Multi-Channel Convolutional Neural Networks for Image Super-Resolution

Ohtani, Shinya
Kato, Yu
Kuroki, Nobutaka
Hirose, Tetsuya
Numa, Masahiro

(Citation)
IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,E100A(2):572-580

(Issue Date)
2017-02

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90006266

IEICE TRANS. FUNDAMENTALS, VOL.E100–A, NO.2 FEBRUARY 2017

PAPER *Special Section on Image Media Quality*

Multi-Channel Convolutional Neural Networks for Image Super-Resolution

Shinya OHTANI[†], ***Nonmember***, **Yu KATO**[†], ***Student Member***, **Nobutaka KUROKI**[†a)], **Tetsuya HIROSE**[†], ***and*** **Masahiro NUMA**[†], ***Members***

SUMMARY This paper proposes image super-resolution techniques with multi-channel convolutional neural networks. In the proposed method, output pixels are classified into $K \times K$ groups depending on their coordinates. Those groups are generated from separate channels of a convolutional neural network (CNN). Finally, they are synthesized into a $K \times K$ magnified image. This architecture can enlarge images directly without bicubic interpolation. Experimental results of 2×2, 3×3, and 4×4 magnifications have shown that the average PSNR for the proposed method is about 0.2 dB higher than that for the conventional SRCNN.

key words: *super-resolution, resolution enhancement, convolutional neural networks, CNN, deep learning*

1. Introduction

Image upscaling techniques are important for a number of applications, such as digital cameras, smart phones, and televisions. Freeman [1] proposed an example-based super-resolution that generates a high resolution image from a single low resolution one with the help of a database learned from many training images. Although it previously required a huge database and a fast search algorithm, many improved versions have overcome those problems. The sparse cording (SC)-based method [2]–[4] is one of the example-based methods. This method generates a high resolution patch with a linear combination of a few atoms in a dictionary. The neighbor embedding (NE) method [5], [6] estimates high resolution patches based on locally similar geometry. Anchored Neighborhood Regression (ANR) [7] is sophisticated by a combination of NE and SC. This method improves the performance by precomputing a neighboring anchor in the high resolution space. A+ [8] is an improved version of ANR which increases the anchoring points. Naive Bayes Super-Resolution Forest (NBSRF) [9] improves both speed and quality by using a novel decision-tree approach to select the best patch in a dictionary.

Recently, a neural network-based super-resolution (SRCNN) [10], [11] was proposed, and it improved image qualities dramatically. In this method, the convolutional neural network (CNN) generates a high frequency image from a low frequency one which was previously enlarged by bicubic interpolation. It enables an end-to-end training between low and high frequency images by using the back propagation technique. Moreover, Gu [12] improved image qualities by using a convolutional sparse cording (CSC) in CNN. Currently, CSC achieves the highest PSNR among the above super-resolution methods.

However, the problem of the conventional SRCNN and CSC is that they require bicubic interpolation for pre-processing. The reason why bicubic method must be used to prepare the low frequency image is that the following CNN does not have an upsampling operation. Since the bicubic filters are outside of the training process of the CNN, they cannot be optimized. We think it will become a bottleneck for improving image qualities.

This paper proposes new CNN architectures for image super-resolution without the bicubic method. Our architecture enables a perfect end-to-end training between small and large images. All filter coefficients are optimized in the back propagation process.

The remainder of this paper is organized as follows: Section 2 explains the conventional SRCNN [10], [11]. Next, Sect. 3 presents three types of CNN architectures. Finally, Sect. 4 compares the performances of several super-resolution techniques.

2. Conventional Super-Resolution with CNN

In this section, we give an overview and explain the problems of conventional super-resolution with CNN (SRCNN) [11].

2.1 Overview

Figure 1 shows an overview of the conventional SRCNNs. It requires the bicubic method for pre-processing. After magnifying an input image, CNN generates a high frequency image through three layers. The first and second layers, $F_1(\boldsymbol{Y})$ and $F_2(\boldsymbol{Y})$, have multi-outputs which are defined as feature maps. Each feature map is the filtering result for the previous layer. The last layer, $F(\boldsymbol{Y})$, has a single-channel image. The details of the calculation are as follows:

Let $\boldsymbol{Y}$ be the low frequency image which is magnified by the bicubic method. The first layer is calculated as

$$F_1(\boldsymbol{Y}) = \max(0, W_1 * \boldsymbol{Y} + B_1), \quad (1)$$

where W_1 is the filter for the convolution and B_1 is the bias. The size of W_1 is $1 \times f_1 \times f_1 \times n_1$, where f_1 is the filter size and n_1 is the number of feature maps. The second layer is calculated as

Manuscript received May 24, 2016.
Manuscript revised September 2, 2016.

[†]The authors are with the Graduate School of Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
a) E-mail: kuroki@kobe-u.ac.jp

DOI: 10.1587/transfun.E100.A.572

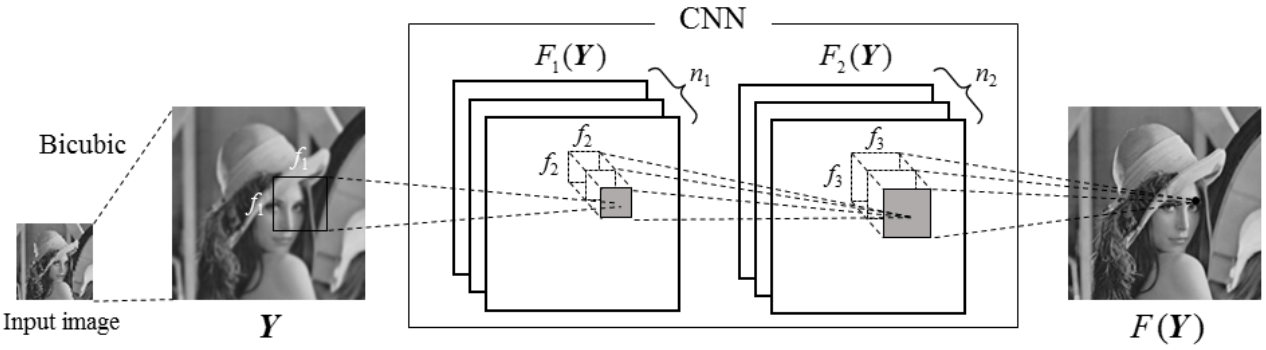


Fig. 1 Overview of conventional SRCNN.

