International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.2 (Mar-2020)
http://dx.doi.org/10.12785/ijcds/090207

The Atrous CNN Method with Shorter Computation Time for Super-Resolution

Young-Man Kwon [1], Seung-Hyeok Jang [2], Dong-Keun Chung[3*], Won-Mo Gal[4*]

[1,2,3] *Department of Medical IT, College of Healthy Industry, Eulji University, Korea*
[4] *Department of Environmental Health and Safety, College of Healthy Industry, Eulji University, Korea*
**: corresponding author*

Received 27 Sep. 2019, Revised 25 Feb. 2020, Accepted 26 Feb. 2020, Published 01 Mar. 2020

Abstract: In this paper, we proposed the atrous CNN method with the shorter computation time while maintaining a high quality of image compatible to the VDSR method. We found that the computation was faster than VDSR through making experiments 30 times. To verify whether image quality for the proposed method is significant statistically or not, we evaluated the quality of the reconstruction image for 100 images. Through the ANOVA analysis, we found that there was no significant difference between methods in the view of the PSNR value and was a significant difference between methods in the view of the SSIM value. As the result of post-hoc analysis, there were two groups; one was the proposed method and the VDSR method. The other was the SRCNN method. In conclusion, the proposed method met our goal of maintaining compatible image quality and reducing computation time compared to VDSR method.

Keywords: Single Image Super Resolution; Convolution Neural Network; Atrous Convolution; VDSR; SRCNN; PSNR; SSIM

1. INTRODUCTION

In the area of single image super-resolution (SISR), it is to reconstruct high-resolution images from low-resolution images. In this area, various methods using deep learning have recently been proposed [1]. SISR is used in biometrics, including medical diagnostics, image compression, text enhancement, resolution enhancement of fingerprint and iris images. For this reason, it has traditionally been very important in the field of computer vision [2].

There are three kinds of approaches in SISR technology [3]. First, Interpolation, such as bicubic interpolation and Lanczos resampling, is famous. These are faster, simpler, but lower performance than other methods. Second, Reconstruction is based on prior knowledge. It uses a lot of time to create images and decreases performance as scale factor increases. Third, the learning-based methods, such as Neighbor embedding, sparse coding, and Deep Learning, have been proposed.

This paper consists of five chapters in total. In Chapter 2, we describe related works. Those are SRCNN and VDSR those are related to the super-resolution, the atrous convolution with large receptive field and the metrics of performance. In Chapter 3, we propose the enhanced atrous CNN method that maintains the quality of image and has the short computation time. It is based on VDSR but uses atrous convolution instead of general convolution. In Chapter 4, experimental results and evaluation of the proposed method are shown. In Chapter 5, conclusions are drawn from the above experimental results.

2. RELATED WORKS

A. Exiting methods in SISR

SISR is to make a high-quality image from a low-quality image [2]. Super-Resolution Convolutional Neural Network (SRCNN) is the first deep learning applied to a SISR problem [4]. SRCNN is the first end-to-end method to process all steps in one integrated framework. The overall procedure for the SRCNN is shown in Fig. 1.

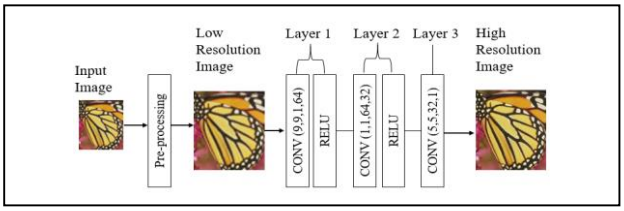

Figure 1. Overall procedure for the SRCNN method

E-mail: ymkwon@eulji.ac.kr, wkdtmd1202@naver.com, tchung@eulji.ac.kr, wonkal@eulji.ac.kr

In the SRCNN, given a single low-resolution image, they first upscale it to the desired size using bicubic interpolation, which is the only pre-processing [4]. The CNN part of SRCNN is composed of three layers. The first layer extracts the feature for each patch in low-resolution images by using (9,9,1,64) convolution filters. The operation of convolution layer is represented by CONV (width, height, the number of channels of input data, the number of filters) in Fig .1**.** The second layer performs nonlinear mapping to other multidimensional vectors images by using (1,1,64,32) convolution filters (This interpretation is only valid for 1×1). The third layer is the restoration step to produce the final high-resolution image by using (5,5,32,1) convolution filters.

