR and input LR images with the minimum moving distance. Hence, generators of GAN-based SR methods with the projection create high-frequency components while maintaining low-frequency ones and suppressing noise and artifacts. Moreover, to generate SR images as close as possible to the consistent sets before the projections, we further propose to add the total distance modified by the projections to the loss function as the *projection loss*. By decreasing the value of the projection loss, the moving distance by each projection is further reduced, which facilitates training of the generators under the consistency. Simulations in Sect. 6 demonstrate that the proposed technique reduces artifacts included in the original SR images of a GAN-based SR method while generating realistic high-frequency components with better values for objective indices in both noise-free and noisy situations.

This paper is concluded in Sect. 7 and the main contributions of this paper are summarized as follows.

1. We define the consistent sets to which HR images belong with high probability, according to the noise level.
2. We apply the exact projections to SR images onto the consistent sets, and add the projection loss in training.
3. We show the effectiveness of the proposed technique integrated into a GAN-based SR method by simulations.

2. Relation to Existing Works

As reviewed in detail in [24], various SR methods based on CNNs have been proposed, and each method has different advantages and limitations. To improve the SR performance, three main research directions[†]: (i) changing the network architecture, (ii) changing the loss function, and (iii) integrating other learning strategies, are considered. It is noteworthy that the proposed technique can be integrated into most SR methods without contradiction since the metric projection is applied to an SR image as the *post-processing* in the output layer and the projection loss is just added to the loss function. The proposed technique is expected to be especially effective for GAN-based SR methods [17]–[21] because we can give them the consistency with the input LR images, which is almost automatically satisfied in single-CNN-based SR methods minimizing MSE or MAE (see simulations in Sect. 6).

To improve the image quality of SR images of SRGAN, EUSR-PCL [18], ESRGAN [19], and RCA-GAN [20] propose different network architectures and loss functions, but they do not evaluate the consistency with given input images. PULSE [21] evaluates errors between the re-downscaling results of SR images and the input images as the *downscaling loss* that is also proposed in [25], [26]. Although the downscaling loss is the same as the projection loss in the noise-free situation (see (15) in Sect. 4), PULSE does not apply the projection and thus cannot guarantee the perfect consistency.

3. Single Image Super-Resolution via Neural Networks

3.1 Formulation of Downscaling without Noise

Let $Y := (Y_{i,j,c}) \in [0, 1]^{I \times J \times C}$ be a low-resolution (LR) image to be enlarged, and let $Y_{i,j,c} \in [0, 1]$ be the (i, j) -th pixel value ($i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$) of color channel c . If Y is a grayscale image, then $C = 1$ and $c = 1$. If Y is an RGB color image, then $C = 3$ and $c = 1, 2, 3$. Suppose that Y is given as the downscaling result of a high-resolution (HR) image $X := (X_{i,j,c}) \in [0, 1]^{IK \times JL \times C}$ with slight anti-aliasing (i.e., slight blurring only within blocks of KL pixels) by

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

where K and L are supposed to be integers larger than 1, and $w_{k,l} \in \mathbb{R}$ are downscaling weights^{††} s.t. $\sum_{k=1}^K \sum_{l=1}^L w_{k,l} = 1$.

[†]Network components, loss functions, keywords and the performance of major methods are summarized in [24, Table 2 and Fig. 8].

^{††}If downscaling is the nearest-neighbor or bilinear interpolation, then (1) is always satisfied. If downscaling is a certain bicubic interpolation and both K and L are equal to or larger than 4, then (1) is also satisfied, but some weights can be negative. However, if downscaling is a bicubic interpolation and K or L is equal to 2 or 3, then (1) is not satisfied. If downscaling is the natural bicubic spline interpolation, then (1) is never satisfied. If standard anti-aliasing, i.e., overlapped blurring as in [27], is done right before interpolation process, then (1) will be an approximated downscaling model.

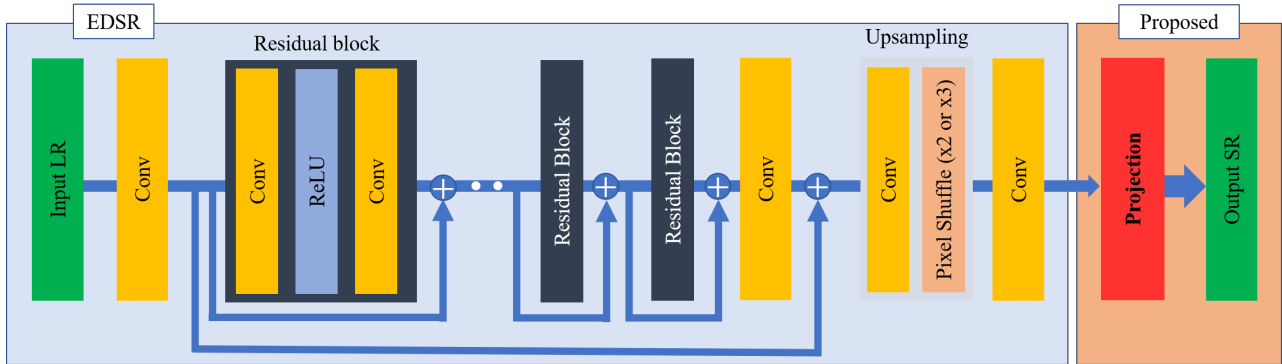


Fig. 1 Proposed generator model based on EDSR. The input is an LR image and the output is an SR image projected onto the consistent set.

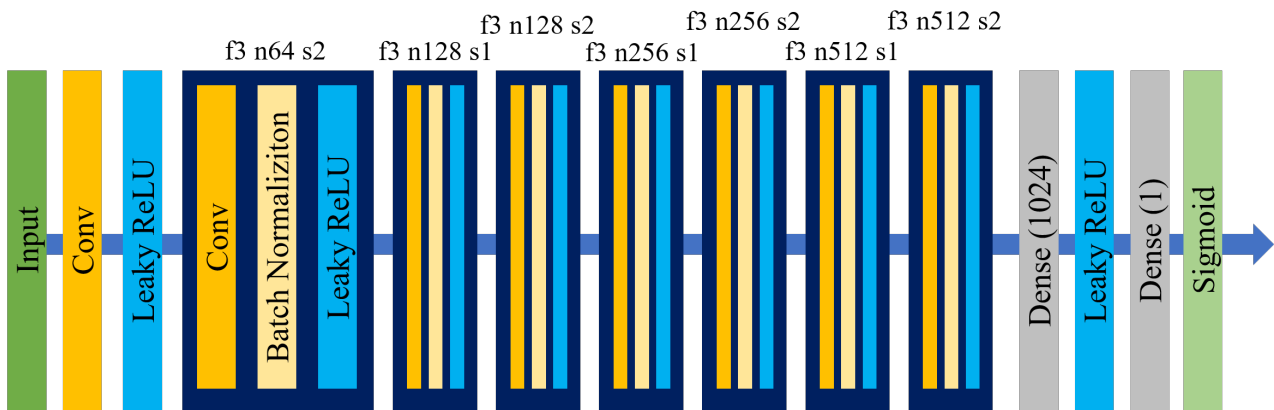


Fig. 2 Discriminator model used in SRGAN. The input is an HR or SR image and the output value is restricted in (0, 1) by a sigmoid function.

Define $\mathbf{y} \in [0, 1]^{CIJ}$ and $\mathbf{x} \in [0, 1]^{CIJKL}$ as the vectorized versions of the LR image Y and the HR image X , respectively. By using a *block-diagonal-like* matrix $A \in \mathbb{R}^{CIJ \times CIJKL}$, the downscaling model in (1) is expressed as a matrix form[†] by

$$\mathbf{y} = A\mathbf{x}. \quad (2)$$

3.2 Super-Resolution via Convolutional Neural Networks

Image super-resolution (SR) based on a single convolutional neural network (CNN) has been studied by many researchers [4]–[16], after the first end-to-end CNN work of SRCNN [4]. In particular, SRResNet proposed by Ledig et al. as a *generator* of SRGAN [17] (see also Sect. 3.3 below) is often adopted as a baseline. SRResNet utilizes the architecture of ResNet [28] that was developed originally for image recognition, and has many *residual blocks* composed of two convolution layers, two batch normalizations, and one rectified linear unit (ReLU). At the end of each residual block, the input value of the residual block is added for *residual learning* that enables CNNs to avoid the gradient vanishing problem.

[†]Although many downscaling methods, including any bicubic interpolation with K and L smaller than 4 and the natural bicubic spline interpolation, are expressed as linear operators A as in (2), the condition in (1) for downscaling is important to easily compute the metric projections onto the consistent sets as in (10) and (24).

Lim et al. proposed EDSR [8] as an improved version of SRResNet, and EDSR is famous as the champion in the NTIRE 2017 Challenge on Single Image Super-Resolution [29]. In EDSR, the batch normalizations are removed since they reduce the flexibility of CNNs and use a lot of memory. The architecture of EDSR is shown in the left blue box of Fig. 1. In this paper, instead of SRResNet, we use EDSR as a baseline method and a *generator* of a GAN-based method. Besides EDSR, EUSR [9], D-DBPN [10], RCAN [11], SAN [12], RFANet [13], USRNet [14], DRN [15], and MADNet [16] are also known as single-CNN-based SR methods.

