ORIGINAL PAPER

Checkerboard artifacts free convolutional neural networks

YUSUKE SUGAWARA, SAYAKA SHIOTA AND HITOSHI KIYA

It is well-known that a number of convolutional neural networks (CNNs) generate checkerboard artifacts in both of two processes: forward-propagation of upsampling layers and backpropagation of convolutional layers. A condition for avoiding the artifacts is proposed in this paper. So far, these artifacts have been studied mainly for linear multirate systems, but the conventional condition for avoiding them cannot be applied to CNNs due to the non-linearity of CNNs. We extend the avoidance condition for CNNs and apply the proposed structure to typical CNNs to confirm whether the novel structure is effective. Experimental results demonstrate that the proposed structure can perfectly avoid generating checkerboard artifacts while keeping the excellent properties that CNNs have.

I. INTRODUCTION

In this paper, we address the problem of checkerboard artifacts in convolutional neural networks (CNNs) [1]. Recently, CNNs have been widely studied in a variety of computer vision tasks such as image classification [2, 3], semantic segmentation [4, 5], super-resolution (SR) imaging [6, 7], and image generation [8], and they have achieved superior performances. However, CNNs often generate periodic artifacts, referred to as checkerboard artifacts, in both of two processes: forward-propagation of upsampling layers and backpropagation of convolutional layers [9].

In CNNs, it is well-known that checkerboard artifacts are generated by the operations of deconvolution [10, 22, 23] and sub-pixel convolution [11] layers. To overcome these artifacts, smoothness constraints [12], post-processing [13], initialization schemes [14], and different upsampling layer designs [9, 15, 16] have been proposed. Most of them cannot avoid checkerboard artifacts perfectly, although they reduce the artifacts. Among them, Odena *et al.* [9] demonstrated that checkerboard artifacts can be perfectly avoided by using resize convolution layers instead of deconvolution ones. However, resize convolution layers cannot be directly applied to upsampling layers such as deconvolution and sub-pixel convolution ones, so this method requires not only a large amount of memory but it also has high computational costs. In addition, it cannot be applied to the backpropagation of convolutional layers.

Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo, Japan

Corresponding author:
Hitoshi Kiya
Email: kiya@tmu.ac.jp

Checkerboard artifacts have been studied to design linear multirate systems including filter banks and wavelets [17–20]. In addition, it is well-known that the artifacts are caused by the time-variant property of interpolators in multirate systems, and a condition for avoiding these artifacts has have been given [17–19]. However, this condition for linear systems cannot be applied to CNNs due to the non-linearity of CNNs.

In this paper, we extend the conventional avoidance condition for CNNs and apply a proposed structure to typical CNNs to confirm whether the novel structure is effective. Experimental results demonstrate that the proposed structure can perfectly avoid generating checkerboard artifacts caused by both processes, while keeping the excellent properties that CNNs have. As a result, it is confirmed that the proposed structure allows us to offer CNNs without any checkerboard artifacts. This paper is an extension of a conference paper [21].

II. PREPARATION

Checkerboard artifacts in CNNs and work related to checkerboard artifacts are reviewed here.

A) Checkerboard artifacts in CNNs

In CNNs, it is well-known that checkerboard artifacts are caused by two processes: forward-propagation of upsampling layers and backpropagation of convolutional layers. This paper focuses on these two issues in CNNs [9, 14, 21].

When CNNs include upsampling layers, there is a possibility that they will generate checkerboard artifacts, which
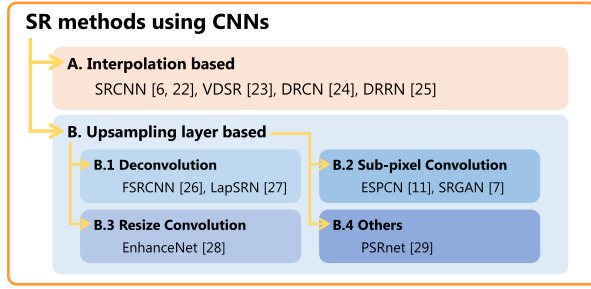
Fig. 1. Classification of SR methods using CNNs. There is a possibility that SR methods will generate checkerboard artifacts, when CNNs include upsampling layers.

Fig. 2. CNNs with an upsampling Layer.

(a) General structure

(b) Polyphase structure

Fig. 3. Linear interpolators with upscaling factor U. (a) General structure, (b) polyphase structure.

is the first issue, referred to as issue A. Deconvolution [22, 23], sub-pixel convolution [11], and resize convolution [9] layers are well-known to include upsampling layers, respectively. Deconvolution layers have many names, including fractionally-strided convolutional layer and transposed convolutional layer, as described in [23]. In this paper, the term "Deconvolution" has the same meaning as others.

Checkerboard artifacts are also generated by the backward pass of convolutional layers, which is the second issue, referred to as issue B. We will mainly consider issue A in the following discussion, since issue B is reduced to issue A under some conditions.

B) SR methods using CNNs

SR methods using CNNs are classified into two classes as shown in Fig. 1. Interpolation-based methods [6, 24–27], referred to as "class A," do not generate any checkerboard artifacts in CNNs, due to the use of an interpolated image as an input to a network. In other words, CNNs in this class do not have any upsampling layers.

When CNNs include upsampling layers, there is a possibility that CNNs will generate checkerboard artifacts. This class, called "class B" in this paper, has provided numerous excellent SR methods [7, 11, 28–33] that can be executed faster than those in class A. Class B is also classified into a number of sub-classes according to the type of upsampling layer. This paper focuses on class B.

CNNs are illustrated in Fig. 2 for an SR problem, as in [11], where the CNNs consist of two convolutional layers and one upsampling layer. I_{LR} and $f_c^{(l)}(I_{LR})$ are a low-resolution (LR) image and c-th channel feature map at layer l, and $f(I_{LR})$ is an output of the network. The two layers have learnable weights, biases, and ReLU as an activation function, where the weight at layer l has $K_l \times K_l$ as a spatial size and N_l as the number of feature maps.

There are numerous algorithms for computing upsampling layers, such as deconvolution [22, 23], sub-pixel convolution [11], and resize convolution [9] ones, which are widely used in typical CNNs. In addition, recently, some excellent SR methods have been proposed [31, 34].

