IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2020, accepted April 16, 2020, date of publication May 6, 2020, date of current version May 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992483

Mathematical Analysis of DCN-Based Super-Resolution

C. LEE, (Member, IEEE), J. PARK, S. WOO, J. KIM, AND J. YOON
School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: C. Lee (chulhee@yonsei.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2018R1D1A1B07050345.

ABSTRACT Although DCN-based super-resolution (DCN-SR) techniques have shown impressive performance, the working mechanism has not been completely understood and DCN-SR methods still produce some artefacts. In this paper, we analyze the working mechanisms of DCN-SR methods. We derive mathematical formulations of the DCN-SR methods and provide some experimental analyses, which show that the effective receptive fields of the DCN-SR methods are considerably smaller than the theoretical receptive fields. Based on the mathematical formulations, experiments were performed. The results indicates that current DCN-SR methods may have some fundamental problems and new types of DCN structures are needed for reliable super-resolution performance.

INDEX TERMS Deep convolutional networks, effective receptive field, mathematical formulation of DCN, ReLU, super-resolution.

I. INTRODUCTION

Recently, deep convolutional networks (DCN) have been successfully applied in many signal processing areas [1]–[5] such as super-resolution [11], and noise removal [13], de-mosaicking [12], etc. In particular, a number of researchers have studied DCN-based super-resolution (DCN-SR) methods, which have provided noticeably better performance compared to traditional super-resolution and interpolation methods [14]–[30]. However, the working mechanism of DCN-based super-resolution (DCN-SR) methods has not always been well understood. Some authors have studied the working models of the DCN methods [6]–[10]. In [6], the authors provided some insight into the intermediate layers with visualization techniques. Also, it was shown that the first-layer features may not be specific to a particular task and can be transferable to other tasks [7]. Some visualizing tools were proposed for DCN [8], which provided some insight into the DCN working mechanism. In [9], the author presented some analysis results of the DCN operations. In [10], the authors found some interesting properties of neural networks and showed that imperceptible perturbation may produce errors in neural networks. In [31], the authors investigated the mathematical model of deep learning frameworks for inverse problems.

The associate editor coordinating the review of this manuscript and approving it for publication was Yong Yang.

II. MATHEMATICAL FORMULATIONS OF DCN-SR METHODS

In general, DCN-SR methods have a number of convolution layers, ReLU layers, etc. They may further include other types of layers (e.g., channel attention layers, sigmoid functions, etc. [16]). In most DCN-SR methods, networks are trained using a number of image patches (K×K images). These patches can be expressed as vectors (N×1) with $N = K^2$. For example, the first convolution layer followed by a ReLU layer can be expressed by the following matrix operation:

$$X^1_{64N\times 1} = \mathrm{ReLU}\left(A^0_{64N\times N} X^0_{N\times 1} + b^0_{64N\times 1}\right) \tag{1}$$

where the superscript represents the layer index. $A^0_{64N\times N}$ is a filter matrix that represents the convolution operations with 64 filters and $b^0_{64N\times 1}$ is a bias vector:

$$\begin{aligned} b^0_{64N\times1} = [&b^0_0, b^0_0, \cdots, b^0_0, b^0_1, b^0_1, \\ &\cdots, b^0_1, \cdots, b^0_{63}, b^0_{63}, \cdots, b^0_{63}]^T \end{aligned}$$

$$\mathbf{A^0_{64N\times N}} = \begin{bmatrix} A^{0,0}_{N\times N} \\ A^{0,1}_{N\times N} \\ \vdots \\ A^{0,63}_{N\times N} \end{bmatrix}$$

where $A^{0,k}_{N\times N}$ is a single filter matrix (N×N) of the 0^{th} layer for the k^{th} filter. $X^0_{N\times 1}$ is an input patch and $X^1_{64N\times 1}$ is the output of the first layer (convolution layer + ReLU layer). In (1), $(A^0_{64N\times N}X^0_{N\times 1}+b^0_{64N\times 1})$ is a vector. If an element of the vector is negative, the ReLU operator sets it to zero. Thus, we can express (1) as follows:

$$X^1_{64N\times 1} =_R A^0_{64N\times N}X^0_{N\times 1} +_R b^0_{64N\times 1} \tag{2}$$

where the left subscript (R) represents the ReLU operation. In other words, a row of matrix ${}_RA^0_{64N\times N}$ and an element of vector ${}_Rb^0_{64N\times 1}$ are set to zero if the corresponding element of $A^0_{64N\times N}X^0_{N\times 1}+b^0_{64N\times 1}$ is negative. We will define ${}_RA^0_{64N\times N}$ as a layer matrix, which applies convolution and ReLU operations. For the VDSR [17], there are 20 layers and the final output image can be expressed as follows:

$$\begin{aligned} X^{20}_{N\times1} &= A^{19}_{N\times64N}X^{19}_{64N\times1} +_R b^{19}_{N\times1} \\ &= A^{19}_{N\times64N}\left(A^{18}_{64N\times64N}X^{18}_{64N\times1} +_R b^{18}_{64N\times1}\right) +_R b^{19}_{N\times1} \\ &= A^{19}_{N\times64N}A^{18}_{64N\times64N}\ X^{18}_{64N\times1} + A^{19}_{N\times64N}\,{}_Rb^{18}_{64N\times1} +_R b^{19}_{N\times1} \end{aligned}$$

Consider the center pixel of the output image (Fig. 1), which can be expressed as follows:

$$y = \mathrm{ReLU}\left(\varphi^{\mathrm{center},19}_{1\times64N}X^{19}_{64N\times1} + b^{19}_{1\times1}\right)$$

where $\varphi^{\mathrm{center},19}_{1\times64N}$ is the row of $A^{19}_{N\times64N}$ corresponding to the center pixel. Using

$$X^k_{64N\times1} =_R A^{k-1}_{64N\times64N}X^{k-1}_{64N\times1} +_R b^{k-1}_{64N\times1} \quad (k = 1\text{-}18),$$

the center pixel of the output image is given by

$$\begin{aligned} y &= W_{1\times N}X^0_{N\times1} + \sum_{k=0}^{18} Q^k_{1\times64N}b^k_{64N\times1} +_R b^{19}_{1\times1} \\ &= W_{1\times N}X^0_{N\times1} + b^{\mathrm{total}}_{1\times1} \end{aligned} \tag{3}$$

where

$$\begin{aligned} W_{1\times N} &= \varphi^{\mathrm{center},19}_{1\times64N}\,{}_RA^{18}_{64N\times64N}\cdots{}_RA^{1}_{64N\times64N}\,{}_RA^{0}_{64N\times N} \\ &= \varphi^{\mathrm{center},19}_{1\times64N}\prod_{j=0}^{18}{}_RA^j = Q^0_{1\times64N}\,{}_RA^0_{64N\times N} \\ Q^k_{1\times64N} &= \varphi^{\mathrm{center},19}_{1\times64N}\prod_{j=k+1}^{18}{}_RA^j \\ b^{\mathrm{total}}_{1\times1} &= \sum_{k=0}^{18}Q^k_{1\times64N}b^k_{64N\times1} + {}_Rb^{19}_{1\times1} \end{aligned}$$