Let $\{(\mathbf{y}_n, \mathbf{x}_n)\}_{n=1}^N$ be training data composed of N pairs of LR and true HR images. Let $\hat{\mathbf{x}}_n \in \mathbb{R}^{CIJKL}$ be the outputs, called SR images, of a certain SR network for the input LR images $\mathbf{y}_n \in [0, 1]^{CIJ}$. As a loss function to be minimized for training of the SR network, the mean square error (MSE)

$$l_{\text{MSE}} = \frac{1}{CIJKLN} \sum_{n=1}^N \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_2^2 \quad (3)$$

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

$$l_{\text{MAE}} = \frac{1}{CIJKLN} \sum_{n=1}^N \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_1 \quad (4)$$

leads to slightly better results [8], where $\|\cdot\|_1$ is the ℓ_1 norm.

3.3 Super-Resolution via Generative Adversarial Networks

Ledig et al. proposed the first SR method using a generative adversarial network (GAN) [22], and this was named SRGAN [17]. Conventional methods [4]–[16] train a single CNN by minimizing, e.g., (3) or (4), but they cannot sufficiently reconstruct high-frequency components of the true HR images. On the other hand, SRGAN is composed of two CNNs, i.e., a *generator* and a *discriminator*, and can create realistic high-frequency components, although they might be different from the original ones, which are seen as *artifacts*.

In SRGAN, the generator $G : \mathbb{R}^{CIJ} \rightarrow \mathbb{R}^{CIJKL}$, i.e., SRResNet, and the discriminator $D : \mathbb{R}^{CIJKL} \rightarrow (0, 1)$ are alternately updated. Figure 2 shows the architecture of the discriminator. The discriminator D judges whether the input is a true HR image or a generated SR image by maximizing

$$l_D = \frac{1}{N} \sum_{n=1}^N \log(D(\mathbf{x}_n)) + \frac{1}{N} \sum_{n=1}^N \log(1 - D(\hat{\mathbf{x}}_n)). \quad (5)$$

Here, we call l_D the *discrimination loss*, and $\hat{\mathbf{x}}_n := G(\mathbf{y}_n) \in \mathbb{R}^{CIJKL}$ is an SR image generated from the LR image \mathbf{y}_n by the generator G at the current iteration. In the standard GAN techniques, the second term $\frac{1}{N} \sum_{n=1}^N \log(1 - D(\hat{\mathbf{x}}_n))$ in (5) is used as a loss function to train G , but in the context of SR,

$$l_A = -\frac{1}{N} \sum_{n=1}^N \log(D(\hat{\mathbf{x}}_n)) \quad (6)$$

is also used as the *adversarial loss* due to its better gradient behavior [17]. As a result, in training of the generator G , the total loss function to be minimized is expressed as

$$l_G = l_C + \kappa l_A, \quad (7)$$

where l_C is the *content loss* evaluating the consistency between $\hat{\mathbf{x}}_n$ and \mathbf{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) and directly evaluates errors between $\hat{\mathbf{x}}_n$ and \mathbf{x}_n . In more complicated cases [17], [18], the content loss l_C is defined, e.g., as MSE of the pre-trained VGG feature maps [30] or MAE of the differential images.

4. Orthogonal Projection in the Noise-Free Situation

In the conventional loss function in (7), the content loss l_C mainly evaluates differences between generated SR images $\hat{\mathbf{x}}_n$ and target HR images \mathbf{x}_n . However, it is not considered whether *re-downscaling results* $A\hat{\mathbf{x}}_n$ are close to given LR images \mathbf{y}_n or not. On the other hand, the true HR images \mathbf{x}_n always satisfy $A\mathbf{x}_n = \mathbf{y}_n$ in the noise-free situation. To make the most of training data, this paper also considers errors between the re-downscaling results $A\hat{\mathbf{x}}_n$ and the LR images \mathbf{y}_n .

More concretely, we propose a modification technique which enables *any generator* to generate SR images $\check{\mathbf{x}}_n$ satisfying $A\check{\mathbf{x}}_n = \mathbf{y}_n$ in the noise-free situation. The proposed

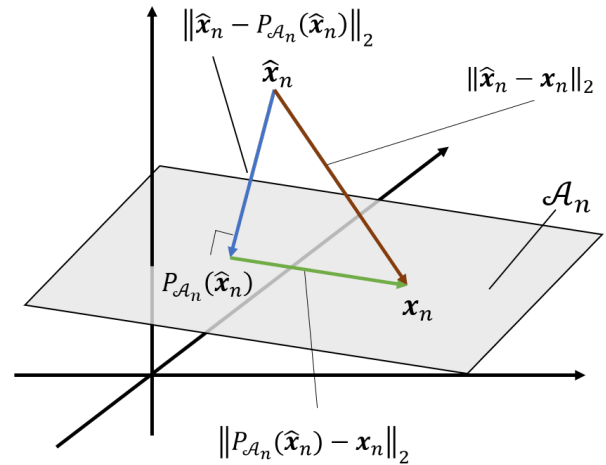


Fig. 3 Linear manifold \mathcal{A}_n in (8) and errors horizontal/vertical to \mathcal{A}_n .

modification is expected to be especially effective for GAN-based SR methods [17]–[21] since each generator can learn high-frequency components while keeping the original low-frequency ones, i.e., information on the input LR images \mathbf{y}_n .

First, we define the set of all SR images whose downscaling results are the same as the input LR image \mathbf{y}_n by

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

where $\mathcal{N}(A)$ is the null space of the matrix A . From (8), it is found that the set \mathcal{A}_n is a *linear manifold*, and the true HR image \mathbf{x}_n always belongs to \mathcal{A}_n . As shown in the right red box of Fig. 1, we propose to apply the *orthogonal projection* $P_{\mathcal{A}_n} : \mathbb{R}^{CIJKL} \rightarrow \mathcal{A}_n$ onto \mathcal{A}_n to the conventional output image $\hat{\mathbf{x}}_n$ of some generator. The proposed SR image $\check{\mathbf{x}}_n := P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n)$ always has the perfect consistency as $A\check{\mathbf{x}}_n = \mathbf{y}_n$. Note that the word “orthogonal projection” strictly refers to the *metric projection* onto a subspace, but here we use the same word for the metric projection onto the linear manifold because \mathcal{A}_n is a shifted version of the subspace $\mathcal{N}(A)$ and a *generalized Pythagorean equation* holds (see (11) below).

The proposed SR image $\check{\mathbf{x}}_n$ is concretely expressed as

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

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

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

Figure 3 shows the relation between \mathbf{x}_n , $\hat{\mathbf{x}}_n$, and $\check{\mathbf{x}}_n$. As shown in Fig. 3, MSE between $\hat{\mathbf{x}}_n$ and \mathbf{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

$$\|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_2^2 = \|\hat{\mathbf{x}}_n - P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n)\|_2^2 + \|P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n) - \mathbf{x}_n\|_2^2 \quad (11)$$

$$\geq \|P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n) - \mathbf{x}_n\|_2^2 = \|\check{\mathbf{x}}_n - \mathbf{x}_n\|_2^2. \quad (12)$$

From (12), if $A\hat{\mathbf{x}}_n \neq \mathbf{y}_n$, then MSE always becomes smaller, i.e., PSNR always improves, by applying the orthogonal projection $P_{\mathcal{A}_n}$. Since the output image is changed from $\hat{\mathbf{x}}_n$ to $\check{\mathbf{x}}_n = P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n)$, we also change, e.g., the MSE content loss to

$$l_C = \frac{1}{CIJKLN} \sum_{n=1}^N \|P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n) - \mathbf{x}_n\|_2^2. \quad (13)$$

When we only consider a weighted sum of l_C in (13) and l_A in (6) as the total loss function l_G for a generator G , in fact SR results are *not good* since error information of the vertical components to \mathcal{A}_n is lost. Thus, to generate SR images as close as possible to the linear manifold \mathcal{A}_n before the orthogonal projection, we further propose to add MSE of the vertical components to \mathcal{A}_n as the *projection loss*

$$l_P = \frac{1}{CIJKLN} \sum_{n=1}^N \|\hat{\mathbf{x}}_n - P_{\mathcal{A}_n}(\hat{\mathbf{x}}_n)\|_2^2. \quad (14)$$

Note that, under the downscaling model in (1), the projection loss l_P is also expressed as

$$\begin{aligned} l_P &= \frac{1}{CIJKLN (\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2)^2} \sum_{n=1}^N \|A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n)\|_2^2 \\ &= \frac{1}{CIJKLN \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} \sum_{n=1}^N \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2. \end{aligned} \quad (15)$$

From (15), we can see that the projection loss is essentially the same as the *downscaling loss* in [21], i.e., MSE between the re-downscaling results $A\hat{\mathbf{x}}_n$ and the input LR images \mathbf{y}_n . Finally, the total loss function for the generator G is defined as

$$l_G = l_C + \lambda l_P + \kappa l_A, \quad (16)$$

where $\lambda > 0$ and $\kappa > 0$. Since the content loss l_C in (13) is more important than the projection loss l_P in (14), we recommend to set a value smaller than 1 for λ . We evaluate the horizontal and vertical MSEs at a ratio of 1 : λ in (16) while they have been equally evaluated at a ratio of 1 : 1 in (7).