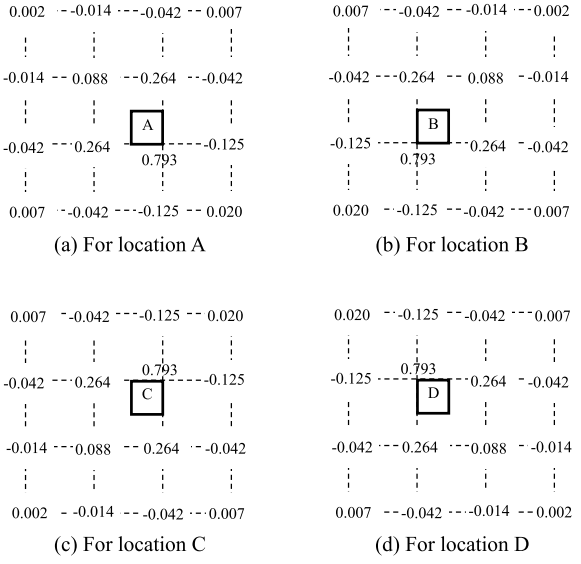


Fig. 2 16 coefficients of bicubic filter.

$$F_2(\boldsymbol{Y}) = \max(0, W_2 * F_1(\boldsymbol{Y}) + B_2)\,. \tag{2}$$

The size of W_2 is $n_1 \times f_2 \times f_2 \times n_2$, where n_2 is the number of feature maps in the second layer. Finally, the super-resolution image of the third layer is calculated as

$$F(\boldsymbol{Y}) = W_3 * F_2(\boldsymbol{Y}) + B_3. \tag{3}$$

The size of W_3 is $n_2 \times f_3 \times f_3 \times 1$.

In the training process, parameters W_1, W_2, W_3, B_1, B_2 and B_3 are determined by the back propagation technique.

2.2 Problems

The conventional SRCNN requires the bicubic method for pre-processing. This causes two problems. The first is that 16 coefficients of the bicubic filter cannot be optimized because they are outside of the training process of the CNN. Figure 2 shows the location of four output pixels in 2×2 magnification and 16 filter coefficients for each location. One output pixel is calculated by applying a 16-tap filter to 16 reference pixels. There are four types of output pixel locations: (A) upper left, (B) upper right, (C) lower-left and (D) lower-right of the nearest input pixel. Different filter coefficients are prepared for (A), (B), (C) and (D). This means that output pixels in odd columns, even columns, odd rows and even rows are obtained by different calculations. As a result, unnatural periodical patterns often appear on the enlarged images, as shown in Fig. 3. We think this degrades the performance of the SRCNN.

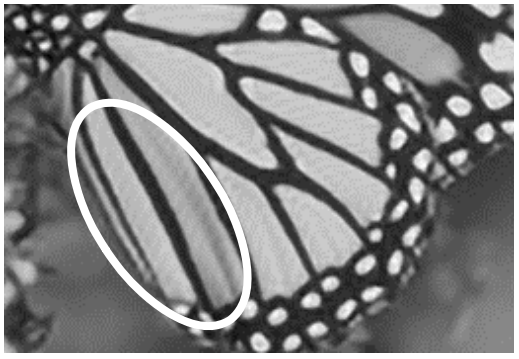


Fig. 3 Periodical degradation derived from the bicubic filter.

The other problem is the increase of memory and calculation costs in the CNN. Since the output image of the bicubic method is four times larger than the input image, CNN also prepare four times larger feature maps in all layers. This wastes a lot of memory. Moreover, the learning speed becomes slow because the filter sizes of f_1, f_2 and f_3 also become large.

3. Proposed Method

To obtain a $K \times K$ magnified image without the bicubic method, we need a new architecture which can generate $K \times K$ pixels per one input pixel. In this section, three types of CNN architectures are proposed.

3.1 Type A: Parallel CNNs

As mentioned in Sect. 2.2, output pixels of a 2×2 magnified image are classified into four groups, (A), (B), (C), and

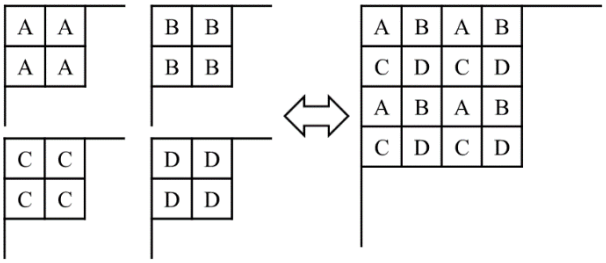


Fig. 4 Separation and synthesis of pixels.

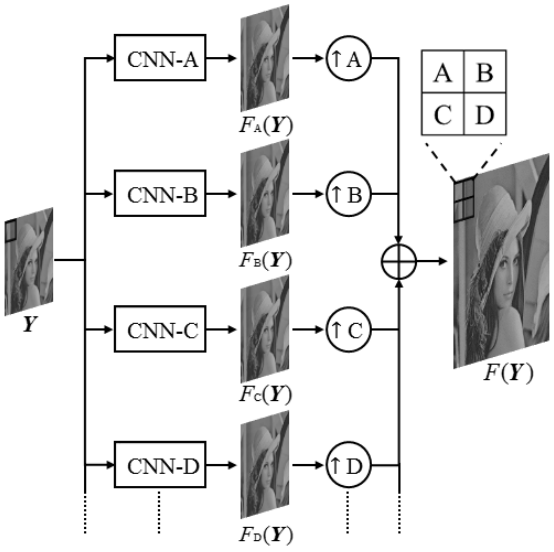


Fig. 5 Type A: Parallel CNNs.