In other words, an output pixel is obtained by taking the inner product between the weight mask ($W_{1\times N}$) and the input patch ($X_{N\times1}$) and adding the total bias ($b^{\mathrm{total}}_{1\times1}$). Also, the total bias term ($b^{\mathrm{total}}_{1\times1}$) can be expressed:

$$b^{\mathrm{total}}_{1\times1} = \sum_{k=0}^{L-1}\sum_{j=0}^{n^k_{\mathrm{bias}}}\alpha^k_j b^k_j = W_{1\times N_b}B_{N_b\times1}$$

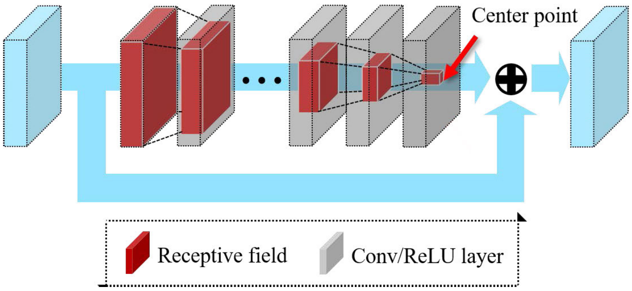

FIGURE 1. The center pixel of the output layer in the VDSR.

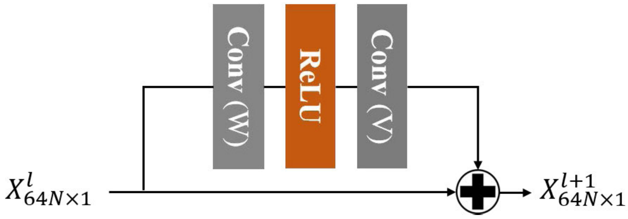

FIGURE 2. Residual block.

where n^k_{bias} is the number of bias terms (number of filters) of the k^{th} layer and N_b is the total number of biases. $W_{1\times N_b}$ is a bias weight vector and $B_{N_b\times1}$ is a bias vector. Thus, if a DCN-SR method consists of convolution and ReLU layers, the output can be expressed as follows:

$$\begin{aligned} y &= W_{1\times N}X^0_{N\times1} + \sum_{k=0}^{L-1}\sum_{j=0}^{n^k_{\mathrm{bias}}}\alpha^k_j b^k_j \\ &= W_{1\times N}X^0_{N\times1} + W_{1\times N_b}B_{N_b\times1} \end{aligned} \tag{4}$$

It is noted that $W_{1\times N}$ and $W_{1\times N_b}$ are functions of the input patch. In other words, the two terms can be expressed as follows:

$$y = W_{1\times N}\left(X^0_{N\times1}\right)X^0_{N\times1} + W_{1\times N_b}\left(X^0_{N\times1}\right)B_{N_b\times1} \tag{5}$$

Some DCN-SR methods (e.g., EDSR [15]) use residual blocks (Fig. 2). Similarly, we can express the residual block as follows:

$$\begin{aligned} X^{l+1}_{64N\times1} &= V^l_{64N\times64N}\left({}_RW^l_{64N\times64N}X^l_{64N\times1} +_R b^l_{64N\times1}\right) \\ &\quad + q^l_{64N\times1} + X^l_{64N\times1} \\ &= \left(V^l_{64N\times64N}\ {}_RW^l_{64N\times64N} + I_{64N\times64N}\right)X^l_{64N\times1} \\ &\quad + V^l_{64N\times64N}\,{}_Rb^l_{64N\times1} + q^l_{64N\times1} \\ &= R^l_{64N\times64N}\ X^l_{64N\times1} + V^l_{64N\times64N}\ {}_Rb^l_{64N\times1} \\ &\quad + q^l_{64N\times1} \end{aligned}$$

where $R^l_{64N\times64N} = V^l_{64N\times64N}\,{}_RW^l_{64N\times64N} + I_{64N\times64N}$ and $q^l_{64N\times1}$ is the bias vector of the second convolution operation ($V^l_{64N\times64N}$). In other words, any output pixel value can be expressed as a linear function. Thus, it can be shown that the output of the EDSR (enhanced deep super resolution network) can be expressed in a way similar to (3) and (4).

The RRDB (residual in residual dense block) [17] has a much larger number of layers and the theoretical receptive

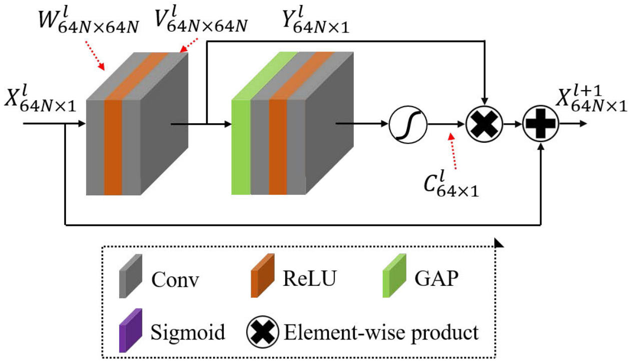

FIGURE 3. Residual channel attention block (RCAB).

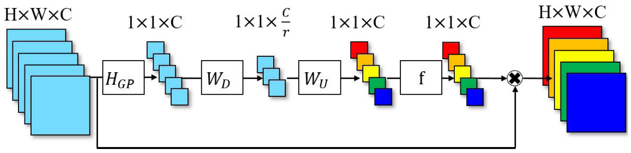

FIGURE 4. Channel attention layer (CA) of the RCAB.

field can cover the entire image. Nevertheless, the linear model of (4) is still valid for the RRDB since it only uses convolution and ReLU layers. The operation that combines several images into a higher resolution image can be also expressed as a matrix operation and is still a linear operator. Consequently, the dynamic linear model of (4) can be used to model the output pixel values of the RRDB.