5. Metric Projection in the Noisy Situation

5.1 Formulation of Downscaling with Gaussian Noise

In this section, we extend the proposed projection technique in the previous section to noisy input images. Specifically, we consider the following noisy downscaling model

$$\mathbf{y} = \mathcal{T}(A\mathbf{x} + \boldsymbol{\varepsilon}) = A\mathbf{x} + \tilde{\boldsymbol{\varepsilon}}, \quad (17)$$

where $\mathcal{T} : \mathbb{R}^{CIJ} \rightarrow [0, 1]^{CIJ}$ denotes the threshold operator that replaces negative values with 0 and values greater than 1 with 1, and $\boldsymbol{\varepsilon} \in \mathbb{R}^{CIJ}$ is additive white Gaussian noise of variance σ^2 . Figure 4 shows a noisy LR color image based



(a) LR image without noise. (b) LR image with Gaussian noise.

Fig. 4 LR images without and with white Gaussian noise of $\sigma = \frac{5}{255}$.

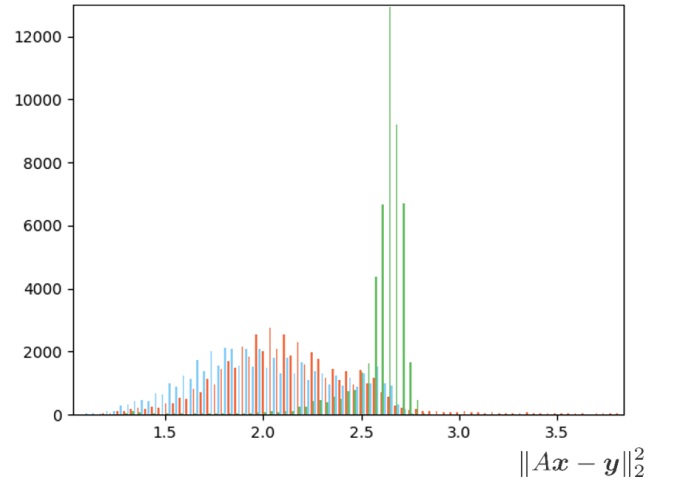


Fig. 5 Histograms of $\|A\mathbf{x}_n - \mathbf{y}_n\|_2^2$ and $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2$. Green one shows the true distribution of the energy of noise. Blue and red ones show the distributions of square errors between outputs' re-downscaling results and noisy input images for single-CNN-based and GAN-based methods, respectively.

on (17) with $\sigma = \frac{5}{255}$. In numerical experiments of Sect. 6, we use the same standard deviation for each component of $\boldsymbol{\varepsilon}$.

Figure 5 shows the distribution of the energy of noise $\tilde{\boldsymbol{\varepsilon}}$, i.e., $\|\tilde{\boldsymbol{\varepsilon}}_n\|_2^2 = \|A\mathbf{x}_n - \mathbf{y}_n\|_2^2$ ($n = 1, 2, \dots, N$), as a green histogram, where the image sizes of the LR color images \mathbf{y}_n are 48×48 and the number of the images is $N = 50,736$. From $(\frac{5}{255})^2 \times 48 \times 48 \times 3 \approx 2.657$, the green histogram has a peak around 2.65. If there was no thresholding, values of $\frac{1}{\sigma^2} \|\tilde{\boldsymbol{\varepsilon}}_n\|_2^2 = \frac{1}{\sigma^2} \|\boldsymbol{\varepsilon}_n\|_2^2$ would follow the *chi-square distribution with CIJ degrees of freedom*. However, in fact there is the threshold operator, and the absolute values of noise components are suppressed. As a result, the distribution of $\|\tilde{\boldsymbol{\varepsilon}}_n\|_2^2$ goes away from symmetry and has a long tail in the left side.

5.2 Consistent Set for the Noisy Low-Resolution Input

As stated in [31], to maximize the performance, we should build CNNs that simultaneously perform denoising and SR. In the noisy situation, SR images generated by single-CNN-based methods become further over-smoothed for denoising, while SR images generated by GAN-based methods tend to include more artifacts. Figure 5 also shows the distributions

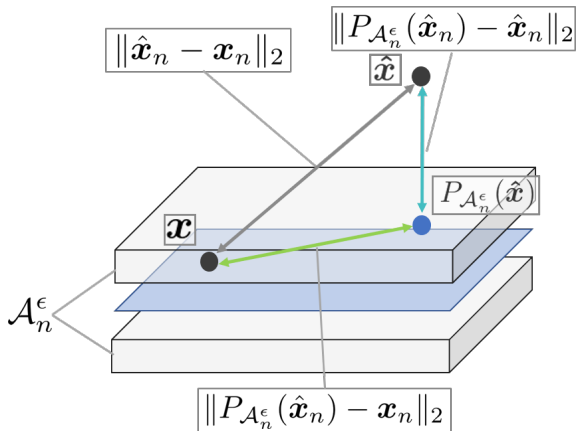


Fig. 6 Consistent set \mathcal{A}_n^ϵ in (18) and errors before/after the projection.

of square errors between outputs' re-downscaling results and noisy input LR images, i.e., $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2$ ($n = 1, 2, \dots, N$), as a blue histogram for a single-CNN-based method (EDSR) and a red histogram for a GAN-based method (EDSR-GAN). Note that we used the MAE content loss $l_C = l_{\text{MAE}}$ in (4) for both methods. From Fig. 5, we can see that both blue and red histograms of $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2$ have peaks located to the left of the green histogram's peak and also possess larger variances than the green histogram of $\|A\mathbf{x}_n - \mathbf{y}_n\|_2^2$. Furthermore, the red histogram of the GAN-based method has a very long tail in the right side, which has led to the serious artifacts.

If we directly apply the orthogonal projection in (10), SR images having the perfect consistency $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2 = 0$ are generated. However, this direct approach keeps noise $\tilde{\mathbf{e}}_n$ included in the input LR images \mathbf{y}_n . From Fig. 5, we can see that the energy of noise $\|\tilde{\mathbf{e}}_n\|_2^2 = \|A\mathbf{x}_n - \mathbf{y}_n\|_2^2$ will be within a certain range with high probability, e.g., from 2 to 2.8 with 98%. Based on this observation for training data, we try to improve the image quality by making the values $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2^2$ closer to a range pre-defined from the distribution of $\|\tilde{\mathbf{e}}_n\|_2^2$.

We define the set of all SR images whose root square errors between downscaling results and the noisy LR image \mathbf{y}_n are within a consistent range $[\epsilon_{\text{lb}}, \epsilon_{\text{ub}}]$ ($0 \leq \epsilon_{\text{lb}} \leq \epsilon_{\text{ub}}$) by

$$\mathcal{A}_n^\epsilon := \{\mathbf{x} \in \mathbb{R}^{CIJKL} \mid \epsilon_{\text{lb}} \leq \|A\mathbf{x} - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}\}. \quad (18)$$

We call this set a *consistent set*, where ϵ_{lb} and ϵ_{ub} are lower and upper bounds of the root square errors and are set, e.g., to $\epsilon_{\text{lb}} = \sqrt{2}$ and $\epsilon_{\text{ub}} = \sqrt{2.8}$ from Fig. 5, so that the true HR image \mathbf{x}_n will belong to \mathcal{A}_n^ϵ with high probability. If we set $\epsilon_{\text{lb}} = \epsilon_{\text{ub}} = 0$, then \mathcal{A}_n^ϵ in (18) becomes \mathcal{A}_n in (8). As long as $\epsilon_{\text{lb}} > 0$, the consistent set \mathcal{A}_n^ϵ is *nonconvex* as illustrated in Fig. 6, but we can give the exact *metric projection* onto \mathcal{A}_n^ϵ . We explain the metric projection in the next section.