C) Works related to checkerboard artifacts

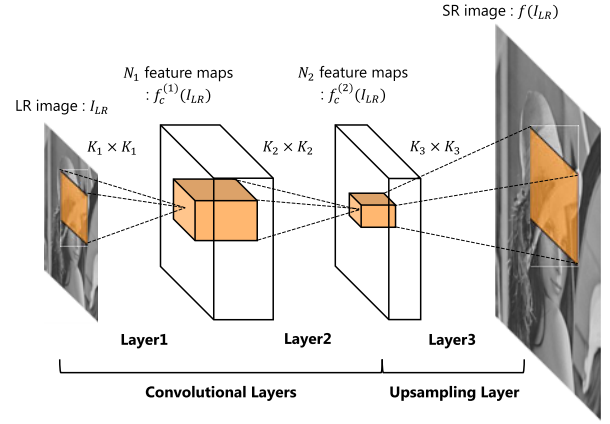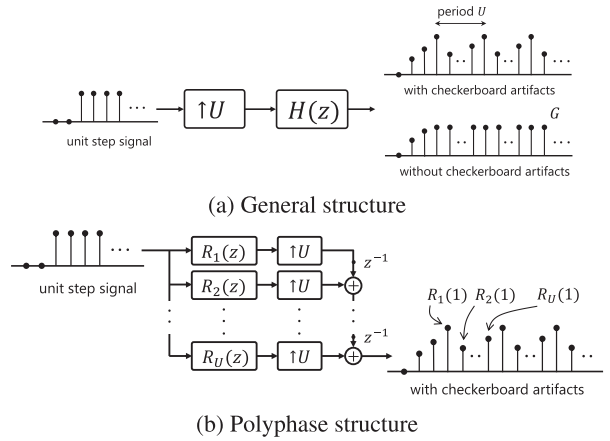Checkerboard artifacts have been discussed by researchers for designing multirate systems including filter banks and wavelets [17–20]. However, most research has been limited to cases of using linear systems, so it cannot be directly applied to CNNs due to the non-linearity of CNNs. Some pieces of works related to checkerboard artifacts for linear systems are summarized, here.

It is known that linear interpolators, which consist of upsamplers and linear time-invariant systems, cause checkerboard artifacts due to their periodic time-variant property [17–19]. Figure 3 illustrates a linear interpolator with an upsampler $\uparrow U$ and a linear time-invariant system $H(z)$, where positive integer U is an upscaling factor and $H(z)$ is the z transformation of an impulse response. The interpolator in Fig. 3(a) can be equivalently represented as a polyphase structure as shown in Fig. 3(b). The relationship between $H(z)$ and $R_i(z)$ is given by

$$H(z) = \sum_{i=1}^{U} R_i(z^U) z^{-(U-i)}, \quad (1)$$

where $R_i(z)$ is often referred to as a polyphase filter of the filter $H(z)$.

The necessary and sufficient condition for avoiding checkerboard artifacts in the system is shown as

$$R_1(1) = R_2(1) = \cdots = R_U(1) = G. \quad (2)$$

Table 1. Correspondence relation of technical terms in signal processing and computer vision

Signal processing	Computer vision
Filter, linear system	Filter, kernel
Impulse response, Filter coefficient	Weight
DC value of filter	Summation of weights
z^{-1}	One pixel shift
Factor U	Stride U (in deconvolution layer)
Zero-order hold	Nearest neighbor

This condition means that all polyphase filters have the same DC value, i.e. a constant G [17–19]. Note that each DC value $R_i(1)$ corresponds to the steady-state value of the unit step response in each polyphase filter $R_i(z)$. In addition, the condition of equation (2) can be also expressed as

$$H(z) = P(z)H_o(z), \tag{3}$$

where,

$$H_o(z) = \sum_{i=0}^{U-1} z^{-i}, \tag{4}$$

and $H_o(z)$ and $P(z)$ are an interpolation kernel of the zero-order hold with factor U and a time-invariant filter, respectively. Therefore, a linear interpolator with factor U does not generate any checkerboard artifacts, when $H(z)$ includes $H_o(z)$. In the case without checkerboard artifacts, the step response of the linear system has a steady-state value G as shown in Fig. 3(a). Meanwhile, the step response of the linear system has a periodic steady-state signal with the period of U, such as $R_1(1), \ldots, R_U(1)$, if equation (3) is not satisfied.

To intermediate between signal processing field and computer vision one, the correspondence relation of some technical terms is summarized in Table 1.

III. PROPOSED METHOD

CNNs are non-linear systems, so conventional work related to checkerboard artifacts cannot be directly applied to CNNs. A condition for avoiding checkerboard artifacts in CNNs is proposed here.

A) CNNs with upsampling layers

We focus on upsampling layers in CNNs, for which there are numerous algorithms such as deconvolution [22, 23], sub-pixel convolution [11], and resize convolution [9]. For simplicity, one-dimensional CNNs will be considered in the following discussion.

It is well-known that deconvolution layers with non-unit strides cause checkerboard artifacts [9]. Figure 4 illustrates a system representation of deconvolution layers [22, 23] that consist of interpolators, where H_c and b are a weight and a bias in which c is a channel index, respectively. The deconvolution layer in Fig. 4(a) can be equivalently represented as a polyphase structure in Fig. 4(b), where $R_{c,n}$ is a polyphase