For the RCAN (residual channel attention networks), the linear model of (4) may not be valid due to the multiplication operations and sigmoid functions of the channel attention layer (Fig. 3-4) [18]. Assuming 64 filters are used, the residual channel attention block (RCAB) can be expressed as follows:

$$X^{l+1}_{64N\times1} = Y^l_{64N\times1} \otimes C^l_{64\times1} + X^l_{64N\times1} \tag{6}$$

where $\otimes$ denotes element-wise (channel-wise) product and

$$Y^l_{64N\times1} = V^l_{64N\times64N}\left({}_R W^l_{64N\times64N} X^l_{64N\times1} +_R b^l_{64N\times1}\right) + q^l_{64N\times1}$$

The first term of (6) is not a linear operation since $C^l_{64\times1}$ is computed by a series of operations that include average operations and sigmoid functions. Thus, for the RCAN, the output pixel value needs to be expressed as follows:

$$y = W_{1\times N} X^0_{N\times1} + W_{1\times N_b} B_{N_b\times1} + f(X^0_{N\times1}, \{b^k_j\}) \tag{7}$$

where f is a non-linear function. Thus, the gradients of $C^l_{64\times1}$ with respect to the input image and the bias terms cannot be expressed as linear functions. However, due to the average operation, the contribution of each pixel would be much smaller in the non-linear function f. In other words, if the image size is $M \times N$, $C^l_{64\times1}$ always include a term $((1/MN)^m \; with \; m > 0)$ as follows:

$$C^l_{64\times1} = \left(\frac{1}{MN}\right)^m C'^l_{64\times1}.$$

Since MN is very large (in the order of $10^5 \sim 10^6$), the gradients of $f\left(X^0_{N\times1}, \{b^k_j\}\right)$ with respect to $X^0_{N\times1}$ and $\{b^k_j\}$ tend to be very small compared to the gradients of $W_{1\times N_b B_{N_b\times1}}$ and $W_{1\times N_b} B_{N_b\times1}$. Thus, even for the RCAN, we may approximate the output value using (4):

$$y \approx \tilde{y} = W_{1\times N} X^0_{N\times1} + W_{1\times N_b} B_{N_b\times1}. \tag{8}$$

Fig. 5 shows the difference histogram between the actual output (y of (7)) and the linear approximation ($\tilde{y}$ of (8)). The maximum difference was 7.02×10^{-4} for the 8-bit images. Thus, the dynamic linear model of (4) is still valid for the RCAN.

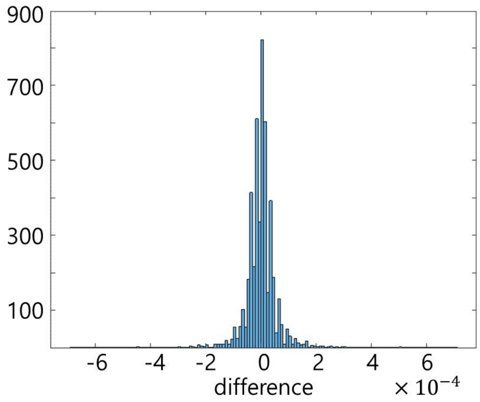

FIGURE 5. Difference histogram between the actual output (y) and the linear approximation ($\tilde{y}$). The maximum absolute difference is 7.02×10^{-4} for the 8-bit images.

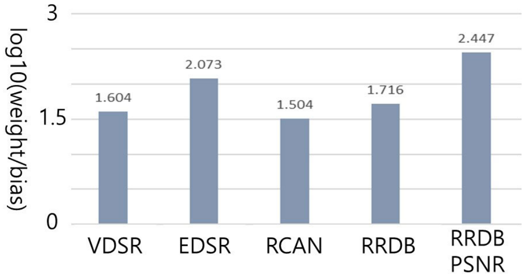

FIGURE 6. Log ratio of $W_{1\times N} X^0_{N\times1} / W_{1\times N_b} B_{N_b\times1}$ computed over 1600 training images.

III. EFFECTIVE RECEPTIVE FIELDS

As can be seen in (4), the output pixel value consists of two terms: $W_{1\times N} X^0_{N\times1}$ and $W_{1\times N_b} B_{N_b\times1}$. Fig. 6 shows the log ratios of $W_{1\times N} X^0_{N\times1} / W_{1\times N_b} B_{N_b\times1}$, which were computed using 1600 training images that were randomly selected from the DIV2K database. It appears that $W_{1\times N} X^0_{N\times1}$ is about 30-280 times larger than $W_{1\times N_b} B_{N_b\times1}$. Fig. 7 shows the contributions of the pixel and bias values when the RRDB DCN-SR methods were used for the baboon image. The other three methods (VDSR, EDSR, RCAN) have very small

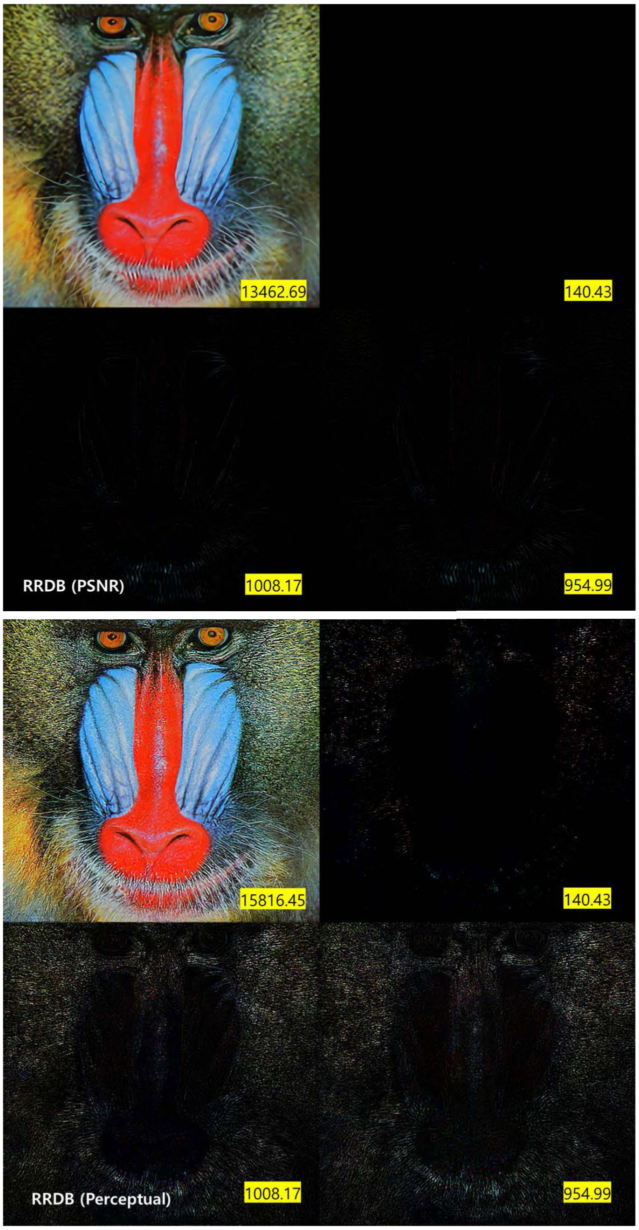

FIGURE 7. Contributions of pixels and bias values, and energy distribution (top-left: positive weight mask, top-right: negative weight mask, bottom-left: positive bias mask, bottom-right: negative bias mask).

bias values so that their bias contributions are invisible. The bias contribution mainly appeared around the edges. Also, $W_{1\times N}X^{0}_{N\times 1}$ and $W_{1\times N_b}B_{N_b\times 1}$ can have negative values, though the sum is rarely negative. The perceptual RRDB method showed considerably larger bias contributions than the other methods.