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left\|F(Y_{i;}\theta) - X_i\right\|^2 \qquad (1)$$

SRCNN uses loss function as Mean Squared Error (MSE) like (1). It is defined as the average of the sum of squared differences between the predicted and actual target values of the model.

The bigger the filter size and the deeper layer, the better the performance, but it takes a lot of training time. SRCNN outputs good quality images in conventional SISR problems. However, this has a problem that only reflects a very narrow context. Since the preprocessing process is required for the image, this has also a problem that works only on a single scale. It means that a model has to be trained again if a new scale is on demand. In addition, the learning speed is slow due to a large amount of computation.

To solve the limitation of SRCNN, Very Deep Super-Resolution (VDSR) has been proposed. VDSR uses deep CNNs inspired by the vgg-net used in the image-net classification [5]. The overall procedure for the VDSR is shown in Fig. 2.

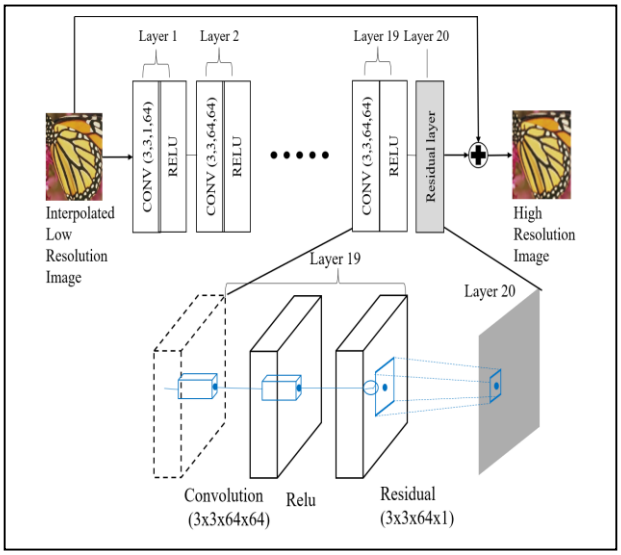

Figure 2. Overall procedure for the VDSR

VDSR uses 20 layers with the 3x3 convolution operation to get rid of the limitation of output image quality due to small receptive field, which is a disadvantage of SRCNN. Except for the input and last layer, 64 channels are used to compute 3x3 spatial regions. The input layer means the input image and the last layer is used to reconstruct the image. The performance is improved by learning from the whole point of view because of expanding the receptive field through the deeper layer. VDSR solves the slow learning problem by applying a learning rate of 10,000 times higher than SRCNN, which is the problem that SRCNN is slow in learning. VDSR uses loss function as MSE similar to SRCNN.

Especially, there are two key ideas of VDSR to learn the deep networks well [5]. The first is residual learning, which adds the input image to the network-created image just before the final high-resolution image. This makes the problem much simpler than the task of creating a complete image by making only the details of a high-resolution image compared to the original low-resolution image. Second, gradient clipping is used because the learning rate is high, and the network is at risk of diverging. This is the method to limit gradient values if they are above a certain range.

The VDSR method has 20 layers with the 3x3 convolution so that it has a large 41 × 41 receptive field. Therefore, since it has a large number of parameters, it requires a large amount of computation. However, because this method uses the low-resolution and high-resolution image as training data, there is no need to train again on the demand of different scale.