(a) General structure

(b) Polyphase structure
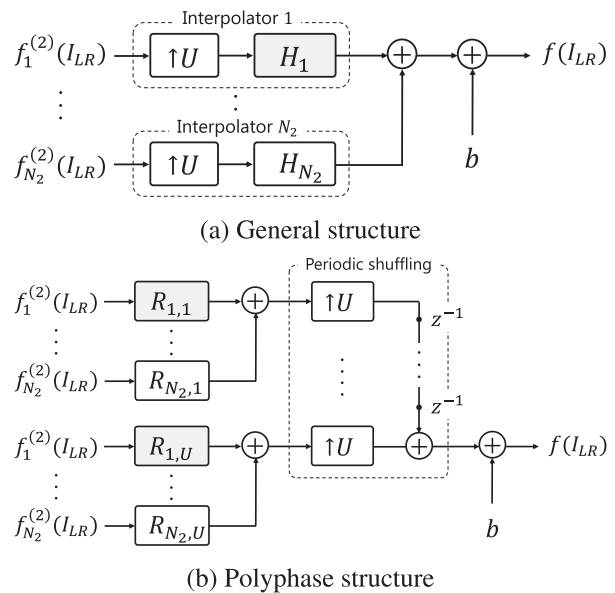
Fig. 4. Deconvolution layer [22, 23]. (a) General structure, (b) Polyphase structure.
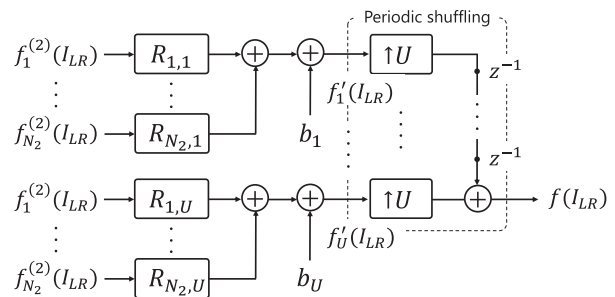
Fig. 5. Sub-pixel convolution layer [11].

filter of the filter H_c in which n is a filter index. This is a non-linear system due to the bias b.

Figure 5 illustrates sub-pixel convolution layers [11], where $R_{c,n}$ and b_n are a weight and a bias, and $f'_n(I_{LR})$ is an intermediate feature map in channel n. Comparing Fig. 4(b) with Fig. 5, we can see that the polyphase structure in Fig. 4(b) is a special case of the sub-pixel convolution layers in Fig. 5. In other words, Fig. 5 is reduced to Fig. 4(b), when satisfying $b_1 = b_2 = \cdots = b_U$. Therefore, we will focus on sub-pixel convolution layers as a general case of upsampling layers to discuss checkerboard artifacts in CNNs.

B) Checkerboard artifacts in upsampling layers

Let us consider the unit step response in CNNs. In Fig. 2, when the input I_{LR} is the unit step signal I_{step}, the steady-state value of the c-th channel feature map in layer 2 is given as

$$\hat{f}_c^{(2)}(I_{step}) = A_c, \tag{5}$$

where A_c is a positive constant value that is decided by filters, biases, and ReLU. Therefore, from Fig. 5, the steady-state value of the n-th channel intermediate feature map is

given by, for sub-pixel convolution layers,

$$\hat{f}'_n(I_{step}) = \sum_{c=1}^{N_2} A_c \overline{R}_{c,n} + b_n, \qquad (6)$$

where $\overline{R}_{c,n}$ is the DC value of the filter $R_{c,n}$.

Generally, the condition, which corresponds to equation (2) for linear multirate systems,

$$\hat{f}'_1(I_{step}) = \hat{f}'_2(I_{step}) = \cdots = \hat{f}'_U(I_{step}) \qquad (7)$$

is not satisfied, so the unit step response $f(I_{step})$ has a periodic steady-state signal with the period of U. To avoid checkerboard artifacts, equation (7) has to be satisfied, as well as for linear multirate systems.

C) Upsampling layers without checkerboard artifacts

To avoid checkerboard artifacts, CNNs must have the non-periodic steady-state value in the unit step response. From equations (6), equation (7) is satisfied if

$$\overline{R}_{c,1} = \overline{R}_{c,2} = \cdots = \overline{R}_{c,U}, \ c = 1, 2, \ldots, N_2 \qquad (8)$$

$$b_1 = b_2 = \cdots = b_U. \qquad (9)$$

Note that, in this case,

$$\hat{f}'_1(K \cdot I_{step}) = \hat{f}'_2(K \cdot I_{step}) = \cdots = \hat{f}'_U(K \cdot I_{step}) \qquad (10)$$

is also satisfied, where K is an arbitrary constant value. However, even when each filter H_c in Fig. 5 satisfies equation (2) or equation (3), equation (9) is not met, but equation (8) is met. Both equations, i.e., equations (8 and 9) have to be met to avoid checkerboard artifacts in CNNs. Therefore, we have to find a novel way of avoiding checkerboard artifacts in CNNs. Note that equations (5) and (7) correspond to values in case that the input I_{LR} is the unit step I_{step}. Therefore, other general inputs, the output feature map would not be the same even when equations (5) and (7) are met.

In this paper, we propose adding the kernel of the zero-order hold with factor U, i.e., H_o in equation (4), after upsampling layers, as shown in Fig. 6. In this structure, the unit-step response outputted from H_o have constant value as the steady state values, even when an arbitrary periodic signal with a period of U is inputted to H_o. As a result, Fig. 6 can satisfy equation (7). In other words, the steady-state values of the step response are not periodic in this case.

The difference between the conventional upsampling layers and the proposed structure is whether the structure has H_o forcibly inserted for avoiding checkerboard artifacts or not. The operation of sub-pixel convolution and deconvolution layers can be interpreted as a combination of upsampling and convolution, where upsampling corresponds to the operation that is to insert $(U-1)$ zeros between sample values. The conventional upsampling layers do not include H_o generally unless forcibly inserting H_o into convolution, so checkerboard artifacts are generated.
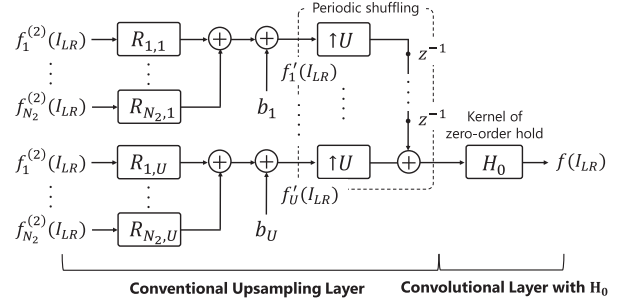

Fig. 6. Proposed upsampling layer structure without checkerboard artifacts. Kernel of zero-order hold with factor U is added after upsampling layers.

There are three approaches to using H_o in CNNs that differ in terms of how CNNs are trained as follows.

1) TRAINING CNNs WITHOUT H_o

The first approach for avoiding checkerboard artifacts, called "approach 1," is to add H_o to CNNs after training the CNNs. This approach allows us to perfectly avoid checkerboard artifacts generated by a pre-trained model.