5.3 Exact Metric Projection onto the Consistent Set

To derive the exact metric projection onto the consistent set in (18), we further define two sets of SR images. One is the set of all SR images whose downscaling errors are under ϵ_{ub}

$$\mathcal{A}_n^{\text{ub}} := \{\mathbf{x} \in \mathbb{R}^{CIJKL} \mid \|A\mathbf{x} - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}\}, \quad (19)$$

and the other is the set whose downscaling errors are over ϵ_{lb}

$$\mathcal{A}_n^{\text{lb}} := \{\mathbf{x} \in \mathbb{R}^{CIJKL} \mid \|A\mathbf{x} - \mathbf{y}_n\|_2 \geq \epsilon_{\text{lb}}\}. \quad (20)$$

The set $\mathcal{A}_n^{\text{ub}}$ in (19) is convex, but the set $\mathcal{A}_n^{\text{lb}}$ in (20) is non-convex. Moreover, the intersection of $\mathcal{A}_n^{\text{ub}}$ and $\mathcal{A}_n^{\text{lb}}$ coincides with the consistent set \mathcal{A}_n^ϵ in (18), i.e., $\mathcal{A}_n^{\text{ub}} \cap \mathcal{A}_n^{\text{lb}} = \mathcal{A}_n^\epsilon$. Let $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}$ and $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}$ be the metric projections onto $\mathcal{A}_n^{\text{ub}}$ and $\mathcal{A}_n^{\text{lb}}$, respectively. Then, the metric projection $P_{\mathcal{A}_n^\epsilon}$ onto the consistent set \mathcal{A}_n^ϵ is expressed as

$$P_{\mathcal{A}_n^\epsilon}(\hat{\mathbf{x}}_n) = \begin{cases} P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 > \epsilon_{\text{ub}}, \\ P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n) & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}, \\ \hat{\mathbf{x}}_n & \text{if } \epsilon_{\text{lb}} \leq \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}. \end{cases} \quad (21)$$

In general, exact computations of $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}$ and $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}$ are very difficult [32]. Thanks to the blockwise downscaling model in (1), we can give the exact metric projection $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}$ onto $\mathcal{A}_n^{\text{ub}}$ by

$$P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) = \begin{cases} \hat{\mathbf{x}}_n - \frac{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n) & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 > \epsilon_{\text{ub}}, \\ \hat{\mathbf{x}}_n & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}. \end{cases} \quad (22)$$

If we set $\epsilon_{\text{ub}} = 0$, then $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}$ in (22) becomes $P_{\mathcal{A}_n}$ in (10). We can also give the exact metric projection $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}$ onto $\mathcal{A}_n^{\text{lb}}$ by

$$P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n) = \begin{cases} \hat{\mathbf{x}}_n + \frac{\epsilon_{\text{lb}} - \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n) & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}, \\ \hat{\mathbf{x}}_n & \text{if } \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \geq \epsilon_{\text{lb}}. \end{cases} \quad (23)$$

Both metric projections in (22) and (23) move the vector $\hat{\mathbf{x}}_n$ along the orientation of $A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n)$, but their moving directions are opposite to each other because the sign of the coefficient of $A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n)$ is negative in (22) and positive in (23). The proofs of (22) and (23) are shown in Appendices.

Note that we implicitly assume $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 > 0$ in (23) because an SR image $\hat{\mathbf{x}}_n$ exactly satisfying $A\hat{\mathbf{x}}_n = \mathbf{y}_n$ is not created by a generator G practically. In addition, if we define a function $\delta_n^\epsilon : \mathbb{R}^{CIJKL} \rightarrow \mathbb{R}$, that is well-defined and differentiable except for the points \mathbf{x} satisfying $A\mathbf{x} = \mathbf{y}_n$, as

$$\delta_n^\epsilon(\mathbf{x}) := \begin{cases} \frac{\|A\mathbf{x} - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\|A\mathbf{x} - \mathbf{y}_n\|_2} & \text{if } \|A\mathbf{x} - \mathbf{y}_n\|_2 > \epsilon_{\text{ub}}, \\ \frac{\|A\mathbf{x} - \mathbf{y}_n\|_2 - \epsilon_{\text{lb}}}{\|A\mathbf{x} - \mathbf{y}_n\|_2} & \text{if } \|A\mathbf{x} - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}, \\ 0 & \text{if } \epsilon_{\text{lb}} \leq \|A\mathbf{x} - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}, \end{cases}$$

then the metric projection $P_{\mathcal{A}_n^\epsilon}(\hat{\mathbf{x}}_n)$ in (21) is also written by

$$P_{\mathcal{A}_n^\epsilon}(\hat{\mathbf{x}}_n) = \hat{\mathbf{x}}_n - \frac{\delta_n^\epsilon(\hat{\mathbf{x}}_n)}{\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T(A\hat{\mathbf{x}}_n - \mathbf{y}_n). \quad (24)$$

5.4 Proposed Output Image and the Total Loss Function

Since the consistent set \mathcal{A}_n^ϵ is nonconvex, there is a possibility that the metric projection $P_{\mathcal{A}_n^\epsilon}$ increases errors between the SR image and the true HR image when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}$. On the other hand, when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 > \epsilon_{\text{ub}}$, the metric projection $P_{\mathcal{A}_n^\epsilon}(\hat{\mathbf{x}}_n) = P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)$ almost always decreases MSE because $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}$ is a *nonexpansive mapping* and satisfies

$$\|P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) - \mathbf{x}_n\|_2^2 = \|P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) - P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\mathbf{x}_n)\|_2^2 \leq \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_2^2$$

if $\mathbf{x}_n \in \mathcal{A}_n^{\text{ub}}$ holds. Based on this, we propose to use $\check{\mathbf{x}}_n := P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)$ as the final output SR image in the noisy situation, which means that we directly use $\hat{\mathbf{x}}_n$ as the final output when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}$ including the case of $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}$.

Similar to (13), we change, e.g., the MAE content loss to

$$l_C = \frac{1}{CIJKLN} \sum_{n=1}^N \|P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) - \mathbf{x}_n\|_1. \quad (25)$$

Further, to generate SR images as close as possible to the consistent set \mathcal{A}_n^ϵ before the projection, we propose to add two projection losses. One is the mean square distance to $\mathcal{A}_n^{\text{lb}}$:

$$l_P^{\text{lb}} = \frac{1}{CIJKLN} \sum_{n=1}^N \|\hat{\mathbf{x}}_n - P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n)\|_2^2, \quad (26)$$

and the other is the mean square distance to $\mathcal{A}_n^{\text{ub}}$:

$$l_P^{\text{ub}} = \frac{1}{CIJKLN} \sum_{n=1}^N \|\hat{\mathbf{x}}_n - P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)\|_2^2. \quad (27)$$

Finally, the proposed total loss function for a generator G is

$$l_G = l_C + \lambda_{\text{lb}} l_P^{\text{lb}} + \lambda_{\text{ub}} l_P^{\text{ub}} + \kappa l_A, \quad (28)$$

where $\lambda_{\text{lb}} > 0$ and $\lambda_{\text{ub}} > 0$. From Fig. 5, we find that root square errors between re-downscaling results and noisy input images are often under the lower bound ϵ_{lb} but seldom over the upper bound ϵ_{ub} . Since the latter cases are considered to cause serious artifacts in GAN-based SR methods, we recommend to set λ_{lb} and λ_{ub} so that $\lambda_{\text{lb}} < \lambda_{\text{ub}}$ can hold.

6. Numerical Experiments

In this section, we compare SR images generated by EDSR, EDSR-Projection, EDSR-GAN and EDSR-GAN-Projection. Here, ‘Projection’ means that proposed projection is added in the output layer of the generator (EDSR) as shown in Fig. 1. ‘GAN’ means that the discriminator of SRGAN in Fig. 2 is also trained by maximizing the discrimination loss l_D in (5).

6.1 Datasets and Experimental Settings

We create training image patches from 800 images in DIV2K dataset [33]. We set the sizes of HR color patches to 96×96

for $\times 2$ scale (i.e., $K = L = 2$) and to 144×144 for $\times 3$ scale (i.e., $K = L = 3$). In each case, all LR color patches of size 48×48 are given by downscaling of the HR patches with the *arithmetic mean matrix* A , i.e., $w_{k,l} = \frac{1}{KL}$ for all k and l in (1). For the noisy situation, we add white Gaussian noise of standard deviation $\sigma = \frac{5}{255}$ to the LR patches and then apply the thresholding \mathcal{T} . The number of the training patch pairs is $N = 59,296$ for $\times 2$ scale and $N = 50,736$ for $\times 3$ scale.

We use Adam [34] as the optimizer, where we set $\alpha = 10^{-4}$ in the training of all the generators, $\alpha = 10^{-5}$ only in the training of the discriminators, and $(\beta_1, \beta_2) = (0.9, 0.999)$ in every training. We set the minibatch size to 96 for the single-CNN-based methods and to 64 for the GAN-based methods, where each training patch pair is used either as it is, rotated 90 degrees, flipped vertically, or flipped horizontally when creating minibatches. For stable GAN training, pre-trained EDSR and EDSR-Projection are used as the initial values of the generators of EDSR-GAN and EDSR-GAN-Projection, respectively, and we alternately update each generator and each discriminator using the same minibatch. It takes about 1 day to train each single-CNN-based method and 2 days to train each GAN-based method by PyTorch 1.3.1 on Windows 10 Home (Core i7-7700K, 64 GB, GeForce GTX 1080 Ti).

We evaluate SR images generated by each method for four different datasets, Set5 [35], Set14 [36], BSD100 [37], and Urban100 [38]. We prepare two kinds of test LR images. One is downscaled by the same mean matrix A , and the other by the bicubic interpolation of Pillow [39]. We use PSNR[†] and FSIM_C [40] as objective indices for the image quality.

6.2 Results in the Noise-Free Situation

We use the MSE content loss $l_C = l_{\text{MSE}}$ in (3) for EDSR and EDSR-GAN and l_C in (13) for EDSR-Projection and EDSR-GAN-Projection. In (16), we set the weight of the adversarial loss l_A in (6) to $\kappa = 10^{-3}$ for EDSR-GAN and EDSR-GAN-Projection and the weight of the projection loss l_P in (14) to $\lambda = 10^{-3}$ for EDSR-Projection and EDSR-GAN-Projection.