B. Atrous Convolution

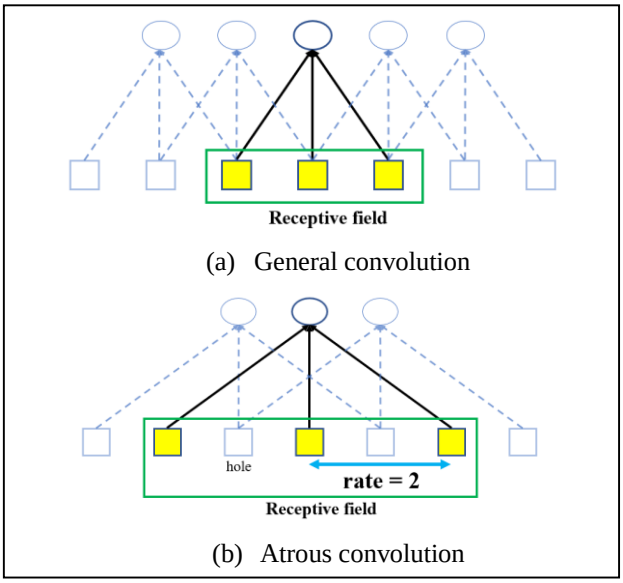

Figure 3. Receptive field of general and atrous convolution

The receptive field can be viewed as a single viewing area. For example, in the case of one dimension, the

receptive field of 3x3 general convolution is shown in Fig. 3 (a). The receptive field is 3. If the receptive field is high, it is good to capture the overall characteristics of a picture through a filter. If we increase the size of the filter to increase the receptive field, the number of parameters increases, and the amount of computation increases.

The atrous Convolution is a method to increase the receptive field forcefully by adding holes inside the filter [6]. Therefore, using the atrous convolution, the receptive field becomes large, but the number of parameters does not increase, so it is possible to obtain a better effect from the viewpoint of the computation amount. The atrous rate defines the interval within kernel [7]. For example, a 3x3 kernel with an atrous rate of 2 has the same field of view as the 5x5 kernel. For one dimension, the receptive field of this case is shown in Fig. 3 (b).

C. Evaluation metrics

We need evaluation metrics to evaluate the quality of the resulting image. There are many metrics in this area. We will use the Peak Signal-to-Noise Ratio (PNSR) and Structural SIMilarity (SSIM) as metrics.

The PSNR is mainly used to evaluate image quality information in image or video lossy compression. The PSNR indicates the ratio of the noise to the maximum signal that the signal can have [8]. It can be calculated using the MSE without considering the power of the signal. The MSE simply evaluates the image quality as a numerical difference between the original image and the distorted image. The formula of PSNR is shown in (2).

$$\mathrm{PNSR} = 10\log\frac{s^2}{MSE} \tag{2}$$

Where, the s is the maximum value of the image. In the case of an 8-bit image, the value of s is 255. The smaller the MSE, the larger the PSNR since the denominator in (2) is the MSE. For lossless images, the PSNR is not defined because the MSE is zero. Since PSNR is measured at the logarithmic scale, [db.] is used as a unit, and the lower the loss, the higher the value. This often results in quality scores that are not consistent with what people feel.

Since the PSNR does not reflect perceptual quality accurately, the SSIM method has been proposed to overcome these limitations. The SSIM is designed to improve existing methods such as the PSNR and MSE. It is a method to measure the similarity to the original image concerning the distortion caused by compression and conversion and compares more precisely than the MSE and PSNR methods. The SSIM is a method of predicting the recognition quality of digital television and film images as well as other types of digital images and video. The SSIM uses luminance, contrast, and structure those are recognized as main contents in human vision. Its formula between images x and y is given by (3).

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(2\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \tag{3}$$

Where, the μ_x and μ_y are means, the σ_x^2 and σ_y^2 are standard derivations, the σ_{xy} is covariance value, and the c is constant. Based on these values, luminance, contrast, and structure are calculated and the value of SSIM is calculated using them. The SSIM value closer to 1 is closer to the original image and the SSIM value closer to 0 is more different from the original image.

3. PROPOSED METHOD

The VDSR method shows better performance metrics compared to the SRCNN. However, it needs more training time because of more parameters. Therefore, we want to develop the method with short computation time while maintaining the same quality of image as VDSR method. Therefore, we propose the new method using the atrous CNN method to achieve this goal. It is shown in Fig. 4.