(D), depending on their locations. In Fig. 2, we can find that the groups are calculated with four different filters of fixed coefficients. This means that a magnified image can be obtained with four individual convolution processes, and the results can be rearranged, as is shown in Fig. 4. Thus, we have already proposed an architecture with four parallel-CNNs [13], as shown in Fig. 5. We think each CNN should be optimized for each group. More details are shown in [13]. In this paper, we extend this architecture into $K \times K$ magnification.

In the Type A architecture, $K \times K$ images are generated from $K \times K$ individual CNNs. For example, CNN-A generates only an image $F_A(\boldsymbol{Y})$ consisting of the upper left pixel group (A). The image $F_A(\boldsymbol{Y})$ is calculated as

$$F_{\mathrm{A1}}(\boldsymbol{Y}) = \max\left(0, W_{\mathrm{A1}} * \boldsymbol{Y} + B_{\mathrm{A1}}\right), \tag{4}$$

$$F_{\mathrm{A2}}(\boldsymbol{Y}) = \max\left(0, W_{\mathrm{A2}} * F_{\mathrm{A1}}(\boldsymbol{Y}) + B_{\mathrm{A2}}\right), \tag{5}$$

$$F_{\mathrm{A}}(\boldsymbol{Y}) = W_{\mathrm{A3}} * F_{\mathrm{A2}}(\boldsymbol{Y}) + B_{\mathrm{A3}}. \tag{6}$$

Note that $\boldsymbol{Y}$ and $F_{\mathrm{A}}(\boldsymbol{Y})$ are the same size. The other images $F_{\mathrm{B}}(\boldsymbol{Y})$, $F_{\mathrm{C}}(\boldsymbol{Y})$, $F_D(\boldsymbol{Y})$ ... are also calculated in the same way. Finally, $K \times K$ images are synthesized into a $K \times K$ magnified image $F(\boldsymbol{Y})$ as

$$F(\boldsymbol{Y}) = \uparrow \mathrm{A}(F_{\mathrm{A}}(\boldsymbol{Y})) + \uparrow \mathrm{B}(F_{\mathrm{B}}(\boldsymbol{Y})) + \uparrow \mathrm{C}(F_{\mathrm{C}}(\boldsymbol{Y})) + \uparrow \mathrm{D}(F_{\mathrm{D}}(\boldsymbol{Y})) + \ldots, \tag{7}$$

where ↑A( ), ↑B( ), ↑C( ), ↑D( ) ... mean up-sampling to their locations on the magnified image. Figure 4 shows the example of a 2×2 synthesis.

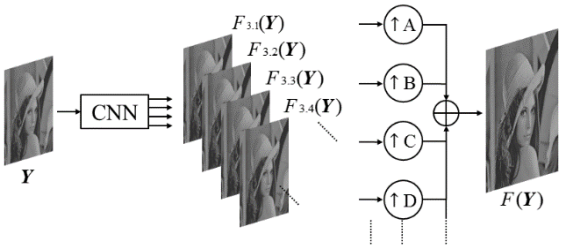


Fig. 6 Type B: Multi-channel CNN.

3.2 Type B: Multi-Channel CNN

Next, we propose a single CNN architecture with multiple output-channels, as shown in Fig. 6. While Type A architecture obtains pixel groups (A), (B), (C), (D) ... from individual CNNs, Type B architecture obtains them from separate channels of a single CNN. They are calculated as

$$F_1(\boldsymbol{Y}) = \max\left(0, W_1 * \boldsymbol{Y} + B_1\right), \tag{8}$$

$$F_2(\boldsymbol{Y}) = \max\left(0, W_2 * F_1(\boldsymbol{Y}) + B_2\right), \tag{9}$$

$$F_3(\boldsymbol{Y}) = W_3 * F_2(\boldsymbol{Y}) + B_3. \tag{10}$$

Note that (10) is different from (3) because $F_3(\boldsymbol{Y})$ has $K \times K$ separate-channels. Namely, the size of W_3 is $n_2 \times f_3 \times f_3 \times (K \times K)$. Let $F_{3.1}(\boldsymbol{Y})$, $F_{3.2}(\boldsymbol{Y})$, $F_{3.3}(\boldsymbol{Y})$, $F_{3.4}(\boldsymbol{Y})$ ... be separate-channels of $F_3(\boldsymbol{Y})$. Then, the super-resolution image $F(\boldsymbol{Y})$ is synthesized as

$$F(\boldsymbol{Y}) = \uparrow \mathrm{A}(F_{3.1}(\boldsymbol{Y})) + \uparrow \mathrm{B}(F_{3.2}(\boldsymbol{Y})) + \uparrow \mathrm{C}(F_{3.3}(\boldsymbol{Y})) + \uparrow \mathrm{D}(F_{3.4}(\boldsymbol{Y})) + \ldots. \tag{11}$$

Type B architecture can reduce the calculation costs compared to Type A, because only the third layer has $K \times K$ channels.

3.3 Type C: Multi-Channel CNN with Rotary-Averaging Technique

Even if the user rotates images by 90, 180, or 270 degrees, super-resolution must work at the same level of quality. Thus, we propose a rotary-averaging technique, as shown in Fig. 7.

In the Type C architecture, an input image is rotated by 0, 90, 180 and 270 degrees and they are each magnified by the Type B method. Then, four output images are reversed by 0, −90, −180 and −270 degrees. Let $R_\theta(\boldsymbol{Y})$ be a rotation operator by θ. The four images are calculated as

$$F_\theta(\boldsymbol{Y}) = R_{-\theta}\left(F\left(R_\theta(\boldsymbol{Y})\right)\right). \tag{12}$$

The above process is equivalent to rotating the whole CNN by 0, 90, 180 and 270 degrees. If the internal filters of the CNN are symmetrical, the four images will be the same.

