



# Low-complexity CNN with 1D and 2D filters for super-resolution

Jangsoo Park<sup>1</sup> · Jongseok Lee<sup>1</sup> · Donggyu Sim<sup>1</sup>

Received: 4 April 2019 / Accepted: 8 September 2020 / Published online: 21 September 2020  
© The Author(s) 2020

## Abstract

This paper proposes a low-complexity convolutional neural network (CNN) for super-resolution (SR). The proposed deep-learning model for SR has two layers to deal with horizontal, vertical, and diagonal visual information. The front-end layer extracts the horizontal and vertical high-frequency signals using a CNN with one-dimensional (1D) filters. In the high-resolution image-restoration layer, the high-frequency signals in the diagonal directions are processed by additional two-dimensional (2D) filters. The proposed model consists of 1D and 2D filters, and as a result, we can reduce the computational complexity of the existing SR algorithms, with negligible visual loss. The computational complexity of the proposed algorithm is 71.37%, 61.82%, and 50.78% lower in CPU, TPU, and GPU than the very-deep SR (VDSR) algorithm, with a peak signal-to-noise ratio loss of 0.49 dB.

**Keywords** Super-resolution · CNN · VDSR · Deep learning

## 1 Introduction

With the recent evolution of computing and display devices, the demand for high-resolution video services is increasing continuously. However, much computing capability and resources are required for high-resolution devices and media services. Owing to the limited resources and a large number of applications, super-resolution (SR) algorithms have been studied for a long time [1–12]. SR algorithms can be employed for image and video analysis of poor-quality inputs. Also, the SR algorithms can be utilized in many commercial applications such as televisions and displays because they can provide better visual quality for low-resolution videos with limited bandwidths. The SR technology recovers a single high-resolution image from one or more low-resolution images. However, the SR technology is known to be ill-posed because it should compute more unknown pixel values than those present originally in an input image. An SR technology that is ideal for all known applications has not been developed so far. Although many algorithms based on conventional signal processing have been studied, they have some limitations when applied in practical applications.

To solve this problem, multi-frame SR approaches have been proposed [2–5]. These approaches can alleviate ill-posed issues with multiple observations; however, they have high computational complexity and require multiple frames to be aligned. Furthermore, multiple frames for inputs of SR algorithms may not be available for many general applications. For single- and multiple-frame SR, deep-learning-based solutions have been studied along with the conventional signal-processing algorithms [13–17].

Convolutional neural networks (CNNs) are widely used for many image processing and computer vision problems, and their variations produce reasonably good performance when compared to the existing signal-processing algorithms with high computational complexity. The super-resolution convolutional neural network (SRCNN) [16] is the first technique that applies a CNN model to SR applications. The SRCNN has three layers to learn the relationship between low-resolution images and the corresponding high-resolution images, directly. It yields better visual quality than conventional methods without any deep-learning models. However, SRCNN does not produce excellent visual quality because it employs a limited number of layers and has a narrow structure. Very deep super-resolution (VDSR) [17] using a deeper CNN consisting of 20 layers has been proposed. It exhibits a visual quality better than that of the SR technologies studied. VDSR learns the difference between low-resolution images and the corresponding high-resolution images using residual

✉ Donggyu Sim  
dgsim@kw.ac.kr

<sup>1</sup> Department of Computer Engineering, Kwangwoon University, Seoul 139701, Korea

learning. In the high-resolution image-reconstruction step, the final high-resolution image is reconstructed by combining the low-resolution input image and its high-frequency components. Although VDSR yields good visual quality by using a deep-layer network, it has high computational complexity and requires a large amount of memory because of the number of weights involved. In general, deep-learning models with deeper and wider structures show good performance; however, they have high complexity because of using a large number of multiplications and the presence of multiple stages from many layers. Most deep-learning algorithms can be implemented on graphics processing units (GPUs), which can process inputs in a shorter time even if they have complicated structures. However, it is difficult to employ GPUs in some environments with limited resources, such as mobile devices, closed-circuit televisions (CCTVs), and so on. In this paper, we propose a low-complexity SR algorithm that maintains the visual quality of the VDSR.