Tables 1 and 2 summarize SR results for the two kind of LR images, in which red numbers indicate better objective index values on the single-CNN-based methods (EDSR and EDSR-Projection), and blue numbers indicate better values on the GAN-based methods (EDSR-GAN and EDSR-GAN-Projection). For the single-CNN-based methods, there exist slight improvements of the objective indices due to the proposed projection, but which has little effect on human eyes. This is because MSE-based CNNs almost satisfy $A\hat{\mathbf{x}}_n = \mathbf{y}_n$ without the orthogonal projection in the *noise-free* situation if the networks are appropriately trained. On the other hand, for the GAN-based methods, we can see that the proposed projection technique improved PSNR in all cases and FSIM_C mainly for $\times 3$ scale in Tables 1 and 2. As shown in Table 2, the proposed projection was also effective even in the case that the downscaling process is different from training data,

[†]Without using the YCbCr transform, we compute PSNR values from MSE between HR and SR images in the RGB color space.

Table 1 Averages of the objective indices (PSNR/FSIM_C) of SR results for Set5, Set14, BSD100, and Urban100 in the *noise-free* situation. Test images are downsampled by the matrix *A* used in training.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	36.54/0.9738	36.55/0.9739	34.65/0.9618	34.66/0.9576
	×3	34.45/0.9436	34.48/0.9441	33.12/0.9290	33.30/0.9315
Set14	×2	34.10/0.9577	34.14/0.9581	32.74/0.9390	32.82/0.9387
	×3	32.73/0.9098	32.73/0.9097	31.79/0.8967	31.97/0.8987
BSD100	×2	33.91/0.9426	33.91/0.9428	32.61/0.9243	32.62/0.9217
	×3	32.57/0.8838	32.58/0.8835	31.68/0.8761	31.91/0.8783
Urban100	×2	34.08/0.9522	34.10/0.9525	32.38/0.9257	32.63/0.9272
	×3	32.52/0.8958	32.58/0.8959	31.54/0.8760	31.66/0.8792

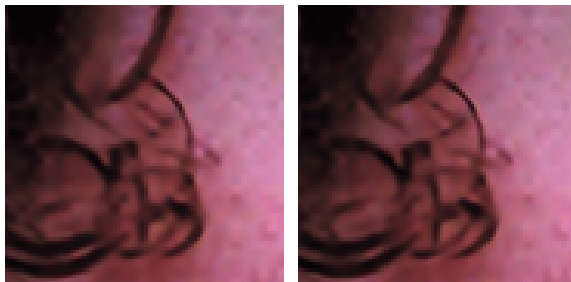
Table 2 Averages of the objective indices (PSNR/FSIM_C) of SR results for Set5, Set14, BSD100, and Urban100 in the *noise-free* situation. Test images are downsampled by the bicubic interpolation.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	36.10/0.9701	36.15/0.9707	34.85/0.9640	35.20/0.9636
	×3	33.97/0.9351	33.98/0.9346	33.19/0.9284	33.35/0.9308
Set14	×2	33.55/0.9437	33.57/0.9442	32.85/0.9351	33.01/0.9366
	×3	32.47/0.8979	32.46/0.8978	31.95/0.8949	32.08/0.8961
BSD100	×2	33.67/0.9373	33.68/0.9379	33.02/0.9295	33.07/0.9295
	×3	32.38/0.8726	32.39/0.8724	31.88/0.8735	32.05/0.8737
Urban100	×2	33.20/0.9340	33.22/0.9345	32.53/0.9246	32.74/0.9262
	×3	32.00/0.8679	32.01/0.8668	31.61/0.8647	31.66/0.8676



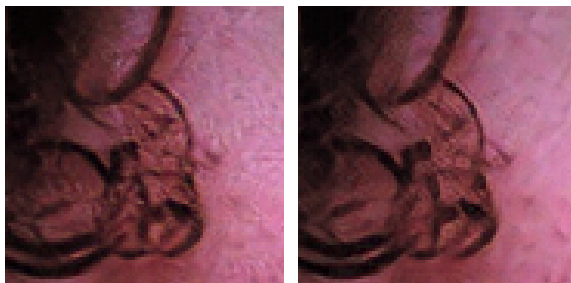
(a) Target HR image.

(b) Enlarged HR image.



(c) EDSR
(32.41/0.9088).

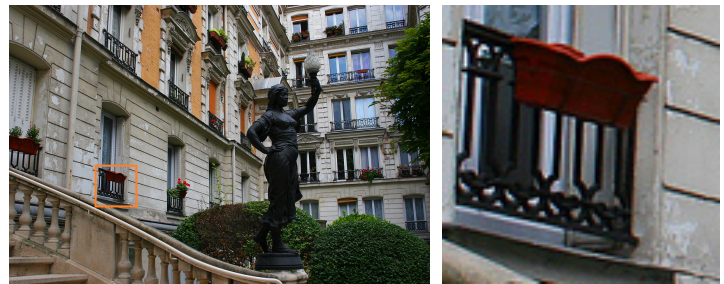
(d) EDSR-Projection
(32.42/0.9087).



(e) EDSR-GAN
(31.76/0.9009).

(f) EDSR-GAN-Projection
(32.00/0.9049).

Fig. 7 Results of ‘head’ in Set5 for ×3 scale (PSNR/FSIM_C).



(a) Target HR image.

(b) Enlarged HR image.



(c) EDSR
(30.70/0.8716).

(d) EDSR-Projection
(30.72/0.8713).



(e) EDSR-GAN
(29.98/0.8507).

(f) EDSR-GAN-Projection
(30.01/0.8525).

Fig. 8 Results of ‘img003’ in Urban100 for ×3 scale (PSNR/FSIM_C).

Table 3 Averages of the objective indices (PSNR/FSIM_C) of SR results for Set5, Set14, BSD100, and Urban100 in the *noisy* situation. Test images are downscaled by the matrix *A* used in training.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	34.80/0.9579	34.80/0.9580	33.78/0.9492	33.79/0.9493
	×3	33.39/0.9282	33.40/0.9288	32.63/0.9222	32.76/0.9233
Set14	×2	33.28/0.9402	33.27/0.9402	32.35/0.9294	32.39/0.9304
	×3	32.21/0.8915	32.23/0.8921	31.56/0.8884	31.64/0.8911
BSD100	×2	33.14/0.9270	33.14/0.9271	32.19/0.9133	32.28/0.9128
	×3	32.08/0.8639	32.09/0.8644	31.50/0.8671	31.44/0.8672
Urban100	×2	33.36/0.9375	33.37/0.9377	32.32/0.9215	32.35/0.9206
	×3	32.14/0.8780	32.15/0.8780	31.52/0.8713	31.53/0.8724

Table 4 Averages of the objective indices (PSNR/FSIM_C) of SR results for Set5, Set14, BSD100, and Urban100 in the *noisy* situation. Test images are downscaled by the bicubic interpolation.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	34.54/0.9546	34.53/0.9542	33.98/0.9505	34.04/0.9514
	×3	33.07/0.9215	33.06/0.9213	32.60/0.9196	32.71/0.9199
Set14	×2	32.87/0.9273	32.86/0.9264	32.45/0.9242	32.44/0.9240
	×3	32.02/0.8816	32.03/0.8824	31.64/0.8849	31.71/0.8864
BSD100	×2	32.95/0.9211	32.94/0.9205	32.50/0.9159	32.54/0.9148
	×3	31.95/0.8541	31.95/0.8542	31.64/0.8637	31.63/0.8635
Urban100	×2	32.72/0.9210	32.72/0.9202	32.36/0.9173	32.38/0.9153
	×3	31.76/0.8540	31.77/0.8547	31.45/0.8569	31.49/0.8577



Fig. 9 Results of 'bird' in Set5 for ×3 scale (PSNR/FSIM_C).



Fig. 10 Results of 'img063' in Urban100 for ×3 scale (PSNR/FSIM_C).

i.e., the proposed technique had a certain robustness against the different downscaling. Interestingly, the objective index values of the single-CNN-based methods in Table 2 are lower than those in Table 1, while some values of the GAN-based methods in Table 2 are better than those in Table 1. In addition, on FSIM_C for $\times 3$ scale of BSD100 in Table 2, EDSR-GAN-Projection gave the best value among all the methods.

Figures 7 and 8 show examples of SR results for the test images. In EDSR and EDSR-Projection, there are almost no differences and both SR results are over-smoothed. In EDSR-GAN, the SR results are more clear but have some artifacts. The proposed GAN-based method (EDSR-GAN-Projection) generates images including realistic high-frequency components while reducing artifacts in Fig. 7(f) and creating clear straight lines in Fig. 8(f) compared with Figs. 7(e) and 8(e).