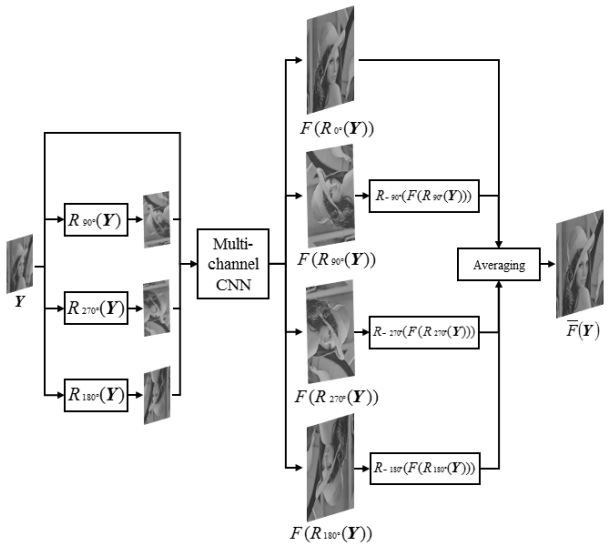


Fig. 7 Type C: Multi-channel CNN with rotary-average technique.

However, it is very rare. So, we obtain the final output by averaging them as

$$\bar{F}(\boldsymbol{Y}) = (F_{0^\circ}(\boldsymbol{Y}) + F_{90^\circ}(\boldsymbol{Y}) + F_{270^\circ}(\boldsymbol{Y}) + F_{180^\circ}(\boldsymbol{Y}))/4. \quad (13)$$

With this method, we can obtain a stable quality that does not depend on the orientation of input images.

4. Experiments and Results

First, the four types of CNN-based super-resolutions for 2×2 magnification are trained by the back-propagation technique in Sect. 4.2. Next, eight types of super-resolution techniques including state-of-the-art methods are applied to the 19 test images in Sect. 4.3. In Sect. 4.4, performances for 3×3 and 4×4 magnifications are shown. Finally in Sect. 4.5, total performances are compared based on their processing speed and PSNRs.

4.1 Experimental Conditions

Four types of CNN-based super-resolutions are
(i) Conventional SRCNN,
(ii) Type A: Parallel CNN,
(iii) Type B: Multi-channel CNN, and
(iv) Type C: Multi-channel CNN with a rotary-averaging technique.

Parameter settings of every CNN for 2×2 magnification are shown in Table 1. The filter sizes and the number of feature maps of the conventional SRCNN are $\{f_1, f_2, f_3\} = \{9, 5, 5\}$ and $\{n_1, n_2\} = \{64, 32\}$ in accordance with [11]. Although $\{f_1, f_2, f_3\} = \{9, 1, 5\}$ is also recommended in [11], the $\{9, 5, 5\}$ model implemented by ourselves shows better performance than the $\{9, 1, 5\}$ model. The filter sizes for our methods are $\{f_1, f_2, f_3\} = \{5, 5, 3\}$. The reason why they are small is that the input image size of our method is four times smaller than that of SRCNNs.

Table 1 Parameter setting of CNN. (2 × 2 magnification).

Parameter	SRCNN [11]	Type A, B, and C
Filter size $\{f_1, f_2, f_3\}$	{9, 5, 5}	{5, 5, 3}
Number of feature maps $\{n_1, n_2\}$	{64, 32}	{64, 32}

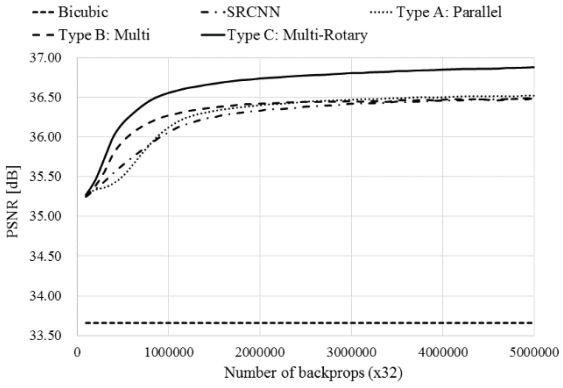


Fig. 8 Average PSNR curve for the number of back propagations (2×2 magnification).

In all experiments, we focus only on the luminance channel in the YCrCb space. PSNR (Peak Signal to Noise Ratio) is calculated only at the center area of the luminance channel to avoid the influence of the image boundaries. All CNNs are implemented with a *Caffe* package [14] on our PC (CPU: Intel Core i7-3770K with 8.00 GB of RAM and GPU: NVIDIA GeFoerce GTX 970 with 4 GB RAM).

4.2 Training of CNNs

All CNNs are trained by the back propagation technique with 91 test images [11]. These images are reduced to 1/(2×2) sizes with the bicubic method and again enlarged to original sizes by the methods (i)–(iv). The original images are used for teaching signals. The number of back propagation is 5 million × 32 (batch size) at the maximum. The learning rate is 10^{-4} for the first and second layers and 10^{-5} for the third layer. The above parameters are common for all methods.

Figure 8 shows the average PSNR curves for the number of back-propagations. A horizontal dashed line at 33.66 dB means PSNR for the bicubic method. We can find that Type B and Type C rise more rapidly than SRCNN and Type A. Since SRCNN and Type A have many filter coefficients in their CNNs, they converge slowly. When the number of back propagation achieves 5 million × 32, however, Type A shows almost the same PSNR as Type B. We think Type A has almost the same ability as Type B, but is inferior to Type B in terms of calculation cost.

On the other hand, PSNRs of Type C are obviously larger than other methods. This is owing to the rotary-averaging technique. This means that Type B does not have symmetric characteristics but Type C does. We think the symmetric architecture is very important to improve image

Table 2 PSNRs of 2×2 magnified images (Set 5).

Images	PSNR [dB]								
		Conventional Methods					Ours		
	Bicubic	ANR [7]	A+ [8]	NBSRF [9]	SRCNN [11]	CSC [12]	Type A: Parallel	Type B: Multi	Type C: Multi-Rotary
baby	36.95	38.33	38.413	38.40	38.22	**38.41**	38.20	38.16	38.38
bird	36.81	40.04	41.154	41.10	40.80	**41.58**	40.75	40.72	41.40
butterfly	27.49	30.66	32.235	33.03	32.60	32.26	32.80	32.81	**33.29**
head	34.74	35.56	**35.67**	**35.67**	35.55	35.65	35.54	35.52	35.62
women	32.30	34.66	35.41	35.52	35.27	35.49	35.32	35.25	**35.70**
average	33.66	35.85	36.58	36.74	36.49	36.68	36.52	36.49	**36.88**

Table 3 PSNRs of 2×2 magnified images (Set 14).