Since $W_{1\times N}$ and $W_{1\times N_b}$ are a function of the input patch, the weight mask ($W_{1\times N}$) of (4) will be different for each output pixel and may have negative values as can be seen in Fig. 7. Fig. 8 shows the average energy of $W_{1\times N}$ that was computed using the 1600 training images for the five methods (VDSR, EDSR, RCAN, RRDB(PSNR), RRDB(perceptual)). Table. 1 shows the theoretical receptive fields of the four methods, even though the training patch size was 41 × 41 for the VDSR and 48×48 for the other methods. Since the RRDB and RCAN have a large number of layers, their theoretical receptive fields can cover the entire image, even though the networks were trained using the 48 × 48 training patches. As can be seen in the figure, it is observed that the effective receptive field is much smaller. Fig. 9 shows the color coded energy distribution of the weight mask ($W_{1\times N}$).

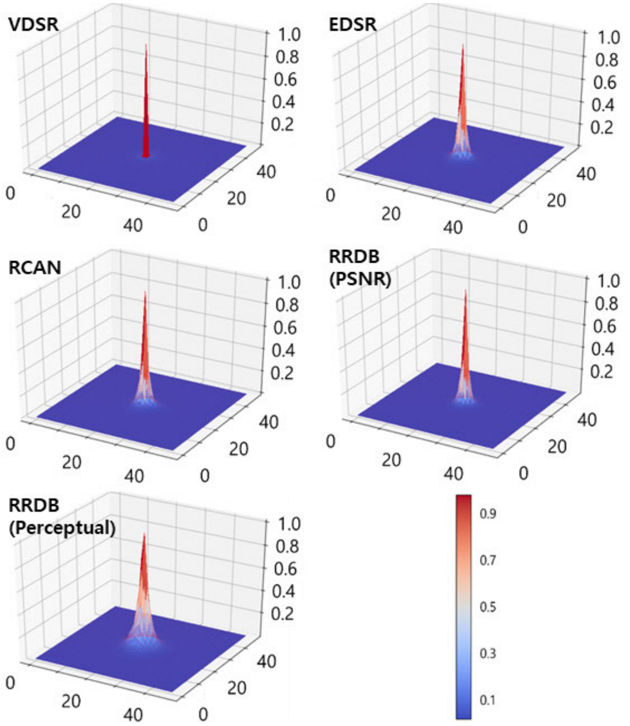

FIGURE 8. Average energy of the weight mask over 1600 training images.

TABLE 1. Theoretical receptive fields.

Models	Receptive Fields
VDSR	41 × 41
EDSR	64 × 64 (training patch size: 48 × 48)
RCAN	possibly entire image (training patch size: 48 × 48)
RRDB	possibly entire image (training patch size: 48 × 48)

It appears that the weight mask reflects image characteristics. For example, for the *baboon* image, the image mask had a circular shape (Fig. 10). On the other hand, for the *coastguard* image, the weight mask had an elliptic shape reflecting the horizontal river waves.

IV. EFFECTS OF BOUNDARY PIXELS

In (4), $W_{1\times N}$ and α^{k}_{j} are functions of the input image. Thus, we can rewrite (4) as follows:

$$y = W_{1\times N}\left(X^{0}_{N\times 1}\right)X^{0}_{N\times 1} + \sum_{k=0}^{L-1}\sum_{j=0}^{n^{k}_{\text{bias}}}\alpha^{k}_{j}\left(X^{0}_{N\times 1}\right)b^{k}_{j}. \quad (9)$$

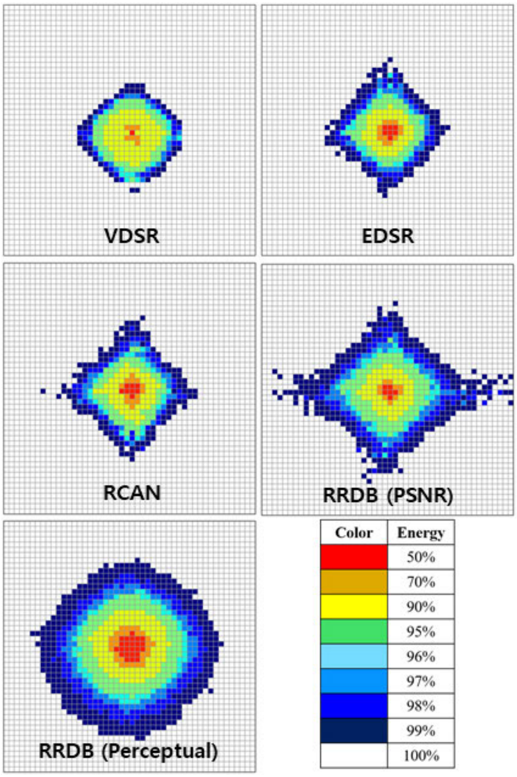

FIGURE 9. Average energy of the weight mask over 1600 training images.