6.3 Results in the Noisy Situation

We use the MAE content loss $l_C = l_{\text{MAE}}$ in (4) for EDSR and EDSR-GAN and l_C in (25) for EDSR-Projection and EDSR-GAN-Projection because we confirmed that MAE leads to more clear SR images than MSE by preliminary experiments. In (28), we set the weight of the adversarial loss l_A in (6) to $\kappa = 10^{-3}$ for EDSR-GAN and EDSR-GAN-Projection, and the weights of the projection losses l_p^{lb} in (26) and l_p^{ub} in (27) to $\lambda_{\text{lb}} = 1$ and $\lambda_{\text{ub}} = 5$ for $\times 2$ scale of EDSR-Projection and EDSR-GAN-Projection and to $\lambda_{\text{lb}} = 0.2$ and $\lambda_{\text{ub}} = 0.8$ for $\times 3$ scale, from objective and subjective perspectives. In each scale, we set the bounds in (18) to $\epsilon_{\text{lb}} = \sqrt{2}$ and $\epsilon_{\text{ub}} = \sqrt{2.8}$.

Tables 3 and 4 summarize SR results for the two kind of LR images in the *noisy* situation. For the single-CNN-based methods, as in the noise-free situation, there are slight improvements of the objective indices mainly in Table 3, but which has little effect on human eyes in most images. For the GAN-based methods, the trend of improvement of the objective indices, excluding FSIM_C for $\times 2$ scale, can be seen. The objective index values of the single-CNN-based methods in Table 4 are lower than those in Table 3, while some values of the GAN-based methods in Table 4 are better than those in Table 3. On FSIM_C for $\times 3$ scale of BSD100 in Table 3 and of Set14, BSD100, and Urban100 in Table 4, the GAN-based methods were better than the single-CNN-based methods.

Figures 9 and 10 show examples of SR results for the test images. In Fig. 9, the results of EDSR and EDSR-Projection are over-smoothed and almost the same as each other. The result of EDSR-GAN is clear but includes large artifacts in the lower left part of Fig. 9(e). On the other hand, the proposed GAN-based method (EDSR-GAN-Projection) generates the clear image as shown in Fig. 9(f) without the large artifacts. Although the averaged improvements in Tables 3 and 4 are small, from these figures, it can be found that the proposed metric projection can suppress serious artifacts. In Fig. 10, there are significant differences between the results of EDSR and EDSR-Projection. Comparing Figs. 10(c) and 10(d), we can see that the diagonal parallel lines in the roof are more accurately reconstructed. The same matter is seen between the SR images of EDSR-GAN and EDSR-GAN-Projection

in Figs. 10(e) and 10(f). We found that straight lines in other images could also be clearly seen by adding the projection.

6.4 Results for Inputs of Mismatched Noise Level

Although the proposed method assumes that the noise level is known, we verify SR results when the noise level of input LR images differs from that of training data. Specifically, we input *noisy* LR images into the networks in Sect. 6.2, and *noise-free* LR images into the networks in Sect. 6.3, where all the test LR images are downscaled by the mean matrix A used in training. The results and an example of the former experiments are shown in Table 5 and Fig. 11, and those of the latter experiments are shown in Table 6 and Fig. 12.

Comparing Tables 1 and 5, we see that all the values of PSNR and FSIM_C are significantly decreased due to noise. Comparing Tables 3 and 5, the values of PSNR and FSIM_C in Table 5 are lower than those in Table 3 since the networks in Sect. 6.2 do not perform denoising as shown in Fig. 11, whether the proposed projection technique is applied or not. Though noise cannot be removed, the proposed GAN-based method (EDSR-GAN-Projection) generated high-frequency components with fewer artifacts in Fig. 11(f) than Fig. 11(e).

Comparing Tables 3 and 6, for PSNR, all the values in Table 6 are better than those in Table 3 since the LR images in Table 6 are noise-free. On the other hand, for FSIM_C , all the values of the single-CNN-based methods in Table 6 are lower than those in Table 3, while most values of the GAN-based methods in Table 6 are better than those in Table 3. This implies that the single-CNN-based methods are more likely to fail in recover of high-frequency components due to excessive denoising than the GAN-based methods.

Comparing Tables 1 and 6, we see that all the values of PSNR and FSIM_C significantly decrease for the single-CNN-based methods. On the other hand, for the GAN-based methods, most values of PSNR increase, and the decrease in the values of FSIM_C is smaller. Thus, the GAN-based methods were relatively robust to reduction of the noise level in the test images. Moreover, as shown in Fig. 12(f), the property that straight lines become clearly visible by adding the projection was retained even when the noise level was reduced.

To summarize, although it is desired to know how much noise LR images contain, GAN-based SR methods with the proposed projection technique work reasonably well even if the noise level is smaller than expected. On the other hand, if the noise level is much larger than when training CNNs, good results are not obtained as with conventional methods.

7. Conclusion

In this paper, for single image SR using CNNs, we proposed to use the metric projection in the output layer for generating SR images having the appropriate consistency with the input LR images. In the noise-free situation, we considered a linear manifold having the perfect consistency with the input image, and applied the orthogonal projection onto this set just before the output. In the noisy situation, we defined a consistent set,

Table 5 Averages of the objective indices (PSNR / FSIM_C) of SR results when *noisy* images are inputs to networks for the *noise-free* situation. Test images are downscaled by the matrix *A* used in training.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	32.34/0.9343	32.29/0.9341	31.07/0.8990	30.86/0.8908
	×3	31.59/0.9119	31.56/0.9120	30.47/0.8551	30.79/0.8801
Set14	×2	31.52/0.9209	31.52/0.9210	30.35/0.8778	30.31/0.8786
	×3	30.87/0.8807	30.86/0.8804	29.96/0.8358	30.30/0.8546
BSD100	×2	31.35/0.9077	31.32/0.9080	29.87/0.8540	30.09/0.8653
	×3	30.72/0.8568	30.71/0.8567	29.79/0.8150	30.14/0.8360
Urban100	×2	31.38/0.9109	31.36/0.9116	30.13/0.8639	30.22/0.8706
	×3	30.68/0.8572	30.68/0.8576	29.80/0.8093	30.02/0.8256

Table 6 Averages of the objective indices (PSNR / FSIM_C) of SR results when *noise-free* images are inputs to networks for the *noisy* situation. Test images are downscaled by the matrix *A* used in training.

Dataset	Scale	EDSR	EDSR-Projection	EDSR-GAN	EDSR-GAN-Projection
Set5	×2	35.30/0.9567	35.27/0.9564	34.78/0.9543	34.84/0.9557
	×3	33.76/0.9255	33.80/0.9264	33.32/0.9271	33.38/0.9258
Set14	×2	33.58/0.9377	33.56/0.9369	33.04/0.9319	33.05/0.9325
	×3	32.43/0.8880	32.45/0.8887	32.02/0.8897	32.04/0.8909
BSD100	×2	33.42/0.9242	33.41/0.9239	32.88/0.9166	32.87/0.9152
	×3	32.26/0.8576	32.28/0.8579	31.90/0.8668	31.86/0.8669
Urban100	×2	33.61/0.9355	33.62/0.9354	32.95/0.9273	32.86/0.9253
	×3	32.30/0.8761	32.32/0.8768	31.83/0.8751	31.84/0.8754



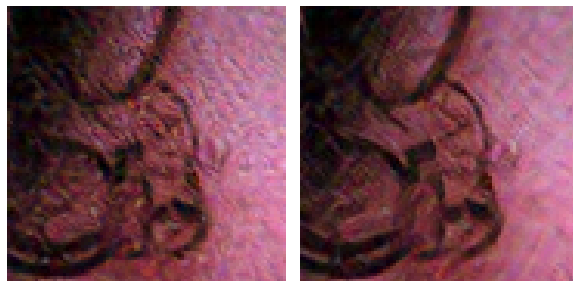
(a) Target HR image.

(b) Enlarged HR image.



(c) EDSR
(31.02/0.8896).

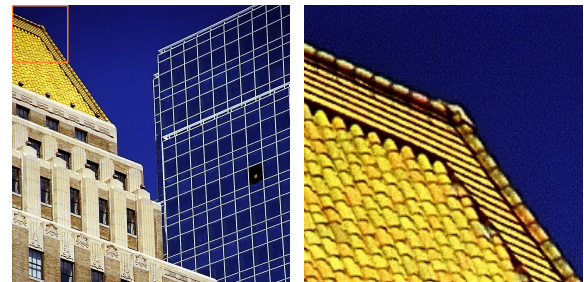
(d) EDSR-Projection
(31.00/0.8896).



(e) EDSR-GAN
(30.32/0.8495).

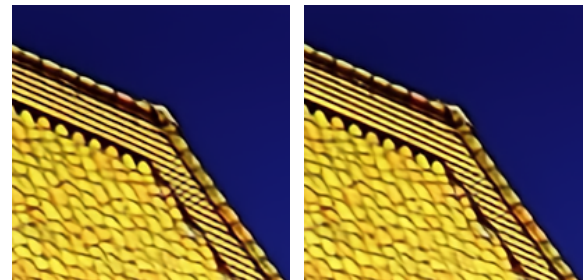
(f) EDSR-GAN-Projection
(30.56/0.8670).

Fig. 11 Outputs from the *noisy* 'head' input for ×3 scale.



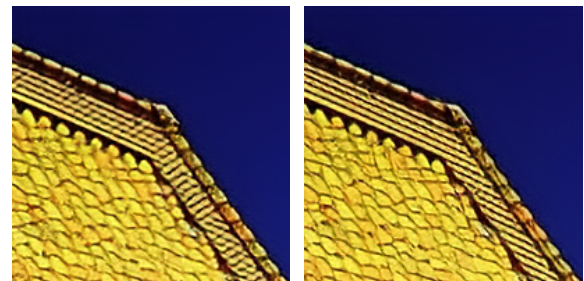
(a) Target HR image.