Recently, low complexity and simple design of deep-networks become more important for practical applications. To reduce the computational complexity of the deep networks, many works have tried to reduce the number of parameters and a smaller number of network nodes. FSRCNN [18] does not need to up-sample low-resolution input images, but the “deconvolution” layer in the middle of the network is embedded instead of the input up-sampling process. ESPCN [19] proposes the “sub-pixel convolution layer” that increases the number of channels at the output layer in proportion to the up-sampling ratio and predict the output tensors of the output layer. Shamsolmoali et al. [20, 21] proposed a dilated dense convolution network to upsample and, applied dense network and residual dense network to SR algorithm to enhance high-frequency component. G-GANISR [22] is gradually trained in terms of scale using GAN for high quality SR images. However, these algorithms support only integer ratio SR. However, a fractional SR ratio, for example, 1.5 and 2.5, is also inevitable for practical applications. Furthermore, the integer ratio SR also has the disadvantage to maintain aspect ratios in the horizontal and vertical directions. On the other hand, the proposed method can support not only the integer ratio but also the fractional ratio SR. Also, conventional SR methods attempt to predict the high-frequency components that are lost in low-resolution images. However, since they are based on a process of generating new frequencies, many parameters should be introduced for mapping real high-frequency components in original images. In the proposed method, the imaginary high-frequency components in the vertical and horizontal directions can be generated with the 1st 1D network and mapped to the original high-frequency ones with the consecutive 2-D networks. The simple 2D networks can handle the rest of the high-frequency components that cannot be handled by the front-end side 1D networks. The high-resolution image-reconstruction

layers extract the high-frequency signals in the diagonal directions and reconstruct the final high-resolution image by combining the various high-frequency signals. Also, the proposed network can provide additional functionality to generate multi-level super-resolution images from internal nodes as well as the final nodes. The fine super-resolution image can be obtained from the final output nodes while the coarse super-resolution image is also extracted from the internal front-end 1D network. The proposed algorithm is compared with several conventional algorithms such as bicubic, SRCNN, and VDSR. We found that the proposed algorithm can yield visual quality comparable to those of VDSR with significantly lower computational complexity that is 71.37% less than that of the VDSR.

The remainder of this paper is organized as follows. In Sect. 2, the existing SR algorithms are discussed in brief. In Sect. 3, we describe the proposed low-complexity SR algorithm with 1D and 2D convolutional filters. In Sect. 4, the performance of the proposed algorithm is evaluated by comparing it with the existing methods in terms of visual quality and speed-up performance. Section 5 concludes the paper.

## 2 Conventional super resolution

SR technology aims to restore high-resolution images with additional high-frequency information from low-resolution images. However, the SR algorithms should be designed and developed according to applications because different applications have different requirements such as low complexity, high visual quality, low power, less memory, and so on. Based on the requirements and technology evolution, several SR algorithms have been developed, ranging from adaptive signal-processing algorithms to deep-learning algorithms. In this section, optimization-based SR technology, based on signal processing, is introduced. Then, SRCNN is presented briefly, as an example of low-complexity CNN-based SR algorithms. The VDSR, which is known as an efficient SR algorithm with a deep and wide network structure for better visual quality, is also discussed.

### 2.1 Problem statement of SR and optimization-based algorithms

SR technologies can be classified into multiple-image SR and single-image SR. The relationship between low-resolution and high-resolution images can be denoted by

$$y_k = H_k X + n_k, \quad (1)$$

where  $\mathbf{X}$  and  $y_k$  are high-resolution and low-resolution images, respectively.  $H_k$  is a down-sampling and distortion model and  $n_k$  is a noise model. We can state that the SR

algorithms reconstruct  $\mathbf{X}$  from  $y_k$  with unknown  $H_k$  and  $n_k$ . For single-image SR algorithms,  $H_k$  is under-deterministic and its rank is smaller than that of  $\mathbf{X}$ . Thus, SR algorithms with a single image have at least one more assumption. On the other hand, multiple-image SR algorithms restore a high-resolution image from multiple images acquired at various attitudes and different times, from the same scene. For the multiple-image SR,  $H_k$  includes various factors such as shift, rotation, blur, down-sampling, and other distortions. Depending on the application and acquisition environment,  $H_k$  could be a full-ranked matrix. However,  $H_k$  is not likely to be full-ranked in many cases. Even the distortion matrix could be full-ranked, but its parameters are not known. Thus, the multiple-image SR algorithms also require several assumptions. Even with the assumptions, the multiple-image SR algorithms involve much computational load such as motion estimation and warping for alignment. Also, they require several frame buffers and latencies.