Images	PSNR [dB]								
		Conventional Methods					Ours		
	Bicubic	ANR [7]	A+ [8]	NBSRF [9]	SRCNN [11]	CSC [12]	Type A: Parallel	Type B: Multi	Type C: Multi-Rotary
baboon	24.86	25.56	25.66	25.70	25.65	25.67	25.66	25.66	**25.76**
Barbara	27.91	28.47	28.53	28.39	28.42	**28.62**	28.39	28.32	28.47
bridge	26.63	27.61	27.85	27.87	27.86	27.92	27.85	27.84	**27.98**
coastguard	29.22	30.40	30.40	30.57	30.54	30.45	30.52	30.50	**30.61**
comic	25.80	27.52	28.03	28.30	28.14	28.14	28.16	28.19	**28.43**
face	34.71	35.54	35.64	**35.65**	35.53	35.63	35.51	35.50	35.59
flowers	30.16	32.09	32.83	32.99	32.88	32.95	32.95	32.93	**33.27**
foreman	35.43	38.19	39.35	39.45	38.95	**39.49**	38.92	38.90	39.10
Lenna	34.55	36.21	36.48	36.54	36.37	36.58	36.39	36.37	**36.59**
man	29.14	30.36	30.78	30.82	30.76	30.86	30.78	30.78	**30.94**
monarch	32.73	35.52	36.82	37.48	37.12	36.91	37.27	37.26	**37.76**
pepper	35.00	36.51	37.08	37.07	36.96	**37.16**	36.94	36.95	37.11
ppt3	26.64	28.73	29.85	**30.50**	29.86	29.88	29.70	29.80	30.17
zebra	30.46	32.94	33.48	**33.81**	33.46	33.60	33.39	33.32	33.76
average	30.23	31.83	32.34	32.51	32.32	32.42	32.32	32.31	**32.54**

qualities.

4.3 PSNRs for Eight Types of Super-Resolutions

Next, we compare eight types of super-resolution techniques including state-of-the-arts methods: ANR [7], A+ [8], NBSRF [9], SRCNN [11], CSC [12], Type A, Type B, and Type C. Codes of ANR, A+, NBSRF, and CSC were implemented and distributed by the authors. They were trained with the same database consisting of 91 images [11]. The filters in SRCNN and our CNNs are fixed after finishing the back-propagation at 5 million × 32. The eight methods were applied to image set of Set 5 and Set 14 [11]. Note that those images are not included in the 91 training images.

Tables 2 and 3 show PSNRs for 2×2 magnified images of Set 5 and Set 14, respectively. Bold letters mean the highest PSNR in each image. We can see that Type C, NBSRF and CSC show good results. Although Type C is not always the best method, its average PSNR achieves 36.88 dB for Set 5 and 32.54 dB for Set 14, which are larger than any of the other averages.

Table 4 Parameter setting of CNN for $K \times K$ magnification.

Parameter	SRCNN [11]	Type A, B and C
Filter size $\{f_1, f_2, f_3\}$	{9, 5, 5}	{5, 3, 1}
Number of feature maps $\{n_1, n_2\}$	{64, 32}	{128, 64}

4.4 Image Quality for $K \times K$ Magnification

Next, we compare image qualities for $K \times K$ magnifications ($K = 3, 4$). Table 4 shows the parameter settings of CNNs. The filter sizes of our methods are $\{f_1, f_2, f_3\} = \{5, 3, 1\}$. They are small because the input image size is only $1/(K \times K)$ of the output image size. The number of feature maps of our methods are $\{n_1, n_2\} = \{128, 64\}$. They are large because Type B and C need $K \times K$ channels for the third layer of CNN, while SRCNN needs only a single channel.

Figures 9, 10 and 11 show examples of magnified images. Visually good images are obtained by (d) A+, (e) NBSRF, (g) CSC, and (j) Type C. In Fig. 11, however, (h) Type A and (i) Type B generate some noises like a mosquito around the edges, although their PSNRs are not so bad. This phenomenon often appears when K is large. We think the

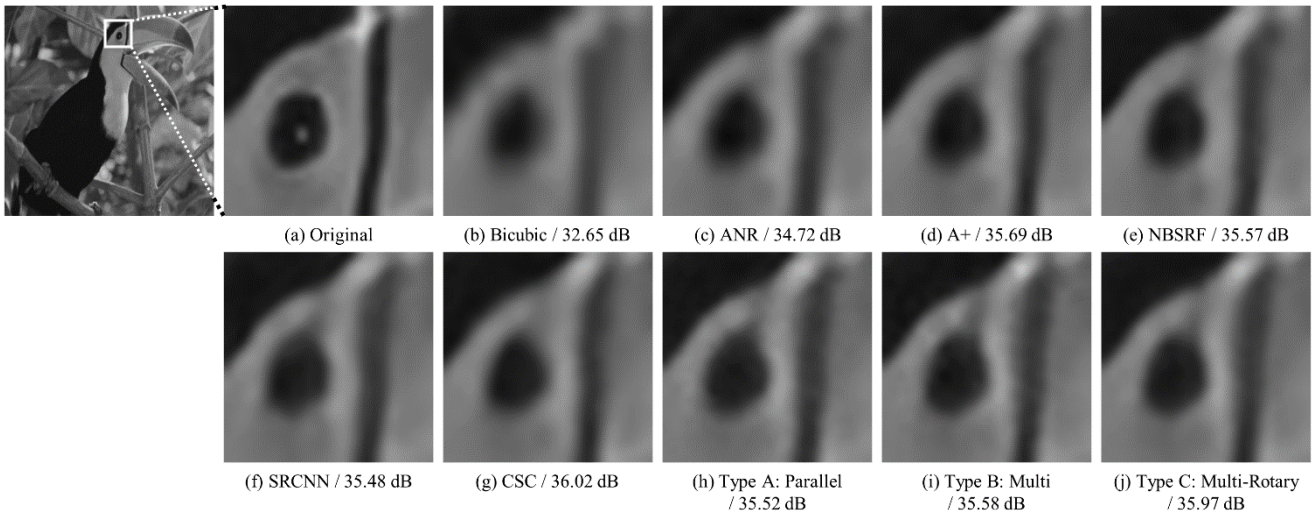


Fig. 9 A part of "bird" by eight super-resolution techniques (3×3 magnification).