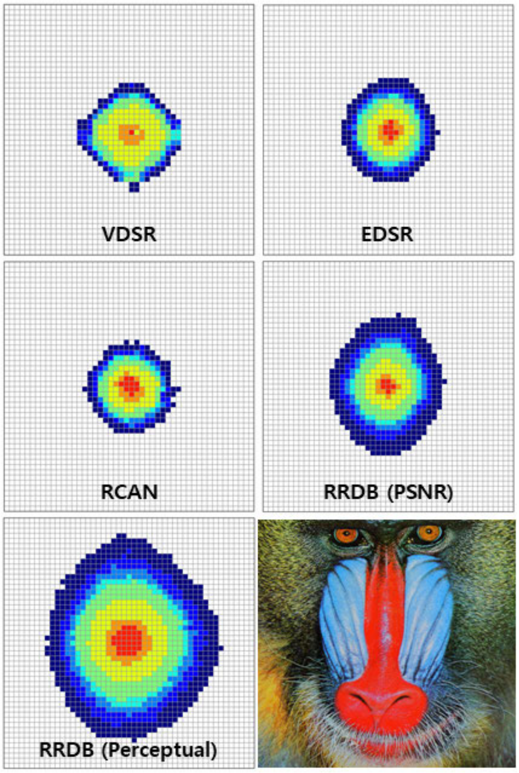

FIGURE 10. Average energy of the weight mask of the *baboon* image.

where L represents the number of layers that include bias terms. Although the effective reception field is much smaller than the theoretical reception field, it is still possible that

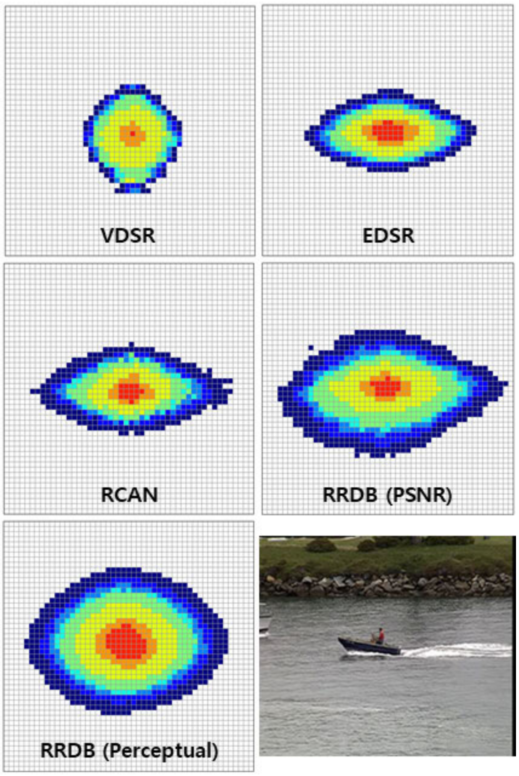

FIGURE 11. Average energy of the weight mask of the *coastguard* image.

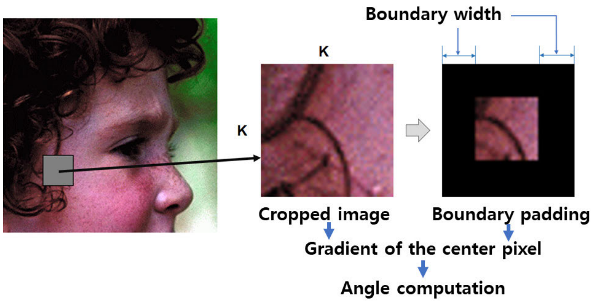

FIGURE 12. Setting the boundary pixels to zero.

TABLE 2. PSNR comparison of DCN-SR methods with block padding (averages of seven images).

	ALL	B16	B20	B24	B28	B32	B36	B40
VDSR	25.41	25.10	25.46	25.65	25.70	25.73	25.75	25.75
EDSR	27.26	25.89	26.27	26.50	26.69	26.81	26.91	26.98
RCAN	27.32	25.85	26.20	26.49	26.67	26.78	26.88	26.96
RRDB	27.44	25.91	26.30	26.55	26.71	26.84	26.93	27.03

boundary pixels may affect $W_{1\times N}$ and α_j^k. Consequently, the boundary pixels may play a certain role in determining the output images. In order to investigate how the boundary pixels might affect $W_{1\times N}(X)$ and $\alpha_j^k(X)$, we extracted a patch from an image. The patch size was the same as used in the training procedure (41 × 41 for the VDSR and 48 × 48 for the others). Then, we set the boundary pixels of input patches to zero (Fig. 12) and applied a DCN-SR method.

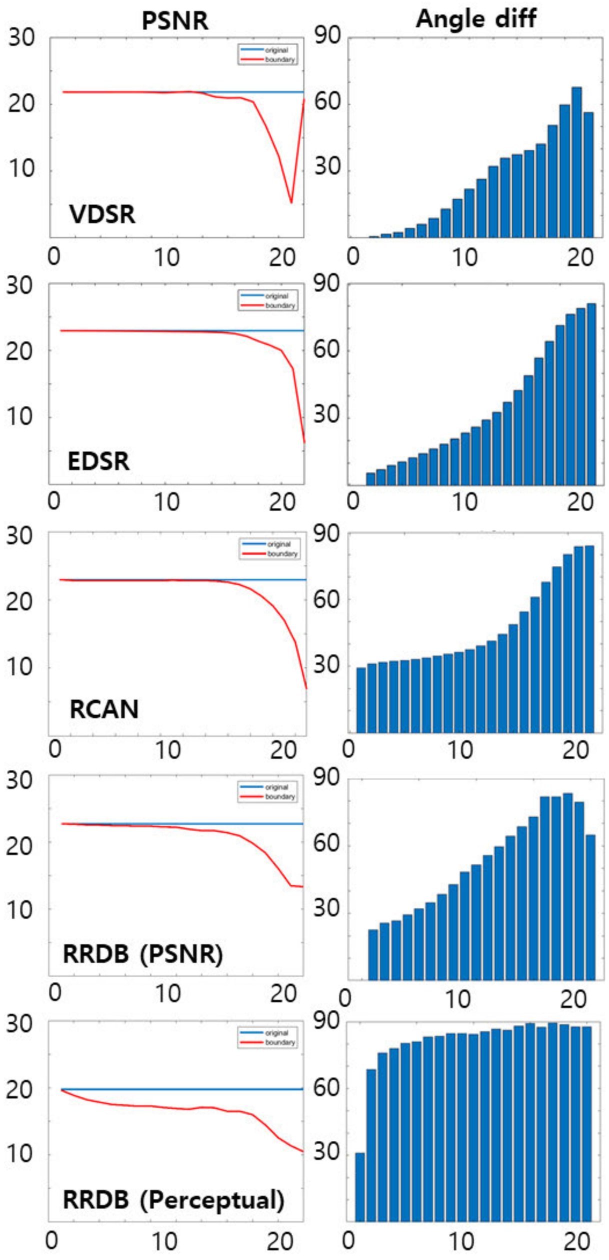

FIGURE 13. Setting the boundary pixels to zero.