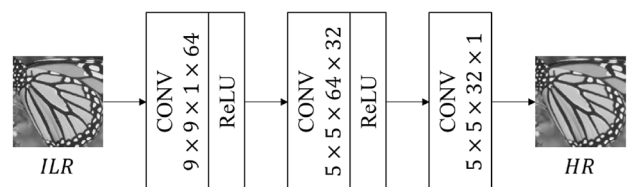
As mentioned before, single-image SR technology restores a high-resolution image from only one image. Single-image SR technology can be classified into external-example-based methods [6–8], which utilize prior assumptions, and internal-example-based methods, which use the information inside the image [9–12]. A dictionary-based SR technology, which is a typical external reference method, is composed of an optimization step and a high-resolution image-restoration step. It uses the relationship between a given low-resolution image and its corresponding high-resolution image. In the optimization step, a low-resolution image and its high-resolution image are divided into patches to generate a feature dictionary for reconstructing the high-resolution image. In the high-resolution image-reconstruction step, a low-resolution patch most similar to the input image patch is selected from the high-resolution feature dictionary generated in the learning step, and then, the high-resolution image is restored using the corresponding high-resolution features. The dictionary-based SR technology uses only a single image and the external information constructed previously to restore a high-resolution image with minimal complexity, compared to that of the multiple-image SR technologies, and it has a reasonably good visual-quality performance. However, the visual quality of the dictionary-based SR algorithms depends on the amount of data learned previously. Therefore, higher computational complexity and memory complexity are required in the learning stage to restore a high-resolution image of better quality. On the other hand, in the case of internal-reference methods, the widely known self-similar reference-based SR technology reconstructs an optimal high-resolution image by extracting similar features from a pyramid structure after constructing a scale-based pyramid as a given input image. Although it has lower visual quality, compared to the other

high-complexity SR algorithms, it has an advantage that only a small amount of memory is required.

## 2.2 CNN-based super-resolution

As the utilization of big data has as useful information increased and is evolving, deep-learning frameworks have started applying many computer vision problems and several milestones have been achieved. Deep-learning frameworks learn a large amount of data and automatically extract feature spaces and integrated information during the learning stage. Among the deep-learning frameworks, CNNs are widely used for image analysis and interpretations [14–17]. Recently, CNN structures have been utilized for SR applications, and they achieve significant improvements in the visual quality of the reconstructed images, for some computational loads.

SRCNN [16] is widely known as the first deep-learning study on SR applications. SRCNN uses a CNN structure with three layers, as shown in Fig. 1. Unlike conventional CNNs, it does not perform pooling and it has a fully connected network layer for SR purposes. The input is the low-resolution image interpolated to the target size by a simple interpolation algorithm such as cubic interpolation. The SRCNN produces an output that is a high-resolution reconstructed image. The SRCNN structure can be interpreted as a sparse representation of SR technology [7], which is a dictionary-based SR method, which directly uses the relationship between the low-resolution input image and the corresponding high-resolution image. The first layer extracts feature information from low-resolution images and the second layer converts the low-level features into high-level features. In the last layer, the final high-resolution image is reconstructed by combining high-level features with proper weighting factors. The training images from the ImageNet large-scale visual recognition competition (ILSVRC) [13] 2013 data provided by ImageNet’s video recognition contest can be used, and learning can be performed using the stochastic gradient descent method. SRCNN can produce higher visual quality than conventional methods based on signal-processing algorithms. However, as deeper and wider



**Fig. 1** Block diagram of super-resolution convolutional neural network (SRCNN). The CONV,  $9 \times 9 \times 1 \times 64$  means that is the shape of a filter as width  $\times$  height  $\times$  input channels  $\times$  output channels in the first convolution layer

network structures can yield better performance, many other modifications have been developed, with higher computational loads. Note that the SRCNN cannot produce high-resolution information at the frame boundaries.