(b) Enlarged HR image.



(c) EDSR
(29.80/0.8346).

(d) EDSR-Projection
(29.79/0.8323).



(e) EDSR-GAN
(29.63/0.8431).

(f) EDSR-GAN-Projection
(29.63/0.8439).

Fig. 12 Outputs from the *noise-free* 'img063' input for ×3 scale.

which includes the true HR image with high probability, and applied the metric projection onto this set. Furthermore, to generate SR images as close as possible to the consistent sets before the projections, we added the total distance moved by the projections to the loss function. The proposed technique can be integrated into most SR methods without contradiction and especially effective for GAN-based methods. This is because generators can learn high-frequency components while maintaining low-frequency ones appropriately and SR images having realistic high-frequency components with few artifacts are generated. In addition, since the true HR image belongs to the consistent set with high probability, the proposed projection can decrease MSE with high probability.

In numerical experiments, we confirmed the effectiveness of the proposed technique in both noise-free and noisy situations. In each situation, the proposed technique reduced artifacts in a GAN-based method while generating realistic high-frequency components with better values of PSNR and $FSIM_C$, and we found that straight lines could be clearly reconstructed by adding the projection. Moreover, we verified the results for the inputs whose noise level is different from training data, and it was shown that the performance of the GAN-based method with the proposed technique was hardly degraded when the noise level was smaller than training data.

Acknowledgments

This work was supported by JSPS KAKENHI (19K20361).

References

- [1] H. Yamamoto, D. Kitahara, and A. Hirabayashi, "Image super-resolution via generative adversarial network using an orthogonal projection," *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Amsterdam, The Netherlands, pp.660–664, Oct. 2020.
- [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, Nov. 2010.
- [3] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, NY, USA, 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, Feb. 2016.
- [5] C. Dong, C.C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, pp.391–407, Oct. 2016.
- [6] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, pp.1646–1654, June 2016.
- [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," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, pp.1874–1883, June 2016.
- [8] B. Lim, S. Son, H. Kim, S. Nah, and K.M. Lee, "Enhanced deep residual networks for single image super-resolution," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Honolulu, HI, USA, pp.136–144, July 2017.
- [9] J.H. Kim and J.S. Lee, "Deep residual network with enhanced upscaling module for super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Salt Lake City, UT, USA, pp.913–921, June 2018.
- [10] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, pp.1664–1673, June 2018.
- [11] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, pp.286–301, Sept. 2018.
- [12] T. Dai, J. Cai, Y. Zhang, S.T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, pp.11065–11074, June 2019.
- [13] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, pp.2359–2368, June 2020.
- [14] K. Zhang, L.V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, pp.3217–3226, June 2020.
- [15] Y. Guo, J. Chen, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, and M. Tan, "Closed-loop matters: Dual regression networks for single image super-resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, pp.5407–5416, June 2020.
- [16] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo, "MADNet: A fast and lightweight network for single-image super resolution," *IEEE Trans. Cybern.*, vol.51, no.3, pp.1443–1453, Mar. 2021.
- [17] 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," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, pp.105–114, July 2017.
- [18] M. Cheon, J.H. Kim, J.H. Choi, and J.S. Lee, "Generative adversarial network-based image super-resolution using perceptual content losses," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, pp.51–62, Sept. 2018.
- [19] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C.C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, pp.63–79, Sept. 2018.
- [20] J. Cai, Z. Meng, and C.M. Ho, "Residual channel attention generative adversarial network for image super-resolution and noise reduction," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Seattle, WA, USA, pp.1852–1861, June 2020.
- [21] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-supervised photo upsampling via latent space exploration of generative models," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, pp.2434–2442, June 2020.
- [22] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, Canada, pp.2672–2680, Dec. 2014.
- [23] L. Condat and A. Montanvert, "A framework for image magnification: Induction revisited," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Philadelphia, PA, USA, vol. 2, pp. 845–848, Mar. 2005.
- [24] Z. Wang, J. Chen, and S.C.H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell. (Early Access)*, 24 pages, Mar. 2020.
- [25] R. Chen, Y. Qu, K. Zeng, J. Guo, C. Li, and Y. Xie, "Persistent memory residual network for single image super resolution," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Salt Lake City, UT, USA, pp.922–929, June 2018.
- [26] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *Proc.*

- IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, pp.9446–9454, June 2018.
- [27] 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, Apr. 1983.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, pp. 770–778, June 2016.
- [29] R. Timofte, E. Agustsson, L. V. Gool, M.H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, X. Wang, Y. Tian, K. Yu, Y. Zhang, S. Wu, C. Dong, L. Lin, Y. Qiao, C.C. Loy, W. Bae, J. Yoo, Y. Han, J.C. Ye, J.S. Choi, M. Kim, Y. Fan, J. Yu, W. Han, D. Liu, H. Yu, Z. Wang, H. Shi, X. Wang, T.S. Huang, Y. Chen, K. Zhang, W. Zuo, Z. Tang, L. Luo, S. Li, M. Fu, L. Cao, W. Heng, G. Bui, T. Le, Y. Duan, D. Tao, R. Wang, X. Lin, J. Pang, J. Xu, Y. Zhao, X. Xu, J. Pan, D. Sun, Y. Zhang, X. Song, Y. Dai, X. Qin, X.P. Huynh, T. Guo, H.S. Mousavi, T.H. Vu, V. Monga, C. Cruz, K. Egiazarian, V. Katkornik, R. Mehta, A.K. Jain, A. Agarwalla, C.V.S. Praveen, R. Zhou, H. Wen, C. Zhu, Z. Xia, Z. Wang, and Q. Guo, “NTIRE 2017 challenge on single image super-resolution: methods and results,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Honolulu, HI, USA, pp.1110–1121, July 2017.
- [30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 14 pages, May 2015.
- [31] L.W. Kang, C.C. Hsu, B. Zhuang, C.W. Lin, and C.H. Yeh, “Learning-based joint super-resolution and deblurring for a highly compressed image,” *IEEE Trans. Multimedia*, vol.17, no.7, pp. 921–934, July 2015.
- [32] I. Yamada, K. Slavakis, and K. Yamada, “An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques,” *IEEE Trans. Signal Process.*, vol.50, no.5, pp.1091–1101, May 2002.
- [33] E. Agustsson and R. Timofte, “NTIRE 2017 challenge on single image super-resolution: dataset and study,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Honolulu, HI, USA, pp.1122–1131, July 2017.
- [34] D.P. Kingma and J.L. Ba, “Adam: A method for stochastic optimization,” *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 15 pages, May 2015.
- [35] M. Bevilacqua, A. Roumy, C. Guillemot, and M.L.A. Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Surrey, UK, 10 pages, Sept. 2012.
- [36] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” *Int. Conf. Curves Surfaces*, Avignon, France, pp.711–730, June 2010.
- [37] 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,” *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Vancouver, Canada, vol. 2, pp.416–423, July 2001.
- [38] J.B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, pp.5197–5206, June 2015.
- [39] F. Lundh and A. Clark, Pillow: The Friendly Python Imaging Library Fork, <https://pillow.readthedocs.io/en/stable>, accessed Apr. 24, 2021.
- [40] 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, Aug. 2011.
- [41] H.H. Bauschke and P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Space*, 2nd ed., Springer, New York, NY, USA, 2017.
- [42] D. Salas and L. Thibault, “Quantitative characterizations of nonconvex bodies with smooth boundaries in Hilbert spaces via the metric projection,” *J. Math. Anal. Appl.*, vol.494, no.2, 21 pages, Feb. 2021.