Next, we computed the gradient of the center pixel and kept only the center pixel as the output. We repeated this procedure for the 1600 patches. We increased the zero padding from 1 to 20 (Fig. 12). We also compared the weight mask ($W_{1\times N}$) of the original patches and the zero-padded patches, and computed the angles between the two vectors (masks) as follows:

$$\theta = \cos^{-1}\left(\frac{W_{1\times N}\cdot W_{b,1\times N}}{||W_{1\times N}||||W_{b,1\times N}||}\right)$$

where $W_{b,1\times N}$ represents the weight mask for the zero-padded patch, $\cdot$ represents the dot product and $||\cdot||$ represents the norm of the vector. Fig. 13 shows the output pixel value differences of the original patches and the zero-padded patches (averages of the 1600 samples). It can be seen that zero padding up to 10 pixels produced very small differences except for with the RRDB (perceptual). These results indicate that the effective receptive field DCN-SR methods may be much smaller than expected.

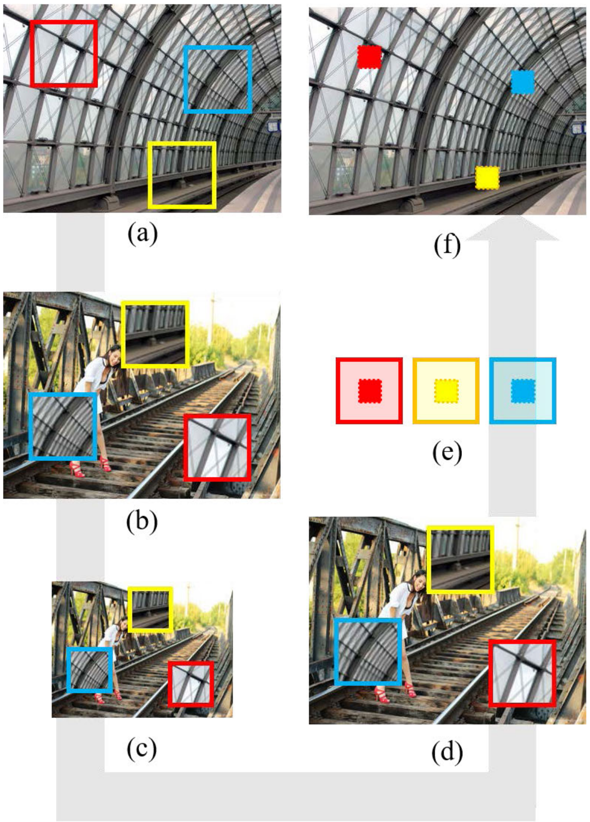

FIGURE 14. Pipeline of block image experiment. (a) Crop block images with stride 4. (b) Randomly synthesize block images to background image. (c) Generate LR image (bicubic interpolation). (d) DCN based super-resolution. (e) Crop 4 × 4 center region from block images. (d) Reposition to original location.

Next, we conducted another experiment. We extracted a block from a target image (Fig. 14(a)) and put it into a background image (Fig. 14(b)). The background image with embedded blocks was reduced (Fig. 14(c)) and then enlarged using a DCN-SR method ((Fig. 14(d)). Finally, only the pixels of the center block (4 × 4) were retained (Fig. 14(e)). We repeated this procedure so that the entire target image was processed using the DCN-SR method. We computed the PSNR between the enlarged target image through block padding and the original high-resolution image. We compared this PSNR with conventional PSNR values that were computed without block padding. We varied the block size (from 4 × 4 (B16) to 10 × 10 (B40) in the low resolution images). Table 2 shows the PSNR comparison (averages of seven images). A PSNR decrease by 0.4∼0.5dB was observed for the EDSR, RCAN and RRDB when the block size was 7 to 10 in the low resolution images. For the VDSR, when the block size was 5 to 10, the PSNR performance

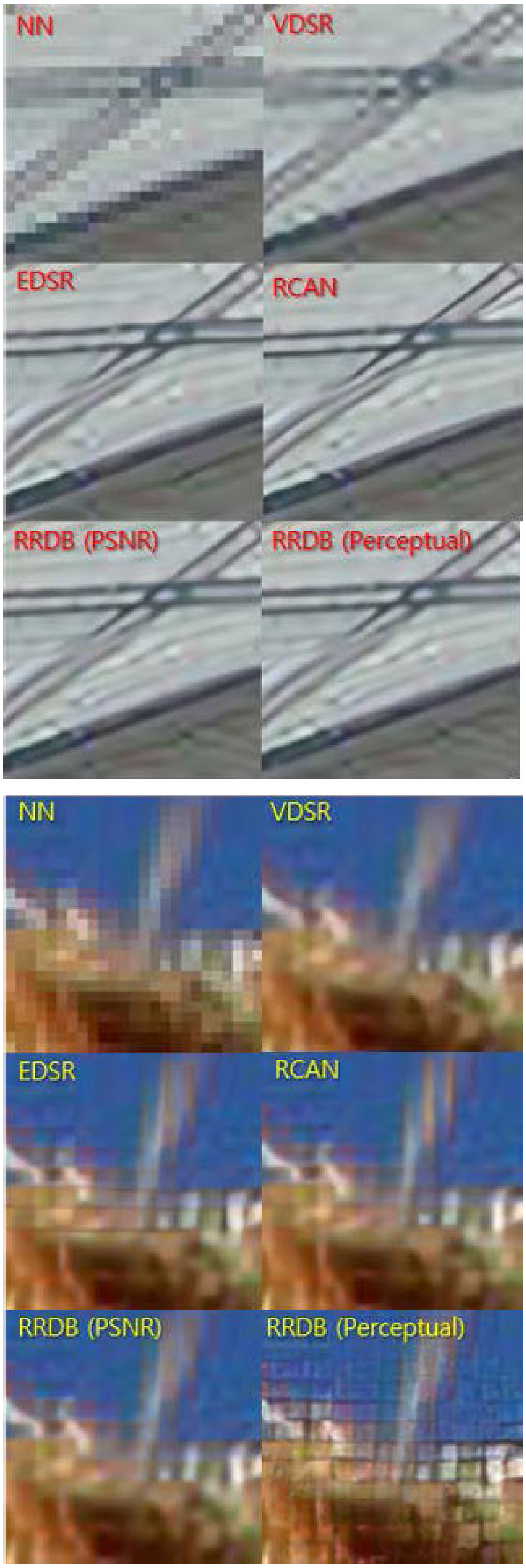

FIGURE 15. Enlargement of original images without reduction (NN: nearest neighbor interpolation).