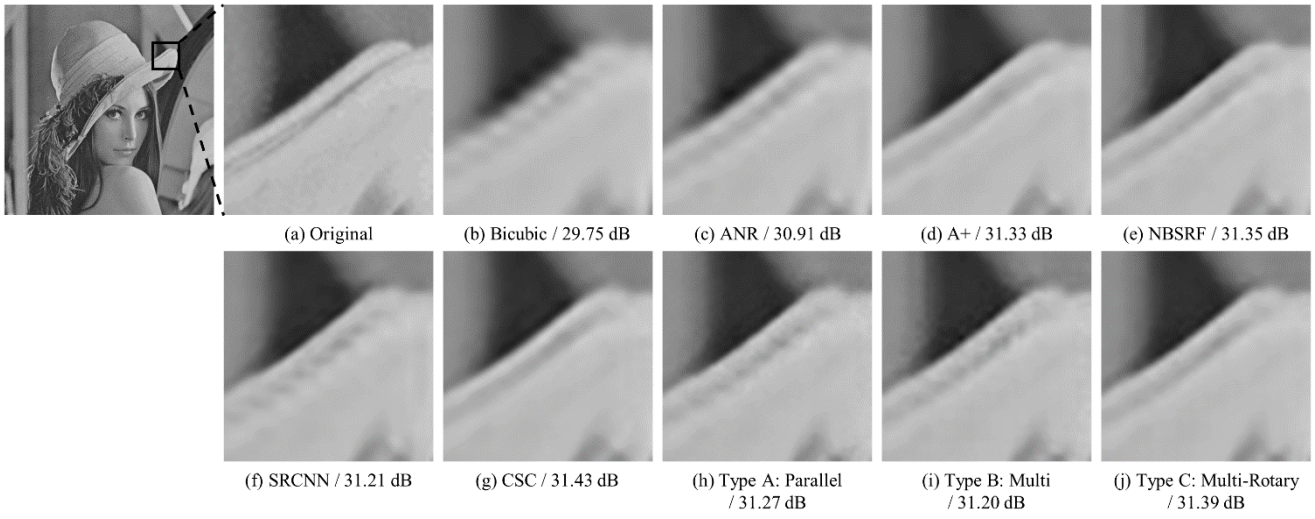


Fig. 10 A part of "Lenna" by eight super-resolution techniques (4×4 magnification).

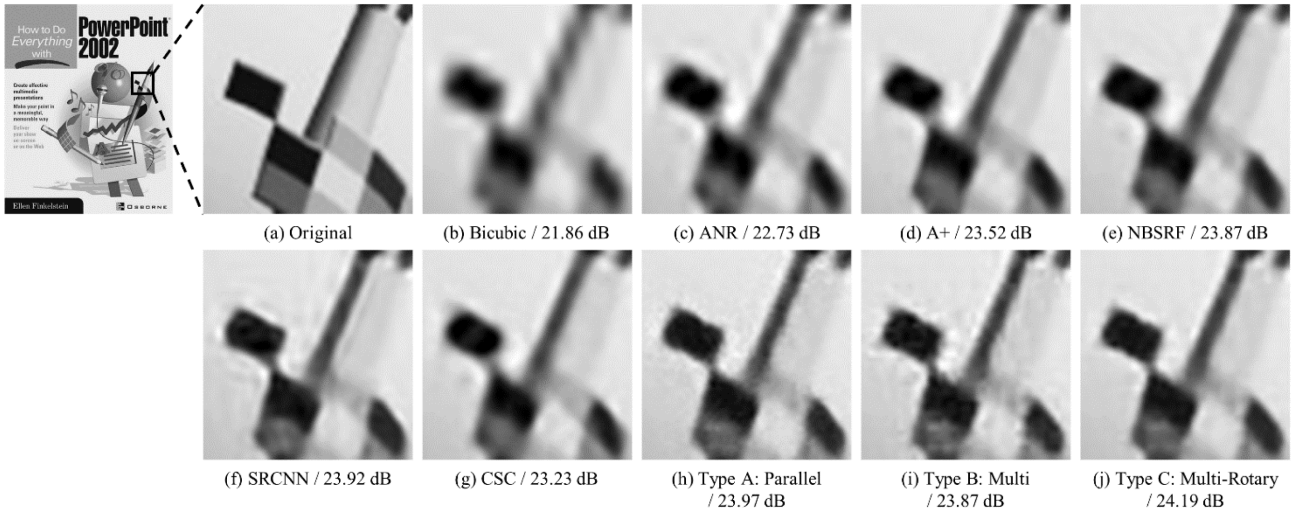


Fig. 11 A part of "ppt3" by eight super-resolution techniques (4×4 magnification).