VDSR [17] yields a reasonably good visual quality when compared to the existing SR technologies based on deep-learning and signal processing. VDSR employs 20 layers for the CNN, as shown in Fig. 2. Similar to SRCNN, a low-resolution image interpolated to the target size is fed into the VDSR model. The final high-resolution image is reconstructed as the sum of the input low-resolution image and high-frequency signals. VDSR recovers the high-frequency components, instead of the original signals, during the learning stage. The training images used are 91 images extracted from Yang [7] and 200 images extracted from the Berkeley segmentation dataset set (BSDS) [23]; i.e., a total of 291 images are used. Similar to SRCNN, the stochastic descent method is employed for learning, by minimizing the mean square error (MSE) of the original high-resolution images and their reconstructed images. VDSR is based on learning the residual signals, to efficiently deal with high-frequency information. In the literature, the learning time could be accelerated by reducing the learning rate every 20 epochs. Also, the adaptive gradient clipping method is employed for stable learning.

Deep-learning algorithms are known to show good performance as they learn large amounts of data by using deeper and wider network structures. This general concept also works for SR applications; however, it leads to high computational loads. That is, deeper CNN-based SR algorithms can produce better visual quality for SR by using a large number of multiplications because of the large number of weights. Deep-learning algorithms can be implemented on fast platforms such as GPUs; however, the GPUs consume large amounts of power. High-performance GPUs cannot be adopted in many applications such as CCTVs, low-power smartphones, digital cameras, and other mobile devices that require low power and have limited memory. Thus, a low-complexity SR algorithm should be developed by maintaining a reasonably good visual quality.

### 3 Proposed 1D–2D CNN-based super resolution

For image SR, a high-resolution image can be represented by a sum of a low-frequency signal and high-frequency signals. In general, the low-frequency signal is included in the given input image and the high-frequency signals should be recovered based on prior assumptions, image formation models, and data. The SR algorithms based on deep-learning frameworks focus on data-associated nonlinear mapping with a set of parameters from a vast amount of real data. When the SR algorithm based on a deep-learning model is given, we can compute the model parameters,  $\hat{\theta}$ , which can be denoted by

$$\hat{\theta} = \arg \min D(\text{GT}, \text{HR}), \quad (2)$$

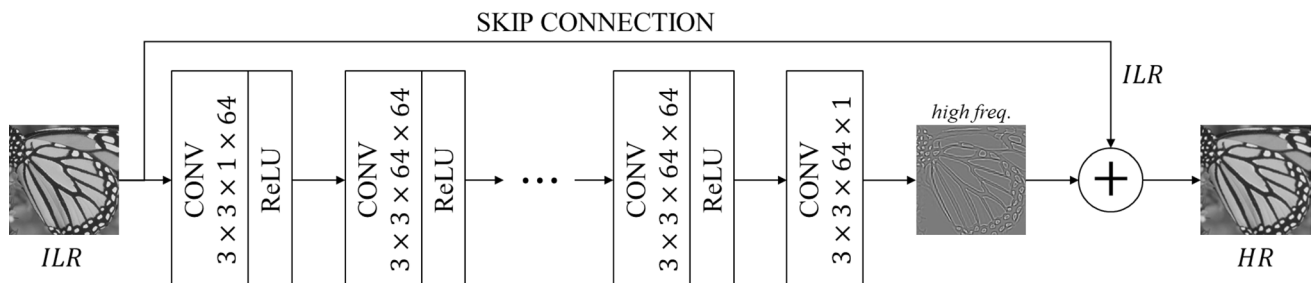
where GT represents the ground truth, which is a high-resolution original image, and HR represents its reconstructed high-resolution image.  $D(\cdot)$  is a cost function between the ground truth and reconstructed images. That is, in the learning process, the set of weights is computed by minimizing the cost between the high-resolution original image and the high-resolution image reconstructed using an optimization algorithm. In this paper, we employ a residual prediction model such as the VDSR and the reconstructed high-resolution image, HR is assumed to be computed by

$$\text{HR} = \text{ILR} + \text{Model}(\text{ILR}; \theta), \quad (3)$$

where ILR represents an interpolated low-resolution image that is obtained by the cubic interpolation method. Model (ILR;  $\theta$ ) is the deep-learning model to predict the high-frequency components with a set of parameters including the weight matrix. The proposed SR model is based on a CNN with 1D and 2D filters, to reduce the computational complexity with negligible PSNR loss.