2) TRAINING CNNs WITH H_o

In approach 2, H_o is added as a convolution layer after the upsampling layer shown in Fig. 6, and then, the CNNs with H_o are trained. This approach also allows us to perfectly avoid checkerboard artifacts as well as approach 1. Moreover, this approach generally provides higher quality images than those of approach 1.

3) TRAINING CNNs WITH H_o INSIDE UPSAMPLING LAYERS

Approach 3 is applicable only to deconvolution layers, but approaches 1 and 2 can be used for both deconvolution and sub-pixel convolution layers. Deconvolution layers always satisfy equation (9), so equation (8) only has to be considered. Therefore, CNNs do not generate any checkerboard artifacts when each filter H_c in Fig. 5 satisfies equation (3). In approach 3, checkerboard artifacts are avoided by convolving each filter H_c with the kernel H_o inside upsampling layers.

D) Checkerboard artifacts in gradients

It is well-known that checkerboard artifacts are also generated in gradients of convolutional layers [9] since the operations of deconvolution layers are carried out on the backward pass to compute the gradients. Therefore, both approaches 2 and 3 can avoid checkerboard artifacts as well as for deconvolution layers. Note that, for approach 2, we have to add the kernel of the zero-order hold before convolutional layers to avoid checkerboard artifacts on the backward pass.

It is also well-known that max-pooling layers cause high-frequency artifacts in gradients [9]. However, these artifacts are generally different from checkerboard artifacts, so this paper does not consider these high-frequency artifacts.

Table 2. CNNs used for super-resolution tasks

Network name	Upsampling layer	$K_3 \times K_3$
Deconv	Deconvolution [22, 23]	9×9
Sub-pixel	Sub-pixel convolution [11]	3×3
ResizeConv	Resize convolution [9]	9×9
Deconv + H_o	Deconvolution with H_o (Approach 1 or 2)	9×9
Deconv + H_o (Ap. 3)	Deconvolution with H_o (Approach 3)	9×9
Sub-pixel + H_o	Sub-pixel convolution with H_o (Approach 1 or 2)	3×3

IV. EXPERIMENTS AND RESULTS

The proposed structure without checkerboard artifacts was applied to typical CNNs to demonstrate its effectiveness. In the experiments, two tasks, SR imaging and image classification, were carried out.

A) Super-resolution

The proposed structure without checkerboard artifacts was applied to the SR methods using deconvolution and sub-pixel convolution layers. The experiments with CNNs were carried out under two loss functions: mean squared error (MSE) and perceptual loss.

1) DATASETS FOR TRAINING AND TESTING
We employed 91-image set from Yang *et al.* [35] as our training dataset. In addition, the same data augmentation (rotation and downscaling) as in [28] was used. As a result, a training dataset consisting of 1820 images was created for our experiments. In addition, we used two datasets, Set5 [36] and Set14 [37], which are often used for benchmarking, as test datasets.

To prepare a training set, we first downscaled ground truth images I_{HR} with a bicubic kernel to create LR images I_{LR}, where the factor $U = 4$ was used. The ground truth images I_{HR} were cropped into 72×72 pixel patches, and the LR images were also cropped 18×18 pixel ones, where the total number of extracted patches was 8,000. In the experiments, the luminance channel (Y) of images was used for

MSE loss, and three channels (RGB) of images were used for perceptual loss.

2) TRAINING DETAILS
Table 2 illustrates the CNNs used in the experiments, which were carried out on the basis of the CNNs in Fig. 2. For the other two layers in Fig. 2, we set $(K_1, N_1) = (5, 64)$, $(K_2, N_2) = (3, 32)$ as in [11]. In addition, all networks were trained to minimize the MSE $1/2\|I_{HR} - f(I_{LR})\|^2$ and the perceptual loss $1/2\|\phi(I_{HR}) - \phi(f(I_{LR}))\|^2$ averaged over the training set, where ϕ calculates feature maps at the fourth layer of a pre-trained VGG-16 model as in [13]. Note that Deconv + H_o, Deconv + H_o (Ap. 3), and Sub-pixel + H_o in Table 2 use the proposed structure.

For training, Adam [38] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ was employed as an optimizer. In addition, we set the batch size to 4 and the learning rate to 0.0001. The weights were initialized with the method described in He *et al.* [39]. We trained all models for 200 K iterations. All models were implemented by using the TensorFlow framework [40].

3) EXPERIMENTAL RESULTS
Figure 7 shows examples of SR images generated under perceptual loss, where mean PSNR values for each dataset are also illustrated. In this figure, (b) and (f) include checkerboard artifacts, and (c)–(e), (g)–(i) do not. Moreover, it is shown that the quality of SR images was significantly improved by avoiding the artifacts. Approaches 2 and 3 also provided better quality images than approach 1. Note that ResizeConv did not generate any artifacts, because it uses a pre-defined interpolation like in [6]. In Fig. 8, the usefulness of the proposed avoidance condition is demonstrated. It was confirmed that CNNs generated checkerboard artifacts under the conventional condition unless CNNs satisfied the proposed condition. Figure 9 shows other examples of SR images. In Fig. 9, the trend is almost the same as that in Fig. 7.

Table 3 illustrates the average execution time when each CNNs were run 10 times for some images in Set14. Resize-Conv has the highest computational cost in this table, although it did not generate any checkerboard artifacts. From this table, the proposed approaches have much lower computational costs than with resize convolution layers.