We replacement the convolution layer with the atrous convolution layer to make the receptive field larger using the same filter size as VDSR. We use a residual network at layer 10 (VDSR is composed of 20 layers). We use the activation function as Rectified Linear Unit (Relu) from

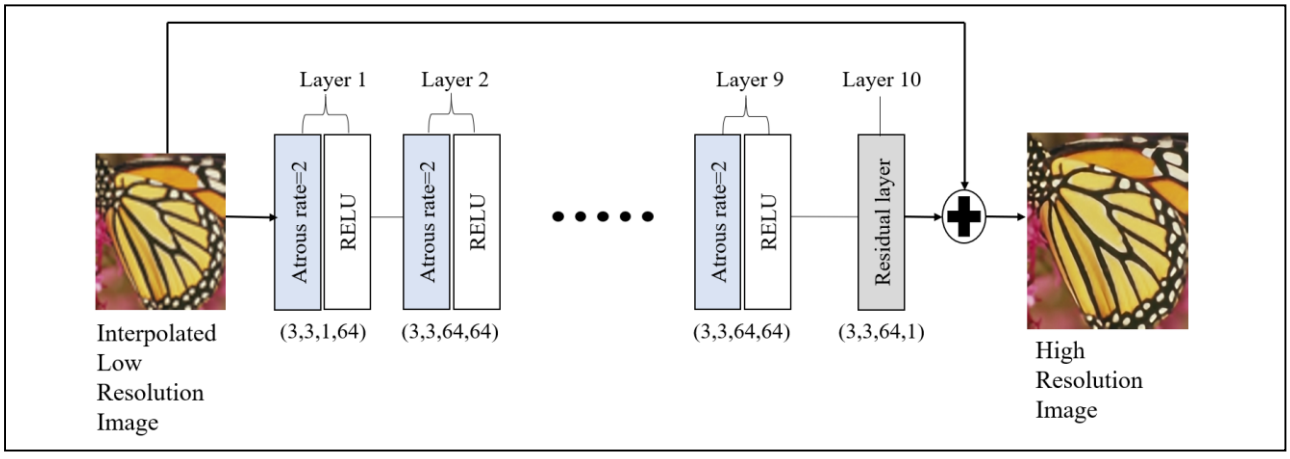

Figure 4. Overall procedure of the proposed method with atrous convolution

layer 1 to 9 just after each atrous convolution layer. In this paper, we use the 3x3 atrous convolution filter with atrous rate=2, stride=1. If we use those parameters, processing the atrous convolution looks like Fig. 5.

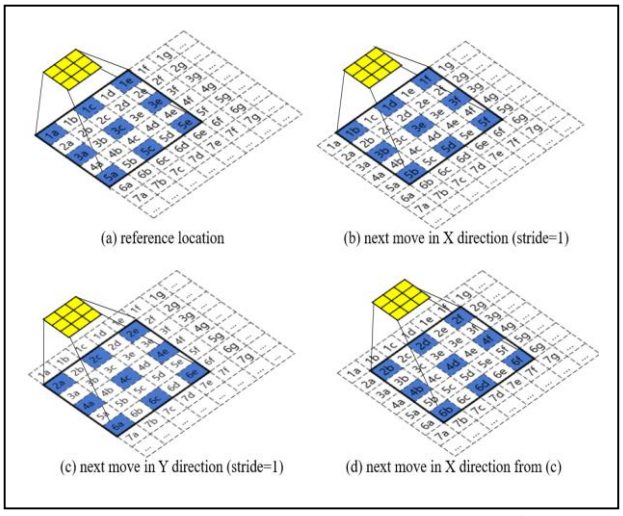

Figure 5. Processing the atrous convolution by using 3x3 filter with atrous rate=2, stride=1

In Fig. 5, the (b) and (c) show a one-step move in X and Y directions to the reference location (a), respectively. The (d) indicates a one-step move in the X direction to the (c). As you can see, the receptive field of the 3x3 atrous convolution with atrous rate=2 is 5x5. Because of this effect, we can reduced the number of atrous convolution layers to 10. Therefore, we can reduce the computation time because of a few parameters for the same receptive field.

4. EXPERIMENT RESULTS

A. *Experimental dataset and configuration*

We made experiments by using the same images that were used for training and performance measurements in the VDSR method. Training data are 91 images of the Yang et al, and the data for measuring the performance are 'Set5', 'Set14', 'B100'.

In the VDSR method, the data preparation is similar to SRCNN with some differences. The input patch size is equal to the size of the receptive field and images are divided into sub-images with no overlap. A mini-batch consists of 64 sub-images, where sub-images from different scales can be in the same batch. The neural network is trained with multiple scale factors of 2, 3, 4 [4]. We used the same procedure as the VDSR method for the data preparation.