#### 3.1 Proposed 1D–2D CNN model

In this paper, we classify the high-frequency signals for high-resolution image restoration into three types, depending



**Fig. 2** Block diagram of a very-deep super-resolution (VDSR). VDSR has 20 consecutive convolutional layers and one skip connection. SR is predicted as the sum of ILR and high frequency image

on the directionality. The proposed algorithm classifies high-frequency signals into horizontal, vertical, and diagonal directions. Figure 3 shows the low-complexity neural network architecture proposed in this paper. The proposed structure consists of two stages. In the first feature-extraction stage, the low-resolution image interpolated to the target size is fed to the system. Then, the high-frequency signals in the horizontal and vertical directions are reconstructed and combined with the low-resolution images interpolated to the target size, to generate high-level feature information. In the 1D convolutional layers, only the horizontal and vertical high-frequency signals are restored so that the high-level features can be different from the target high-resolution original image. In the back-end image-restoration stage, the high-frequency signal in the diagonal direction is extracted to reconstruct the final high-resolution image, together with the early feature information. The proposed structure reconstructs the horizontal and vertical high-frequency signals with a smaller number of computations than the VDSR network. In the high-resolution image-reconstruction layer, the second stage using 2D filters with four layers is designed to reconstruct the high-frequency signal components.

### 3.2 1D convolutional layers

The 1D convolutional layers consist of two types of 1D CNNs, for horizontal and vertical high-frequency signals, and the subsystem is expressed as

$$\widehat{HR} = ILR + Model_v(ILR; \theta_v) + Model_h(ILR; \theta_h), \tag{4}$$

where  $\widehat{HR}$  is the intermediate output image obtained by adding the high-frequency signals in the vertical and horizontal directions to the interpolated  $ILR$ . The horizontal high-frequency  $Model_h(ILR; \theta_h)$  is obtained using  $3 \times 1$  filters and

the  $Model_v(ILR; \theta_v)$  is obtained using vertical directional filters ( $1 \times 3$ ). Each structure has four layers, and the intermediate layers excluding the output layer employ rectified linear units (ReLU) as activation functions. The parameters  $(\theta_h, \theta_v)$  are computed by minimizing the cost function,  $L_{1D}$ , which is defined by

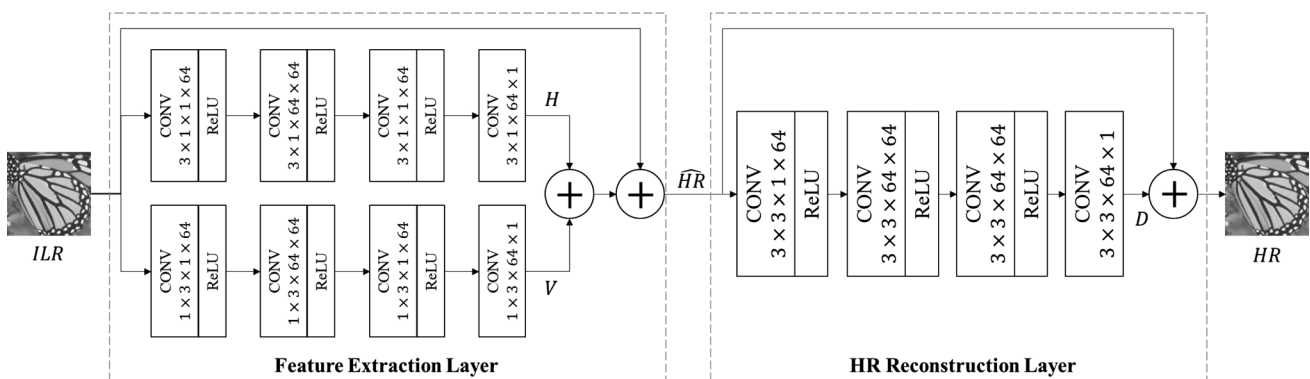
$$L_{1D} = \sum_{i=0}^n |\widehat{HR}_i - GT_i|_1^1. \tag{5}$$

The 1D CNN can be trained by minimizing the cost function, and the model consists of 1D filters; thus, the 1D CNN model can be trained with lower computational load, compared to the existing CNN-based SR algorithms. The 1D CNN model can handle most of the high-frequency signals, by using horizontal and vertical filters. However, the diagonal information cannot be handled by the front-end 1D CNN stage. Thus, the proposed 1D–2D CNN model employs a back-end 2D CNN model with minimal network structure.