(a) Ground Truth Image
(Set5 mean / Set14 mean)

(b) Deconv
(22.06 / 21.58)

(c) Deconv+H_0 (Ap. 1)
(proposed, 28.30 / 26.26)

(d) Deconv+H_0 (Ap. 2)
(proposed, 28.99 / 26.81)

(e) Deconv+H_0 (Ap. 3)
(proposed, 29.33 / 27.06)

(f) Sub-pixel
(22.10 / 21.57)

(g) Sub-pixel+H_0 (Ap. 1)
(proposed, 28.39 / 26.29)

(h) Sub-pixel+H_0 (Ap. 2)
(proposed, 29.03 / 26.84)

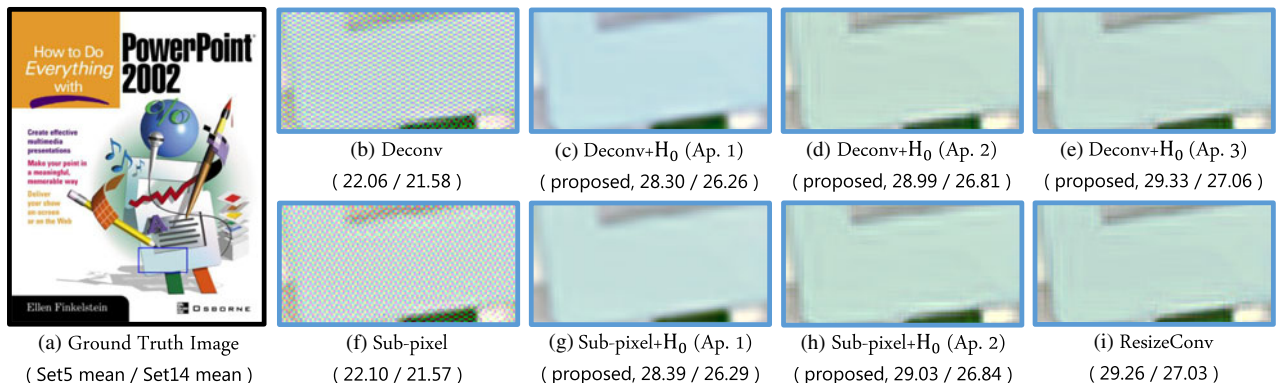(i) ResizeConv
(29.26 / 27.03)

Fig. 7. Experimental results of super-resolution imaging under perceptual loss [PSNR(dB)]. (b) and (f) include checkerboard artifacts, and (c), (d), (e), (g), (h), and (i) do not.
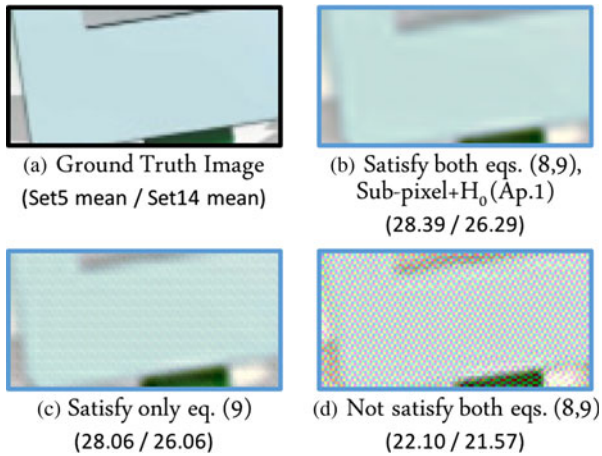
(a) Ground Truth Image
(Set5 mean / Set14 mean)

(b) Satisfy both eqs. (8,9),
Sub-pixel+H_0(Ap.1)
(28.39 / 26.29)

(c) Satisfy only eq. (9)
(28.06 / 26.06)

(d) Not satisfy both eqs. (8,9)
(22.10 / 21.57)

Fig. 8. Super-resolution imaging using perceptual loss under various avoidance conditions [PSNR(dB)] (sub-pixel convolution).

Table 3. Execution time of super-resolution (sec)

Resolution of input image	Deconv	Deconv + H_0 (Ap. 1 or 2)	Deconv + H_0 (Ap. 3)
69×69	0.00871	0.0115	0.0100
125×90	0.0185	0.0270	0.0227
128×128	0.0244	0.0348	0.0295
132×164	0.0291	0.0393	0.0377
180×144	0.0343	0.0476	0.0421

Resolution of input image	Sub-pixel	Sub-pixel + H_0 (Ap. 1 or 2)	ResizeConv
69×69	0.0159	0.0242	0.107
125×90	0.0398	0.0558	0.224
128×128	0.0437	0.0619	0.299
132×164	0.0696	0.0806	0.383
180×144	0.0647	0.102	0.450

Note that the results were obtained on a PC with a 3.30-GHz CPU and a main memory size of 16 GB.

4) LOSS FUNCTIONS

It is well-known that perceptual loss results in sharper SR images despite lower PSNR values [13, 30], and it generates checkerboard artifacts more frequently than under MSE loss as described in [9, 13, 14, 41]. In Fig. 10, which demonstrates artifacts under MSE loss, (b) and (f) also include checkerboard artifacts as well as in Fig. 7, although the distortion is not that large, compared with under perceptual loss. There is a possibility that any loss function causes
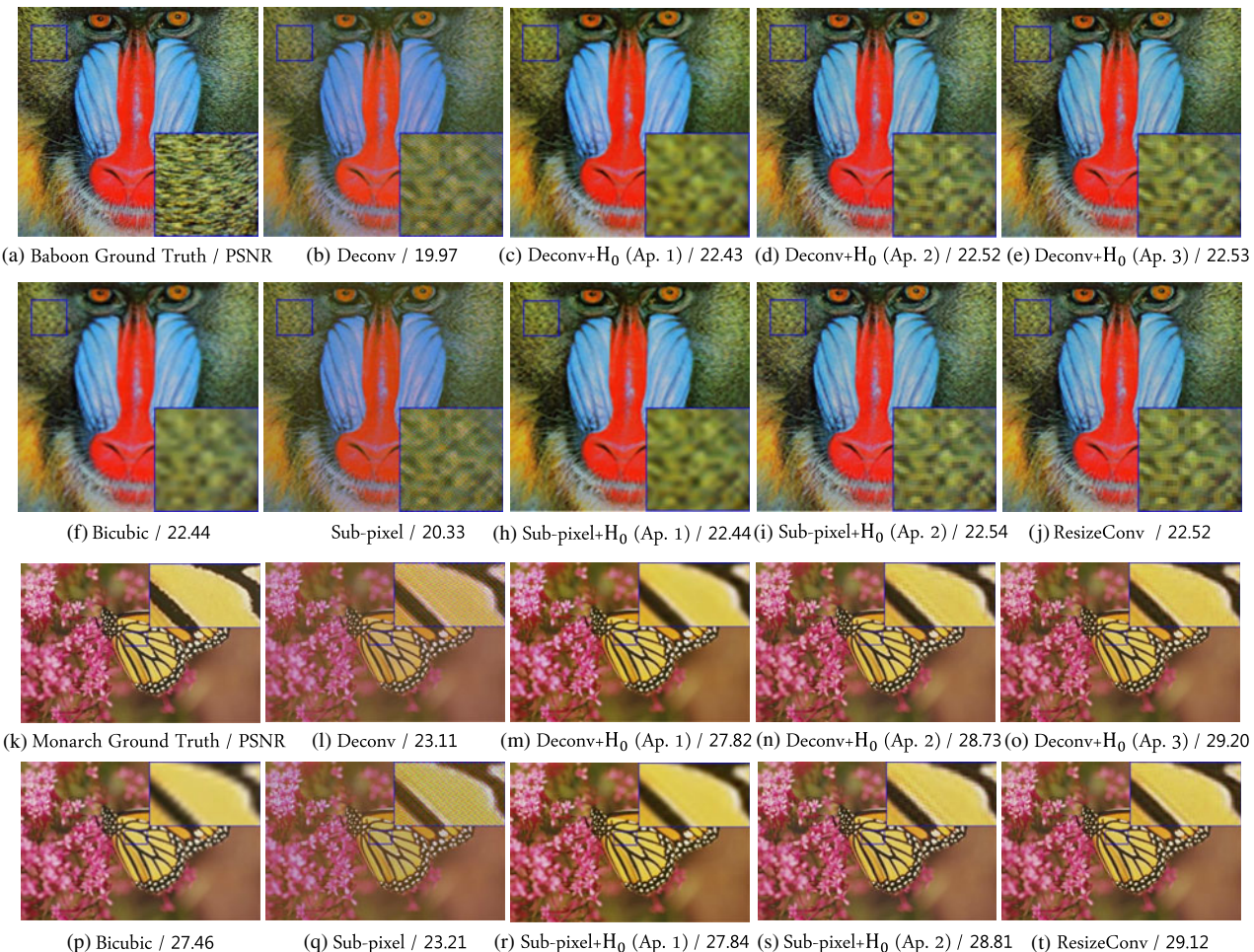