We implemented the proposed method by using Pytorch. We used the existing VDSR method that was implemented by Pytorch [9]. We also used the existing SRCNN that was implemented by Tensorflow.

Initial parameters of the proposed method were set to the same as the VDSR. The initial learning rate of the proposed method was set to 0.1. Momentum and weight decay were set to 0.9 and 1e-4, respectively. The patch size was set to 41x41, which was the input image. We had run 50 epochs and reduced the learning rate by 1/10 every 20 epochs.

In the SRCNN, the initial learning rate was 1e-4. The patch size was set to 33x33, which was the input image. The size of batch was set to 128. We run 100 epochs

B. *Experimental Results and Analysis*

We made experiments by using a machine (CPU: i7-6700K, Memory: 64GB, GPU: Nvidia GTX 1080TI x 2EA). We trained the proposed method 30 times, considering 50 epochs as one run (1260 iterations for each epoch). We had measured the mean of training time and shown them in table 1.

TABLE I. MEAN OF TRAINING TIME

The proposed method	The VDSR method
181 (min.)	253 (min.)

As we can see from Table 1, the computation time of the proposed method compared to VDSR is faster than about 1.4 times (253/181).

To see the process of speed up, we measured the time of iteration and calculated the average speed per each epoch during each training. Since each epoch has the number of 1260 iteration and each run has 50 epochs, the number of iterations per run is 63,000 for training. In the one case of 30 runs, the mean of training time per iteration is shown in the Table 2. In addition, it is shown graphically in Fig. 6.

TABLE II. MEAN OF TRAINING TIME PER ITERATION

The proposed method	The VDSR method
0.1346 (sec.)	0.2410 (sec.)

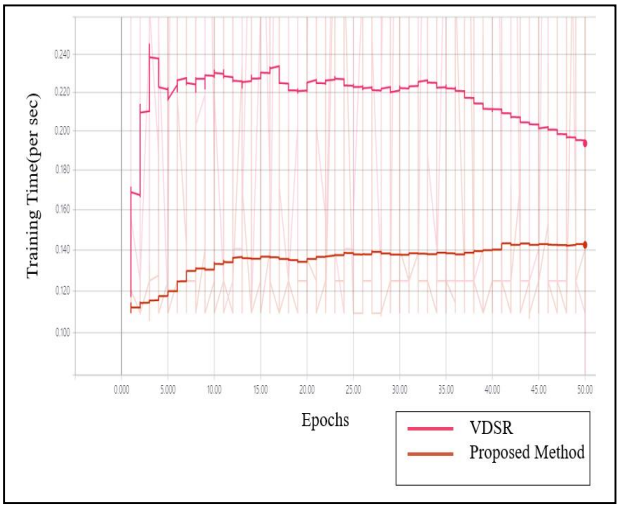

Figure 6. Time taken to train per epoch

TABLE III. THE MEAN VALUES OF PSNR AND SSIM ON TEST DATASET (SET5, SET14, AND B100).

Data Set	Scale	The proposed method		The VDSR method		The SRCNN method	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	x2	**37.09**	**0.9405**	37.30	0.9433	36.34	0.9521
	x3	**33.51**	**0.9103**	33.54	0.9191	32.39	0.9025
	x4	**31.24**	**0.8833**	31.25	0.8972	30.08	0.8721
Set14	x2	**32.67**	**0.9082**	32.83	0.9095	32.18	0.9038
	x3	**29.61**	**0.8520**	29.69	0.8519	29.00	0.8148
	x4	**27.87**	**0.7991**	27.90	0.7995	27.20	0.7417
B100	x2	**31.58**	**0.9075**	31.70	0.9105	31.14	0.8850
	x3	**28.67**	**0.8347**	28.70	0.8370	28.21	0.7807
	x4	**27.15**	**0.7781**	27.17	0.7791	26.71	0.7035

Table 3 shows the mean values of PSNR and SSIM for the proposed method, the VDSR, and the SRCNN on each test dataset. The mean values of the proposed method are higher than those of the SRCNN and have similar mean values with the VDSR.