Appendix A: Proof of (22)

It is obvious that $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n) = \hat{\mathbf{x}}_n$ holds when $\hat{\mathbf{x}}_n \in \mathcal{A}_n^{\text{ub}}$, i.e., $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \leq \epsilon_{\text{ub}}$ holds. Hence, we consider only the case when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 > \epsilon_{\text{ub}}$ holds. In this case, computation of $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)$ can be seen as the following optimization problem

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) := \|\hat{\mathbf{x}}_n - \mathbf{x}\|_2^2 \\ & \text{subject to} && g(\mathbf{x}) := \|A\mathbf{x} - \mathbf{y}_n\|_2^2 - \epsilon_{\text{ub}}^2 \leq 0 \end{aligned} \quad (\text{A.1})$$

The unique optimal solution to this problem corresponds to $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)$. Here, we define a vector $\mathbf{x}^* \in \mathbb{R}^{CIJKL}$ as

$$\begin{aligned} \mathbf{x}^* &:= \hat{\mathbf{x}}_n - \frac{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T (A\hat{\mathbf{x}}_n - \mathbf{y}_n) \\ &= \hat{\mathbf{x}}_n - \frac{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} A^T (AA^T)^{-1} (A\hat{\mathbf{x}}_n - \mathbf{y}_n). \end{aligned}$$

We only have to show that \mathbf{x}^* is the optimal solution to the *convex* optimization problem in (A.1). Since this problem satisfies the *Slater condition* [41, Proposition 27.21], \mathbf{x}^* is the optimal solution if and only if \mathbf{x}^* satisfies the following Karush–Kuhn–Tucker (KKT) conditions

$$\begin{cases} -\nabla f(\mathbf{x}^*) = \mu \nabla g(\mathbf{x}^*), & (\text{A.2}) \\ g(\mathbf{x}^*) \leq 0, & (\text{A.3}) \\ \mu \geq 0, & (\text{A.4}) \\ \mu g(\mathbf{x}^*) = 0, & (\text{A.5}) \end{cases}$$

where $\mu \in \mathbb{R}$ denotes the dual variable. First, from

$$\begin{aligned} g(\mathbf{x}^*) &= \|A\mathbf{x}^* - \mathbf{y}_n\|_2^2 - \epsilon_{\text{ub}}^2 \\ &= \left\| A\hat{\mathbf{x}}_n - \frac{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} (A\hat{\mathbf{x}}_n - \mathbf{y}_n) - \mathbf{y}_n \right\|_2^2 - \epsilon_{\text{ub}}^2 \\ &= \left\| \frac{\epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} (A\hat{\mathbf{x}}_n - \mathbf{y}_n) \right\|_2^2 - \epsilon_{\text{ub}}^2 = \epsilon_{\text{ub}}^2 - \epsilon_{\text{ub}}^2 = 0, \end{aligned}$$

the conditions in (A.3) and (A.5) are satisfied. Next, from

$$\begin{aligned} -\nabla f(\mathbf{x}^*) &= 2(\hat{\mathbf{x}}_n - \mathbf{x}^*) \\ &= \frac{2(\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}})}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T (A\hat{\mathbf{x}}_n - \mathbf{y}_n) \end{aligned}$$

and

$$\mu \nabla g(\mathbf{x}^*) = 2\mu A^T (A\mathbf{x}^* - \mathbf{y}_n) = \frac{2\mu \epsilon_{\text{ub}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} A^T (A\hat{\mathbf{x}}_n - \mathbf{y}_n),$$

by setting the dual valuable μ to

$$\mu = \frac{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 - \epsilon_{\text{ub}}}{\epsilon_{\text{ub}} \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} > 0,$$

the conditions in (A.2) and (A.4) are satisfied. Since all the KKT conditions hold, we have proven that \mathbf{x}^* is the solution

to the problem in (A·1) and $P_{\mathcal{A}_n^\epsilon}^{\text{ub}}(\hat{\mathbf{x}}_n)$ is expressed as (22).

Appendix B: Proof of (23)

It is obvious that $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n) = \hat{\mathbf{x}}_n$ holds when $\hat{\mathbf{x}}_n \in \mathcal{A}_n^{\text{lb}}$, i.e., $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \geq \epsilon_{\text{lb}}$ holds. Hence, we consider only the case when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}$ holds. In this case, $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n)$ can be seen as the solution to the following optimization problem

$$\left. \begin{array}{l} \underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) := \|\hat{\mathbf{x}}_n - \mathbf{x}\|_2^2 \\ \text{subject to} \quad g(\mathbf{x}) := -\|A\mathbf{x} - \mathbf{y}_n\|_2^2 + \epsilon_{\text{lb}}^2 \leq 0 \end{array} \right\}. \quad (\text{A} \cdot 6)$$

Differently from (A·1), the problem in (A·6) is *nonconvex*, and it is difficult to utilize the KKT conditions to prove (23). In the following, we directly solve the the problem in (A·6).

Let $\mathbf{x}^* \in \mathbb{R}^{CIJKL}$ be the solution to the the problem in (A·6) and define $\mathbf{y}^* := A\mathbf{x}^* \in \mathbb{R}^{CIJ}$. Then, from $g(\mathbf{x}^*) \leq 0$, we have $\|\mathbf{y}^* - \mathbf{y}_n\|_2 \geq \epsilon_{\text{lb}}$. Moreover, \mathbf{x}^* is also given from $\hat{\mathbf{x}}_n$ by applying the metric projection onto a linear manifold

$$\mathcal{A}^* := \{\mathbf{x} \in \mathbb{R}^{CIJKL} \mid A\mathbf{x} = \mathbf{y}^*\}.$$

Therefore, from (10), \mathbf{x}^* can be expressed as

$$\mathbf{x}^* = P_{\mathcal{A}^*}(\hat{\mathbf{x}}_n) = \hat{\mathbf{x}}_n - \frac{1}{\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T (A\hat{\mathbf{x}}_n - \mathbf{y}^*). \quad (\text{A} \cdot 7)$$

From (A·7) and (15), we have

$$\begin{aligned} f(\mathbf{x}^*) &= \|\hat{\mathbf{x}}_n - \mathbf{x}^*\|_2^2 = \frac{1}{(\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2)^2} \|A^T (A\hat{\mathbf{x}}_n - \mathbf{y}^*)\|_2^2 \\ &= \frac{1}{\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} \|A\hat{\mathbf{x}}_n - \mathbf{y}^*\|_2^2. \end{aligned}$$

As a result, the optimization problem on \mathbf{x} in (A·6) can be reduced to an optimization problem on \mathbf{y} as

$$\underset{\mathbf{y}}{\text{minimize}} \quad \|A\hat{\mathbf{x}}_n - \mathbf{y}\|_2 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{y}_n\|_2 \geq \epsilon_{\text{lb}}, \quad (\text{A} \cdot 8)$$

and \mathbf{y}^* coincides with the optimal solution to the problem in (A·8). By shifting the metric projection onto a spherical surface [42], the unique solution \mathbf{y}^* is easily computed as

$$\mathbf{y}^* = \mathbf{y}_n + \frac{\epsilon_{\text{lb}}}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} (A\hat{\mathbf{x}}_n - \mathbf{y}_n) \quad (\text{A} \cdot 9)$$

if $0 < \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 < \epsilon_{\text{lb}}$. From (A·9), we have

$$A\hat{\mathbf{x}}_n - \mathbf{y}^* = -\frac{\epsilon_{\text{lb}} - \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2} (A\hat{\mathbf{x}}_n - \mathbf{y}_n). \quad (\text{A} \cdot 10)$$

By substituting (A·10) into (A·7), we finally have the unique optimal solution \mathbf{x}^* to the problem in (A·6) as

$$\mathbf{x}^* = \hat{\mathbf{x}}_n + \frac{\epsilon_{\text{lb}} - \|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2}{\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 \sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T (A\hat{\mathbf{x}}_n - \mathbf{y}_n),$$

and it has been proven that $P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n)$ is expressed as (23).

Note that when $\|A\hat{\mathbf{x}}_n - \mathbf{y}_n\|_2 = 0$, i.e., $A\hat{\mathbf{x}}_n = \mathbf{y}_n$ holds, the optimization problem in (A·8) has infinite solutions as

$$\mathbf{y}^* = \mathbf{y}_n + \epsilon_{\text{lb}} \mathbf{u},$$

where $\mathbf{u} \in \mathbb{R}^{CIJ}$ is any unit vector s.t. $\|\mathbf{u}\|_2 = 1$. In this case, the optimization problem in (A·6) also has infinite solutions, and the metric projection becomes a set-valued mapping as

$$P_{\mathcal{A}_n^\epsilon}^{\text{lb}}(\hat{\mathbf{x}}_n) = \left\{ \hat{\mathbf{x}}_n + \frac{\epsilon_{\text{lb}}}{\sum_{k=1}^K \sum_{l=1}^L w_{k,l}^2} A^T \mathbf{u} \mid \|\mathbf{u}\|_2 = 1 \right\}.$$

In practice, the condition $A\hat{\mathbf{x}}_n = \mathbf{y}_n$ does not hold, and even if it holds, the proposed projection technique works correctly by selecting an appropriate unit vector \mathbf{u} , e.g., as $\mathbf{u} = \frac{1}{\sqrt{CIJ}} \mathbf{1}$.



Hiroya Yamamoto received his B.E. degree and M.E. degree in computer science from Ritsumeikan University, Shiga, Japan, in 2019 and 2021, respectively. His research interest includes image super-resolution via deep neural network.



Daichi Kitahara received his B.E. degree in computer science in 2012, and his M.E. and Ph.D. degrees in communications and computer engineering in 2014 and 2017 from the Tokyo Institute of Technology, Tokyo, Japan, respectively. Since 2017, he has been an Assistant Professor with the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His current research interest includes signal processing and its applications, inverse problem, optimization theory, and multivariate spline theory.



Hiroki Kuroda received his B.E. degree in computer science, and his M.E. and Ph.D. degrees in information and communications engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2013, 2015, and 2019, respectively. Currently, he is an Assistant Professor with the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His current research interests include signal processing and its applications, nonlinear inverse problem, and sparse optimization.



Akira Hirabayashi received his D.E. degree in computer science in 1999, from the Tokyo Institute of Technology, Tokyo, Japan. From 2013, he is a Professor of the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His research interests include compressed sensing, sampling theory, and signal and image processing. Prof. Hirabayashi is also a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Acoustical Society of Japan (ASJ).