(a) Baboon Ground Truth / PSNR
(b) Deconv / 19.97
(c) Deconv+H_0 (Ap. 1) / 22.43
(d) Deconv+H_0 (Ap. 2) / 22.52
(e) Deconv+H_0 (Ap. 3) / 22.53

(f) Bicubic / 22.44
(Sub-pixel / 20.33)
(h) Sub-pixel+H_0 (Ap. 1) / 22.44
(i) Sub-pixel+H_0 (Ap. 2) / 22.54
(j) ResizeConv / 22.52

(k) Monarch Ground Truth / PSNR
(l) Deconv / 23.11
(m) Deconv+H_0 (Ap. 1) / 27.82
(n) Deconv+H_0 (Ap. 2) / 28.73
(o) Deconv+H_0 (Ap. 3) / 29.20

(p) Bicubic / 27.46
(q) Sub-pixel / 23.21
(r) Sub-pixel+H_0 (Ap. 1) / 27.84
(s) Sub-pixel+H_0 (Ap. 2) / 28.81
(t) ResizeConv / 29.12

Fig. 9. Super-resolution examples of "Baboon" and "Monarch" under perceptual loss. PSNR values are illustrated under each sub-figure. (b), (g), (l), and (q) include checkerboard artifacts, and other examples do not.
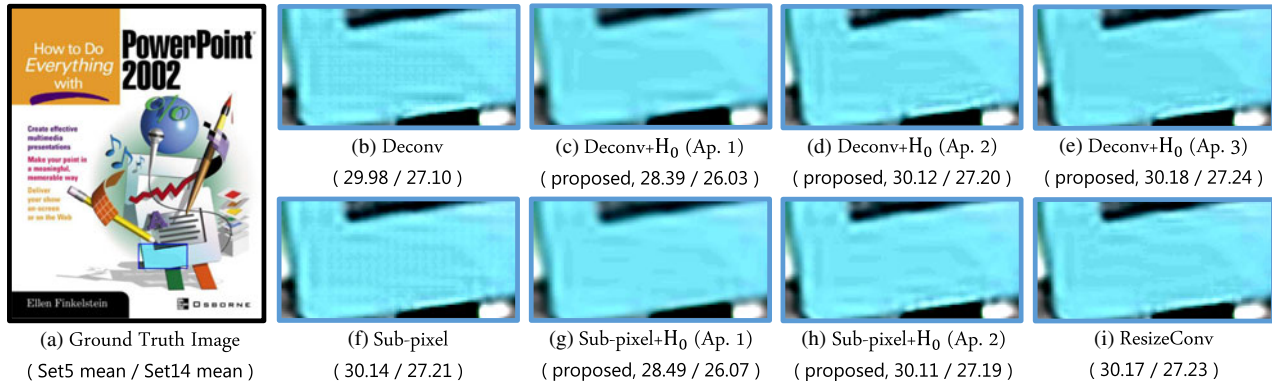
Fig. 10. Experimental results of super-resolution under MSE loss [PSNR(dB)]. (b) and (f) also include checkerboard artifacts as well as in Fig. 7, although the distortion was not that large, compared with under perceptual loss.

Table 4. CNNs used for image classification tasks

Network name	Downsampling layer	Stride
StridedConv	Convolution	2
StridedConv + H_o (Ap. 2)	Convolution with H_o (Approach 2)	2
StridedConv + H_o (Ap. 3)	Convolution with H_o (Approach 3)	2

checkerboard artifacts, but the magnitude of checkerboard artifacts depends on a class of loss functions used for training networks. The proposed avoidance condition is useful under any loss function.

B) Image classification

Next, the proposed structure without checkerboard artifacts was applied to CNNs-based image classification models.

1) DATASETS FOR TRAINING AND TESTING
We employed two datasets, CIFAR10 and CIFAR100, which contain 32×32 pixel color images and consist of 50 000 training images and 10 000 test images [42]. In addition, standard data augmentation (mirroring and shifting) was

used. For preprocessing, the images were normalized by using the channel means and standard deviations.