For reconstructed images of three methods to have difference statistically, we selected B100 dataset with x2, because the difference of mean value between the proposed method and the VDSR method was the largest (0.44db, 0.0225), the difference between the proposed method and the SRCNN method was the smallest (0.12db, 0.003). At first, we made the boxplot of them in Fig. 7.

Second, we made the analysis of variance (ANOVA). As the result of it, it was not significant for PSNR (p-value = 0.5870). However, there is a significant difference for SSIM (p-value = 0.0011 < 0.05). Therefore, we did post-hoc-analysis for SSIM. The difference between the proposed method and the SRCNN was significant (p = 0.0014 < 0.05). However, the difference between the proposed method and the VDSR was not significant (p = 0.6497 > 0.05). In conclusion, from the viewpoint of PNSR, there was no difference between methods. However, in the viewpoint of SSIM, there were two groups; one was the proposed method and the VDSR, the other was the SRCNN.

Fig. 8 shown the result images of each method for one of Set5 with scale x2. We can find dissimilarity between the proposed method and the SRCNN, the VDSR and the SRCNN in the result images. However, we cannot find dissimilarity between the proposed method and the VDSR in the result images.

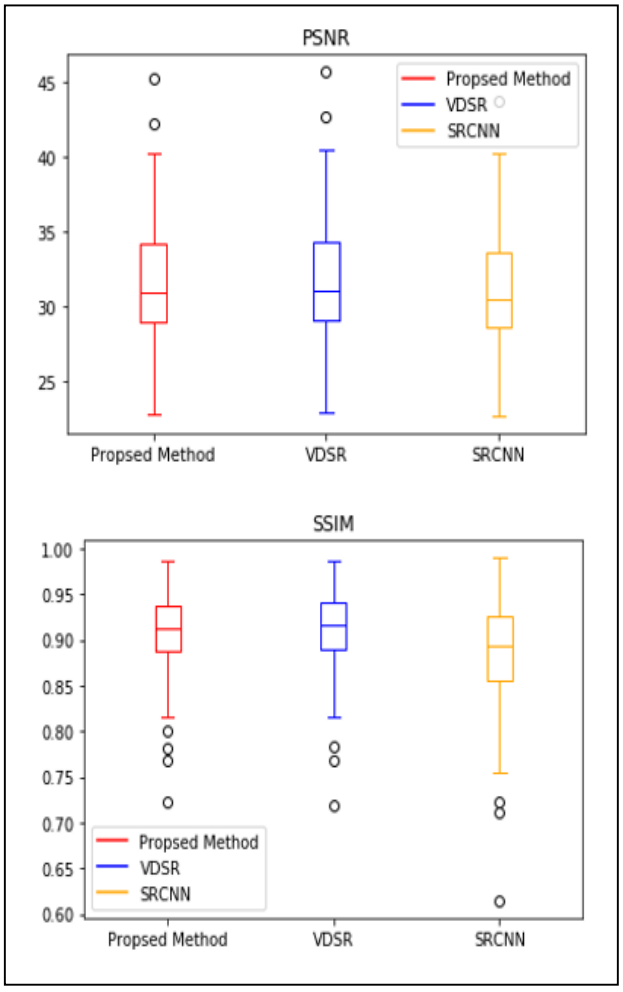

Figure 7. The boxplot of PNSR and SSIM values about scale 2 of B100

CONCLUSION

The existing VDSR produces high-quality images but the learning speed is slow because of the large amount of computation. We reduced the number of layers by applying the atrous convolution to reduce the amount of computation in the VDSR structure. We confirmed that our proposed method was faster 1.4 times than the VDSR by experiments.

To prove the image quality of the proposed method, we compared the resulting image of the proposed method with the VDSR and the SRCNN by using the metrics of PSNR and SSIM. In the view of SSIM, we concluded the difference between the three methods was significant by using the ANOVA and post-hoc analysis. The difference between the proposed method and the SRCNN was significant. However, the difference between the proposed method and the VDSR was not significant. Therefore, we concluded that the proposed method was the compatible image quality to the VDSR method.