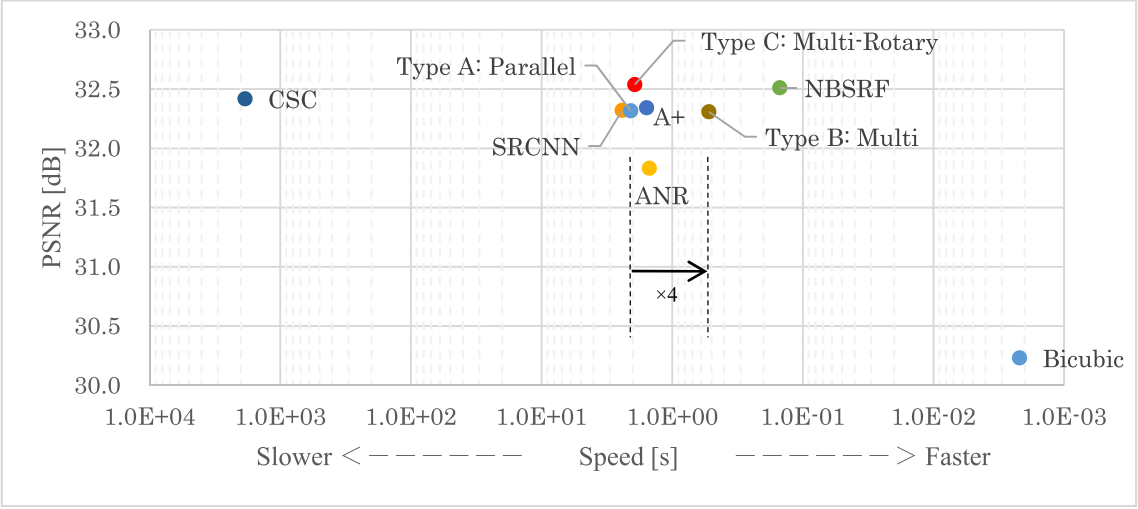


Fig. 12 Processing speed and PSNR (2×2 magnification).

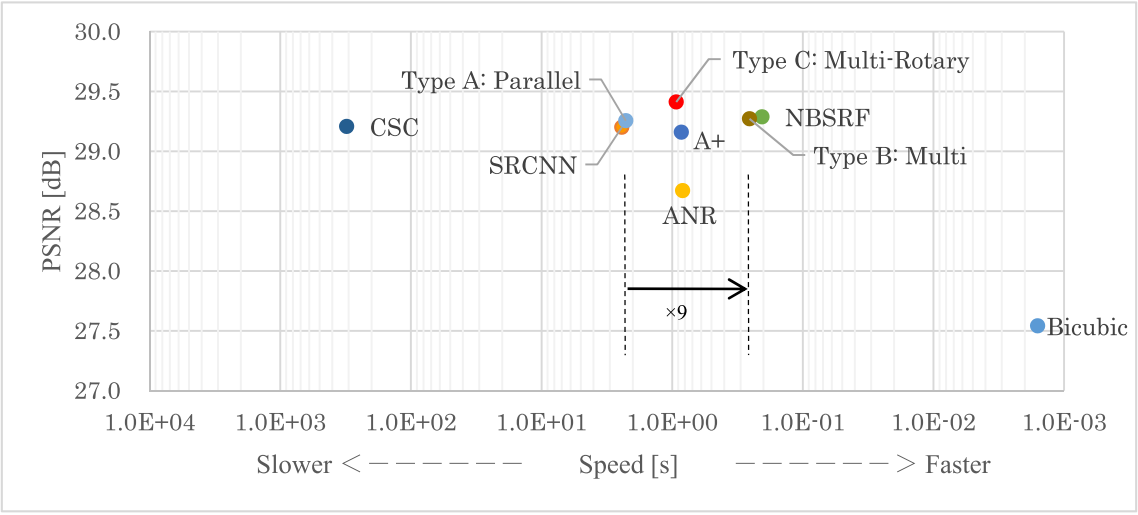


Fig. 13 Processing speed and PSNR (3×3 magnification).

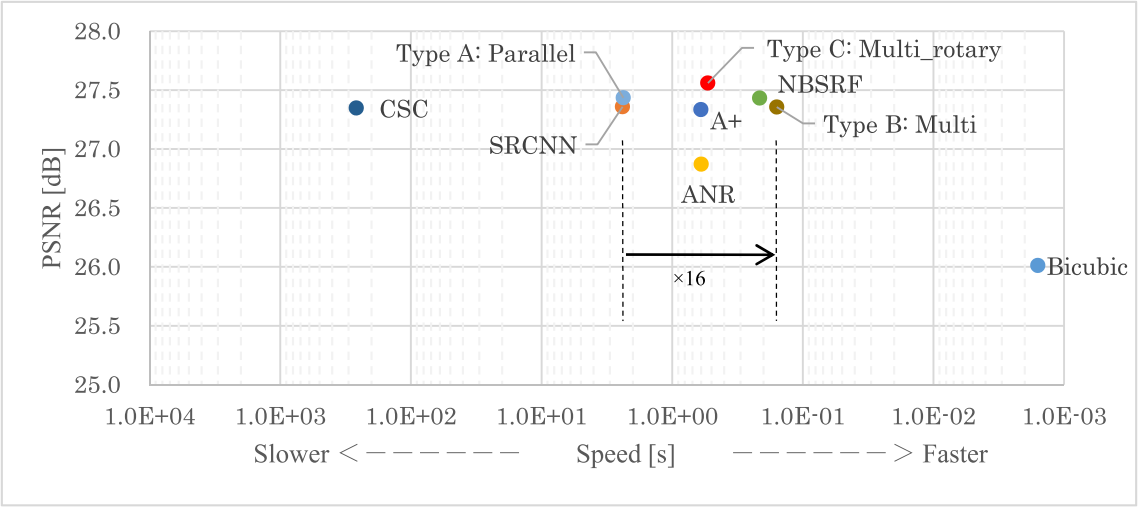


Fig. 14 Processing speed and PSNR (4×4 magnification).

reason is due to a variety of characteristics of $K \times K$ output-channels. For example, Types A and B generate 16 output images for 4×4 magnification. Since all parameters in CNNs are randomly initialized before the back propagation process, the variance of 16 networks becomes large and causes the variance of 16 pixel values. One good way to improve this deterioration is the rotary-averaging technique. In Fig. 11(j), we can see Type C suppress these noises successfully because the rotary-averaging technique smoothes the characteristics of the 16 networks. Thus, Type C can generate visually good images.

4.5 Processing Speed and PSNR

Finally, we show total performances based on processing speed and PSNRs. In order to compare the eight methods fairly, GPU acceleration is not used in this stage. Figure 12 is for 2×2 magnification. We can see that Type B is the fastest among CNN-based methods. This is because Type B consists of a single CNN and its internal filters are small. Type A is about four times slower than Type B because it consists of 2×2 individual CNNs. SRCNN is also slower than Type B because the input image is enlarged 2×2 times by the bicubic method before the CNN process.