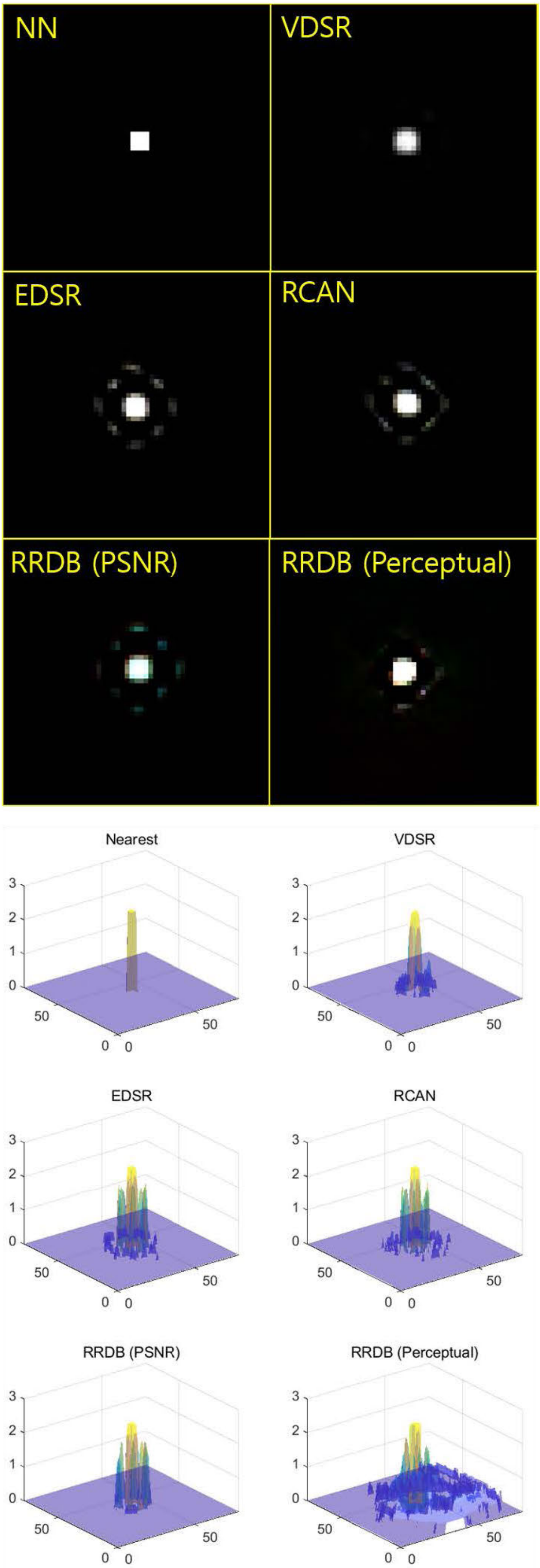

FIGURE 16. Response to the unit pulse (Green channel 3D graphs are in log scales.)

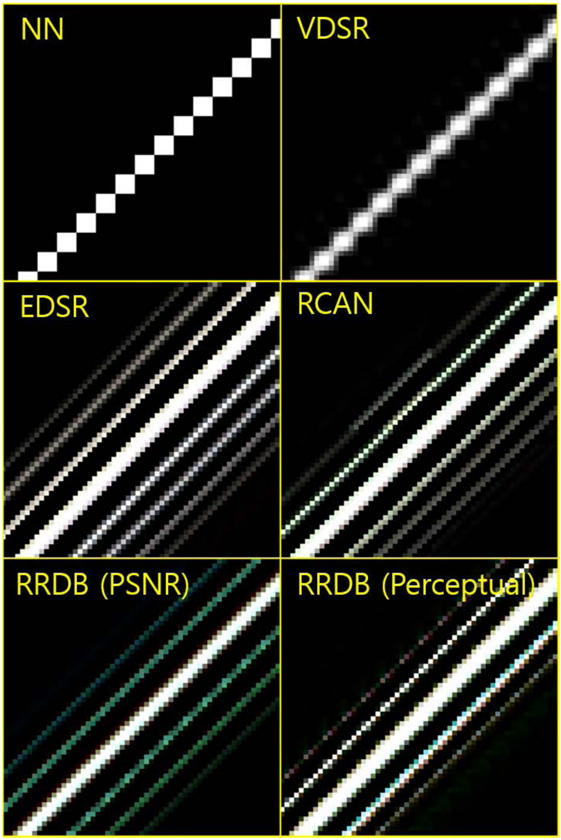

FIGURE 17. Response to the diagonal line.

slightly improved. These results indicate that DCN-SR methods may not always effectively use information from large receptive fields.

In traditional SR performance evaluation, a high-resolution image is reduced and a super-resolution method is applied. Finally, the PSNR between the original high-resolution image and the enlarged image is computed. However, this process has a serious problem. When an image is reduced, low-pass filtering effects are introduced, which may affect performance. So in the next experiments, we applied the DCN-SR methods to the original images without first reducing them. Fig. 15 shows the enlarged images along with the enlarged image when using nearest neighbor interpolation (NN). As can be seen, the EDSR, RCAN and RRDB produced annoying artefacts. The bilinear and bi-cubic interpolation methods produced no such artefacts, even though they produced blurred images (not shown). These results indicate that current DCN-SR methods may not use valid features to perform enlargement operations. In general, DCN-SR methods are not linear systems. Nevertheless, we investigated the DCN-SR responses to some basic patterns. First, we applied the DCN-SR methods to the unit pulse image (a single white dot at the center of an RGB image). Fig. 16 shows the unit pulse responses of the five DCN-SR methods along with the enlarged image using the nearest neighbor interpolation (NN). Except for the VDSR, the four methods showed somehow sinc-function-like artefacts, even though the side lobs were irregular. Next, we made a diagonal line and applied the five DCN-SR methods. Fig. 17 shows the results. Also in this example, except for the VDSR, the four methods (EDSR, RCAN, DBNN (PSNR), DBNN (perceptual)) showed somehow sinc-function-like artefacts. The four methods appeared to be non-symmetric in the three-color channels since they produced chromatic images for an achromatic input image (white line). In Fig. 18, we applied the five methods to a vertical bar image. Again, except for the VDSR, the four methods showed somehow sinc-function artefacts by having additional bar patterns. Furthermore, the four methods produced color artefacts. Finally, we applied the five methods to a diamond pattern (Fig. 19). The VDSR showed an expected output image, even though the eight dots became connected. The EDSR produced large artefacts (color distortions, false stripe patterns). Also, the EDSR showed different directional responses since the stripe patterns occurred in one diagonal direction. The RCAN produced some color artefacts with additional sinc-function-like artefacts. The RRDB (PSNR) showed similar artefacts, though the color artefacts were mostly green. The RRDB (perceptual) also showed large colorful artefacts. Also, it showed different directional responses since the output images were not symmetric even though the input image was symmetrical in the vertical and horizontal directions. All these results indicate that current DCN-SR methods may have some fundamental flaws.