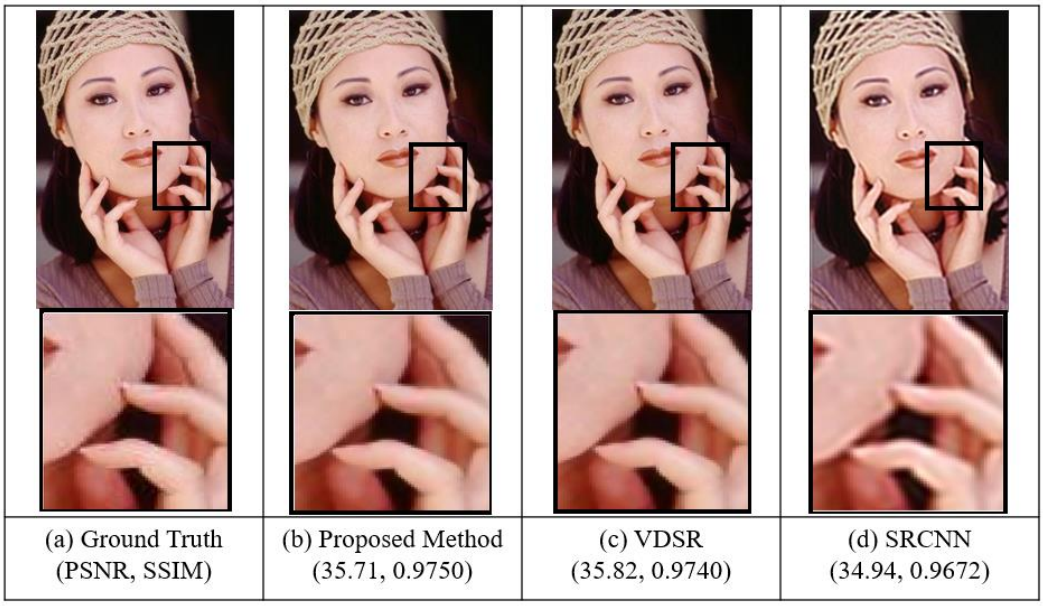

Figure 8. Super-resolution result of scale factor x2, woman (Set5)

ACKNOWLEDGMENT

This work was supported by R.O.K. National Research Foundation under grant NRF-2017R1D1A1B03036372 in 2019.

REFERENCES

[1] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, ‘‘Deep laplacian pyramid networks for fast and accurate super-resolution,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 624–632.

[2] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 349–356.

[3] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, ‘‘Deep learning for single image super-resolution: A brief review,’’ IEEE Trans. Multimedia, to be published.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2015.

[5] 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.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, ‘‘Encoder-decoder with atrous separable convolution for semantic image segmentation,’’ in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 821–833.

[7] E. Strubell, P. Verga, D. Belanger, and A. McCallum, ‘‘Fast and accurate entity recognition with iterated dilated convolutions,’’ in Proc. Conf. Empirical Methods Natural Lang. Process., Copenhagen, Denmark, 2017, pp. 2670–2680.

[8] Y. A. Y. Ai-Najjar and D. C. Soong, ‘‘Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI,’’ Int. J. Sci. Eng. Res., vol. 3, no. 8, pp. 1–5, Aug. 2012.

[9] A Pytroch Implementation of “Accurate Image Super-resolution Using Very Deep Convolution Networks,” http://github.com/twtygqyy/pytorch-vdsr.

Young-Man Kwon 1985.2 M.S. in Department of Electric and Electronic engineering, KAIST, Korea.
1998.3 Complete a Doctorate, in Department of information and communication engineering, KAIST, Korea.
2007.2 Ph.D. in Department of Electronic engineering, Kwangwoon university, Korea
1993.3 ~ Professor in Eulji university, Korea.

Seung-Hyeok Jang
2015.3 ~ Junior in Department of Medical IT, Eulji university, Korea.

Dong-Keun Chung
1996.6 Ph.D. in Department of Electronics Engineering, SungKyunKwan university, Korea.
1990.3 ~ Professor in Eulji university, Korea.

Won-Mo Gal
1998.2 Ph.D. in Department of Industry Engineering, Ajou university, Korea.
1990.3 ~ Professor in Eulji university, Korea.