2) TRAINING DETAILS
Table 4 illustrates the CNNs used in the experiments, which were run on the basis of ResNet-110 [2]. Note that the projection shortcut [2] was used only for increasing the number of dimensions, and all convolutional layers with a stride of 2 in ResNet-110 were replaced by downsampling layers in Table 4.

All of the networks were trained by using stochastic gradient descent with momentum for 300 epochs. The learning rate was initially set to 0.1, and decreased by a factor of 10 at 150 and 225 epochs. The weights were initialized by the method introduced in [39]. We used a weight decay of 0.0001, a momentum of 0.9, and a batch size of 64.

3) EXPERIMENTAL RESULTS
Figure 11 shows examples of gradients, which were computed on the backward pass of the first downsampling layer, for each CNN. In this figure, (a) includes checkerboard artifacts, and (b) and (c) do not.

The results for CIFAR10 and CIFAR100 are illustrated in Table 5, where " + " indicates the use of the standard data augmentation. It is shown that approach 3 performed the
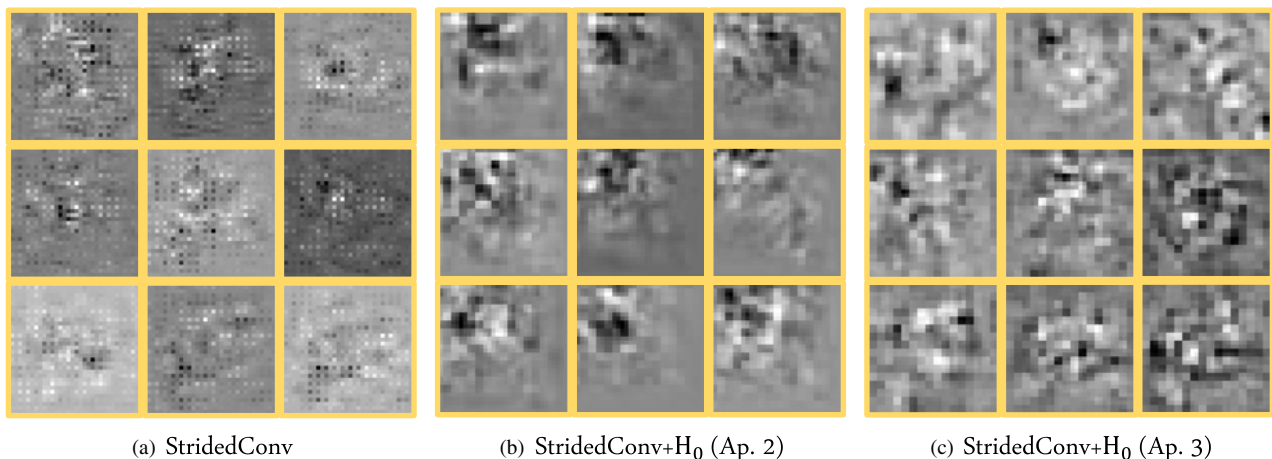

Fig. 11. Gradients computed in first downsampling layer. (a) includes checkerboard artifacts, and (b) and (c) do not.

Table 5. Error rates on CIFAR10, CIFAR100 datasets (%)

Network	CIFAR10	CIFAR10+	CIFAR100+
StridedConv	12.75	6.13	32.72
StridedConv + H_o (Ap. 2)	16.44	10.08	34.91
StridedConv + H_o (Ap. 3)	11.21	5.85	29.34

best in this table. This trend was almost the same for the SR tasks.

V. CONCLUSION

We addressed a condition for avoiding checkerboard artifacts in CNNs. A novel structure without any checkerboard artifacts was proposed by extending the conventional condition for linear systems to CNNs with non-linearity. The experimental results demonstrated that the proposed structure can perfectly avoid generating checkerboard artifacts caused by both of two processes, forward-propagation of upsampling layers and backpropagation of convolutional layers, while maintaining the excellent properties that CNNs have. As a result, the proposed structure allows us to offer CNNs without any checkerboard artifacts.

REFERENCES

[1] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks, in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012, 1097–1105.

[2] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep Residual Learning for Image Recognition, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, 770–778.

[3] Huang, G.; Liu, Z.; Maaten, L.; van der, Weinberger, K.Q.: Densely Connected Convolutional Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, 2261–2269.

[4] Noh, H.; Hong, S.; Han, B.: Learning Deconvolution Network for Semantic Segmentation, in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, 1520–1528.

[5] Shelhamer, E.; Long, J.; Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **39**(4) (2017), 640–651.

[6] Dong, C.; Loy, C.C.; He, K.; Tang, X.: Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, **38**(2) (2016), 295–307.

[7] Ledig, C., *et al.*: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, 105–114.

[8] Radford, A.; Metz, L.; Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, in *arXiv preprint arXiv:1511.06434*, 2015.

[9] Odena, A.; Dumoulin, V.; Olah, C.: Deconvolution and Checkerboard Artifacts. *Distill*, (2016). [Online]. Available: http://distill.pub/2016/deconv-checkerboard.

[10] Zeiler, M.D.; Taylor, G.W.; Fergus, R.: Adaptive Deconvolutional Networks for Mid and High Level Feature Learning, in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, 2018–2025.

[11] Shi, W., *et al.*: Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, in *Proc.*

[12] Dosovitskiy, A., *et al.*: FlowNet: Learning Optical Flow with Convolutional Networks, in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, 2758–2766.

[13] Johnson, J.; Alahi, A.; Li, F.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution, in *Proc. European Conf. on Computer Vision (ECCV)*, Springer. 2016, 694–711.

[14] Aitken, A.P.; Ledig, C.; Theis, L.; Caballero, J.; Wang, Z.; Shi, W.: Checkerboard Artifact Free Sub-Pixel Convolution: A Note on Sub-Pixel Convolution, Resize Convolution and Convolution Resize, in *arXiv preprint arXiv:1707.02937*, 2017.

[15] Gao, H.; Yuan, H.; Wang, Z.; Ji, S.: Pixel Deconvolutional Networks, in *arXiv preprint arXiv:1705.06820*, 2017.

[16] Wojna, Z., *et al.*: The Devil is in the Decoder, in *Proc. British Machine Vision Conf. (BMVC)*, 2017.