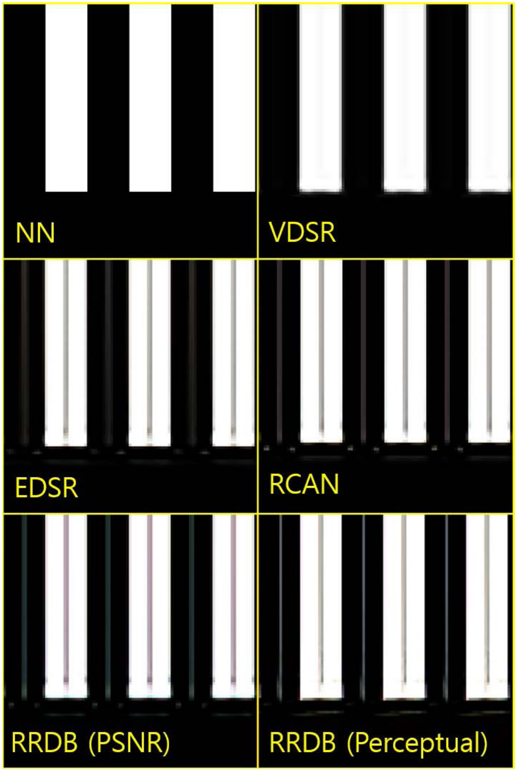

FIGURE 18. Response to the vertical bar.

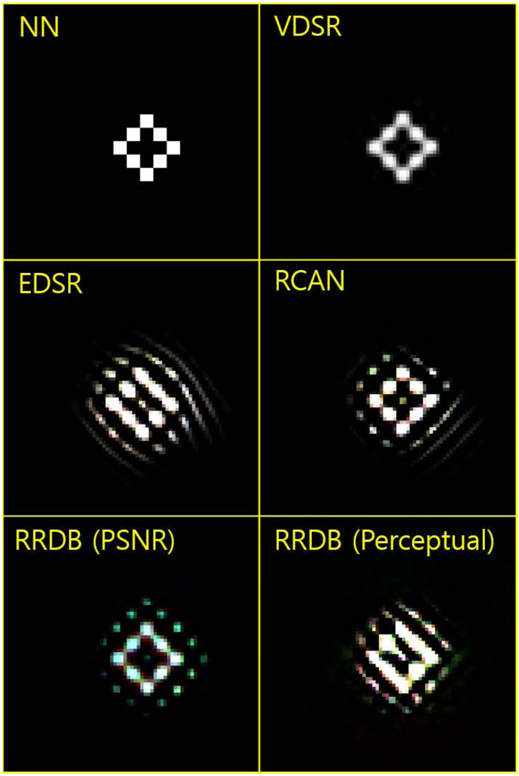

FIGURE 19. Response to the diamond pattern.

It appears that new DCN structures are needed to produce reliable super-resolution performance.

V. CONCLUSIONS

In this paper, we formulated the working mechanism of DCN-based super-resolution methods and showed that DCN-SR methods can be modelled as dynamic linear operations, which take an input image (patch) and bias terms. Based on the formulation, we analyzed the effective receptive fields of several DCN-based super-resolution methods, which were considerably smaller than theoretical receptive fields. These results indicate that significant complexity reduction may be possible without sacrificing performance. Based on the mathematical formulation, a series of experiments were conducted, which indicates that current DCN-SR methods may have some fundamental flaws and new types of DCN structures are needed to produce reliable super-resolution performance.

REFERENCES

[1] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-R. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, Apr. 2015.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[3] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, and X. Tang, "DeepID-net: Deformable deep convolutional neural networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2403–2412.

[4] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[5] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, Feb. 2018.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[8] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015, *arXiv:1506.06579*. [Online]. Available: http://arxiv.org/abs/1506.06579

[9] J. Koushik, "Understanding convolutional neural networks," 2016, *arXiv:1605.09081*. [Online]. Available: http://arxiv.org/abs/1605.09081

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*. [Online]. Available: http://arxiv.org/abs/1312.6199

[11] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.

[12] C. Godard, K. Matzen, and M. Uyttendaele, "Deep burst denoising," in *Proc. Comput. Vis. ECCV*, in Lecture Notes in Computer Science, 2018, pp. 538–554.

[13] J. Zhang, J. Shao, H. Luo, X. Zhang, B. Hui, Z. Chang, and R. Liang, "Learning a convolutional demosaicing network for microgrid polarimeter imagery," *Opt. Lett.*, vol. 43, no. 18, p. 4534, Sep. 2018.

[14] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[15] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

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

[17] 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," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 184–199.

[19] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3791–3799.

[20] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.

[21] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[22] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga, "Deep wavelet prediction for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 104–113.

[23] X. Jia, X. Xu, B. Cai, and K. Guo, "Single image super-resolution using multi-scale convolutional neural network," in *Proc. Adv. Multimedia Inf. Process. (PCM)*, in Lecture Notes in Computer Science, 2018, pp. 149–157.

[24] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11065–11074.

[25] S. Kong and C. Fowlkes, "Image reconstruction with predictive filter flow," 2018, *arXiv:1811.11482*. [Online]. Available: http://arxiv.org/abs/1811.11482

[26] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[27] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3867–3876.

[28] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1664–1673.

[29] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Comput. Vis. ECCV*, in Lecture Notes in Computer Science, 2018, pp. 256–272.

[30] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 864–873.

[31] J. C. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: A general deep learning framework for inverse problems," *SIAM J. Imag. Sci.*, vol. 11, no. 2, pp. 991–1048, Jan. 2018.

C. LEE (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1992. From 1986 to 1987, he was a Researcher with the Acoustics Laboratory, Technical University of Denmark (DTH). From 1993 to 1996, he worked with the National Institutes of Health, Bethesda, MD, USA. In 1996, he joined the Faculty of the Department of Electrical and Computer Engineering, Yonsei University, Seoul, South Korea. His research interests include image/signal processing, pattern recognition, and neural networks.

J. PARK received the B.S. degree in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2018, where he is currently pursuing the M.S. degree. His research interests include DNN-based video quality measurements and super-resolution.

S. WOO received the B.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2014 and 2020, respectively. He is currently working with LG Electronics. His research interests include medical imaging, pattern recognition, and deep learning.

J. KIM received the B.S. degree in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2019. He is currently pursuing the M.S. degree in electrical and electronics engineering with Yonsei University, Seoul. His research interests include DNN-based super-resolution and dependable deep learning.

J. YOON received the B.S. degree in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include DNN-based signal processing and pattern recognition.