In terms of PSNR, Type C achieves the highest value of 32.54 dB, although it is four times slower than Type B, owing to the rotary-averaging technique. The second highest PSNR of 32.51 dB is achieved by NBSRF. It is faster than CNN-based methods with no GPU acceleration. The third highest PSNR of 32.42 dB is achieved by CSC, but it is 963 times slower than Type C.

Figures 13 and 14 are for 3×3 and 4×4 magnifications, respectively. We can find that NBSRF is not very fast for 3x3 and 4x4 magnifications because it must enlarge input images before the patch search process. Type B is about $K \times K$ times faster than Type A because the former has a single CNN while the latter has $K \times K$ individual CNNs. Thus, a multi-channel architecture is useful to reduce calculation costs. Moreover, we can find that the Type C method achieves the highest PSNR, although it is always four times slower than Type B because of the rotary-averaging technique. Thus, Type C is superior to other super-resolution techniques in terms of PSNR. If we want high PSNRs, Type C would be the best method.

5. Conclusion

In this paper, we proposed three types of super-resolution techniques using CNNs: Type A (Parallel CNNs), Type B (Multi-channel CNN), and Type C (Multi-channel CNN with the rotary-averaging technique). Our techniques did not use the bicubic interpolation but directly generated $K \times K$ magnified images from CNNs. It enabled a perfect end-to-end learning between low and high resolution images. In our experiments, PSNRs for Type A were about 2 dB higher than those for the bicubic method, but its processing speed was not so fast because it needs $K \times K$ individual CNNs for $K \times K$ magnification. Type B overcame the above problem by using a single CNN architecture with $K \times K$ output-channels. It worked about $K \times K$ times faster than Type A while keeping image qualities. Moreover, Type C improved image qualities with the rotary-averaging technique and was the best method among the eight super-resolution techniques in terms of PSNR. We have ensured the utilities of our CNN architectures. Implementation for color images will be our future work.

Acknowledgment

Part of this work was supported by JSPS KAKENHI Grant Number 15K00077 and 24500063.

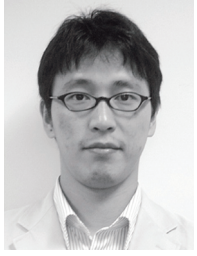
References

[1] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael, "Learning low-level vision," Int. J. Comput. Vis., vol.40, no.1, pp.25–47, Oct. 2000.
[2] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2008.
[3] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Trans. Image Process., vol.19, no.11, pp.2861–2873, 2010.
[4] K.I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.6, pp.1127–1133, 2010.
[5] H. Chang, D.Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," IEEE Conference on Computer Vision and Pattern Recognition, pp.275–282, 2004.
[6] M. Bevilacqua, A. Roumy, C. Guillemot, and M.L.A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," British Machine Vision Conference, pp.135.1–135.10, 2012.
[7] R. Timofte, V.D. Smet, and L.V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," IEEE International Conference on Computer Vision, pp.1920–1977, 2004.
[8] R. Timofte, V.D. Smet, and L.V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," Asian Conference on Computer Vision, pp.111–126, 2014.
[9] J. Salvador, and E. Pérez-Pellitero, "Naive Bayes super-resolution forest," IEEE International Conference on Computer Vision, pp.325–333, 2015.
[10] C. Dong, C.C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," European Conference on Computer Vision, vol.8692, pp.184–199, 2014.
[11] C. Dong, C.C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.2, pp.295–307, 2016.
[12] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang, "Convolutional sparse coding for image super-resolution," IEEE International Conference on Computer Vision, pp.1823–1831, 2015.
[13] S. Ohtani, Y. Kato, N. Kuroki, T. Hirose, and M. Numa, "Super-resolution with four parallel convolutional neural networks," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J99-D, no.5, pp.588–593, May 2016.
[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," ACM International Conference on Multimedia, pp.675–678, 2014.

Shinya Ohtani received a B.S. and M.S. degree in Electrical and Electronic Engineering from Kobe University in 2014 and 2016, respectively. He is now with Toyota Motor Corporation. His research interests include robot vision.

Yu Kato received a B.S. degree in Electrical and Electronic Engineering from Kobe University in 2015. He is now a master's student at Kobe University. His research interests include digital image processing.

Nobutaka Kuroki received the B.E., M.E., and Dr. Eng. degrees in Electronic Engineering from Kobe University in 1990, 1992, and 1995, respectively. Since 1995 to 2005, he was a Research Associate in the Department of Electrical and Electronic Engineering, Kobe University. He is currently an Associate Professor since 2006. His research interests include digital signal processing and digital image processing. Dr. Kuroki is a member of the IEEJ, ITE, and RISP.



Tetsuya Hirose received the B.S., M.S., and Ph.D. degrees from Osaka University in 2000, 2002, and 2005, respectively. From 2005 to 2008, he was a Research Associate at the Department of Electrical Engineering, Hokkaido University. He is currently an Associate Professor of the department of Electrical and Electronics Engineering, Kobe University. His current research interests are in the field of low-power analog/digital integrated circuits design and subthreshold MOSFET functional LSIs for intelligent sensors. Dr. Hirose is a member of the Japan Society of Applied Physics (JSAP).

Masahiro Numa received the B.E., M.E., and Dr. Eng. degrees in Precision Engineering from the University of Tokyo in 1983, 1985, and 1988, respectively. From 1986 to 1989, he had been a Research Associate in the Department of Precision Engineering at the University of Tokyo, where he became a Lecturer in 1989. After moving to Kobe University in 1990, he joined the Department of Electrical and Electronic Engineering as an Associate Professor in 1995, and has been a Professor since 2004. He visited University of California, Santa Barbara, U. S. A., as a visiting scholar in 1996. His research interests include CAD and low-power design methodologies for VLSI, and image processing. Prof. Numa is a member of the ACM and IPSJ.