[17] Harada, Y.; Muramatsu, S.; Kiya, H.: Multidimensional Multirate Filter without Checkerboard Effects, in *Proc. European Signal Processing Conf. (EUSIPCO)*. 1998, 1881–1884.

[18] Tamura, T.; Kato, M.; Yoshida, T.; Nishihara, A.: Design of Checkerboard-Distortion-Free Multidimensional Multirate Filters. *IEICE Trans. Fundamentals Electron. Commun. Computer Sci.*, **E81-A**(8) (1998), 1598–1606.

[19] Harada, Y.; Muramatu, S.; Kiya, H.: Multidimensional Multirate Filter and Filter Bank without Checkerboard Effect. *IEICE Trans. Fundamentals Electron. Commun. Computer Sci.*, **E81-A**(8) (1998), 1607–1615.

[20] Iwai, H.; Iwahashi, M.; Kiya, H.: Methods for Avoiding the Checkerboard Distortion Caused by Finite Word Length Error in Multirate System. *IEICE Trans. Fundamentals Electron. Commun. Computer Sci.*, **E93-A**(3) (2010), 631–635.

[21] Sugawara, Y.; Shiota, S.; Kiya, H.: Super-Resolution using Convolutional Neural Networks without Any Checkerboard Artifacts, in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*. 2018, 66–70.

[22] Vincent, D.; Francesco, V.: A guide to convolution arithmetic for deep learning, in *arXiv preprint arXiv:1603.07285*, 2016.

[23] Shi, W.; Caballero, J.; Theis, L.; Huszar, F.; Aitken, A.; Ledig, C.: Is the deconvolution layer the same as a convolutional layer?, Wangin, Z.: *arXiv:1609.07009*, 2016.

[24] Dong, C.; Loy, C.C.; He, K.; Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution, in *Proc. European Conf. on Computer Vision (ECCV)*, Springer, 2014, 184–199.

[25] Kim, J.; Lee, J.K.; Lee, K.M.: Accurate Image Super-Resolution Using Very Deep Convolutional Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016, 1646–1654.

[26] Kim, J.; Lee, J.K.; Lee, K.M.: Deeply-Recursive Convolutional Network for Image Super-Resolution, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016, 1637–1645.

[27] Tai, Y.; Yang, J.; Liu, X.: Image Super-Resolution via Deep Recursive Residual Network, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, 2790–2798.

[28] Dong, C.; Loy, C.C.; Tang, X.: Accelerating the Super-Resolution Convolutional Neural Network, in *Proc. European Conf. on Computer Vision (ECCV)*, Springer, 2016, 391–407.

[29] Lai, W.; Huang, J.; Ahuja, N.; Yang, M.: Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, 5835–5843.

[30] Sajjadi, M.S.M.; Schölkopf, B.; Hirsch, M.: EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis, in

IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2016, 1874–1883.

Proc. IEEE Int. Conf. on Computer Vision (ICCV), 2017, 4491–4500.

[31] Zhang, K.; Zuo, W.; Zhang, L.: Learning a Single Convolutional Super-Resolution Network for Multiple Degradations, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B; Fu, Y.: Learning a Deep Convolutional Network for Image Super-Resolution, in *Proc. European Conf. on Computer Vision (ECCV)*, Springer, 2018.

[33] Sugawara, Y.; Shiota, S.; Kiya, H.: A Parallel Computation Algorithm for Super-Resolution Methods Using Convolutional Neural Networks, in *Proc. Asia Pacific Signal and Information Processing Association (APSIPA) Annual Summit and Conf.*, 2017, 1169–1173.

[34] Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y.: Image Super-Resolution Using Very Deep ResidualChannel Attention Networks, in *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

[35] Yang, J.; Wright, J.; Huang, T.S.; Ma, Y.: Image Super-Resolution via Sparse Representation. *IEEE Trans. Image. Process.*, **19**(11) (2010), 2861–2873.

[36] Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L.: Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding, in *Proc. British Machine Vision Conf. (BMVC)*, 2012.

[37] Zeyde, R.; Elad, M.; Protter, M.: On Single Image Scale-Up Using Sparse-Representations, in *Proc. Curves and Surfaces*, 2010, 711–730.

[38] Kingma, D.P.; Ba, J.: Adam: A Method for Stochastic Optimization, in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.

[39] He, K.; Zhang, X.; Ren, S.; Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, 1026–1034.

[40] Abadi, M.; Agarwal, A.; Barham, P., et al. : TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/.

[41] Blau, Y.; Michaeli, T.: The Perception-Distortion Tradeoff, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, 6228–6237.

[42] Krizhevsky, A.; Hinton, G.E.: Learning Multiple Layers of Features from Tiny Images, *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Yusuke Sugawara received his B.Eng degree from Takushoku University, Japan in 2016, and his M.Eng degree from Tokyo Metropolitan University, Japan in 2018. He graduated from Tokyo Metropolitan University, Japan in 2018. His research interests include image processing.

Sayaka Shiota received B.E., M.E., and Ph.D. degrees in intelligence and computer science, engineering, and engineering simulation from the Nagoya Institute of Technology, Nagoya, Japan in 2007, 2009, and 2012, respectively. From February 2013 to March 2014, she worked as a Project Assistant Professor at the Institute of Statistics Mathematics. In April of 2014, she joined Tokyo Metropolitan University as an Assistant Professor. Her research interests include statistical speech recognition and speaker verification. She is a member of the Acoustical Society of Japan (ASJ), the IEICE, the ISCA, APSIPA, and the IEEE.

Hitoshi Kiya Hitoshi Kiya received his B.E and M.E. degrees from Nagaoka University of Technology in 1980 and 1982, respectively, and his Dr. Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University, where he became Full Professor in 2000. He is a Fellow of IEEE, IEICE, and ITE. He currently serves as President of APSIPA, and he served as Inaugural Vice President (Technical Activities) of APSIPA from 2009 to 2013 and Regional Director-at-Large for Region 10 of the IEEE Signal Processing Society from 2016 to 2017. He was Editorial Board Member of eight journals, including IEEE Trans. on Signal Processing, Image Processing, and Information Forensics and Security, Chair of two technical committees and Member of nine technical committees including APSIPA Image, Video, and Multimedia TC, and IEEE Information Forensics and Security TC. He has received numerous awards, including six best paper awards.