### 3.3 2D convolutional layers

The high-resolution image-reconstruction stage restores the diagonal high-frequency signals that cannot be restored in the 1D CNN stage using 1D filters. The structure is composed of 2D CNNs consisting of 2D filters for high-frequency signal extraction in the diagonal directions.  $Model_d()$  for extracting diagonal high-frequency signals employs  $3 \times 3$  type filters, and the middle layers, excluding the output layer, use ReLU activation functions. The final high-resolution image is reconstructed by

$$HR = Model_d(\widehat{HR}; \theta_d) + \widehat{HR}, \tag{6}$$



**Fig. 3** Block diagram of the proposed one-dimensional (1D)–two-dimensional (2D) convolutional neural network (CNN) for super-resolution (SR).  $ILR$  is an interpolated low-resolution image,  $V$  is a horizontal high-frequency

signal.  $\widehat{HR}$  is an intermediate high-resolution image resulting from  $ILR + H + V$ ,  $D$  is a diagonal high-frequency signal,  $HR$  is a high-resolution image reconstructed using the proposed structure

where  $\widehat{HR}$  is the output image from the first 1D CNN stage with horizontal and vertical filters. That is, the input includes a low-resolution image and high-frequency signals in the horizontal and vertical directions. Therefore,  $\text{Model}_d(\widehat{HR}; \theta_d)$  for the final high-resolution image extracts the high-frequency signals in all directions. The cost function,  $L_{2D}$  for the learning is defined by

$$L_{2D} = \frac{1}{2} \sum_{i=0}^n \|\text{HR}_i - \text{GT}_i\|_2^2, \quad (7)$$

where  $\text{HR}_i$  and  $\text{GT}_i$  are the final reconstructed high resolution and ground truth images, respectively.

### 3.4 Learning method of the proposed 1D–2D CNN model

In the learning stage of deep-learning algorithms, it is important to select the loss function, optimization method, initial values of weights, and data set, appropriately. In this work, learning was performed for 80 epochs using an Adam optimizer [24] as the optimization method. The initial values of the model weights were initialized using the Xavier initializer [25]. The initial values of the learning rate were set to  $1e^{-4}$ , which were then halved every 40 epochs. The proposed 1D–2D CNN SR model consisted of two stages: 1D CNN and 2D CNN. A joint cost function was proposed for the two stages of the 1D–2D CNN SR model, and is defined by

$$L(\theta_h, \theta_v, \theta_d) = L_{2D}(\theta_d) + \lambda_{1D} L_{1D}(\theta_h, \theta_v) + \lambda_{\text{reg}} \sum \|\theta_h\|_2^2 + \lambda_{\text{reg}} \sum \|\theta_v\|_2^2 + \lambda_{\text{reg}} \sum \|\theta_d\|_2^2, \quad (8)$$

where  $\lambda_{1D}$  and  $\lambda_{\text{reg}}$  are set to 0.5 and  $1e^{-4}$ , respectively. We employed a joint cost function that included the regularization terms for optimization. Thus, the overfitting issue could be alleviated by optimizing the proposed cost function.

In this paper, the same 291 images used for the VDSR [17] were used for learning. The training image set consisted of 91 images from Yang [7] and 200 images extracted from BSDS [23]. The training and test images were used for the

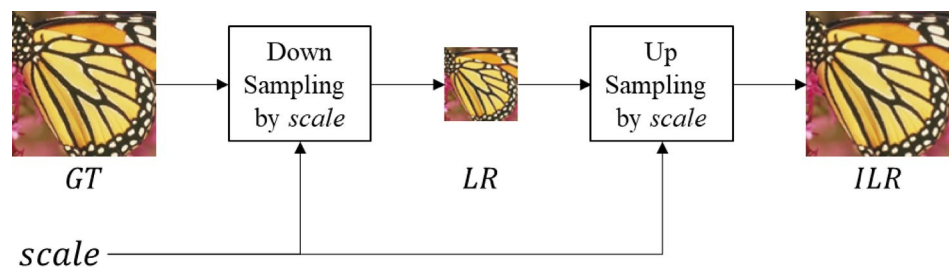
ground truth, high-resolution original images. Their corresponding low-resolution images were obtained by down-sampling the high-resolution images, as shown in Fig. 4. In this paper, the bicubic interpolation method was used for down-sampling and up-sampling, and the learning data were constructed with scales of 2, 3, and 4. In this paper, we used a data augmentation method such as flip and rotation for high-resolution original images and their down-sampled images. The training images were divided into batches of 256 images, and each image was separated into patches of  $48 \times 48$  pixels.

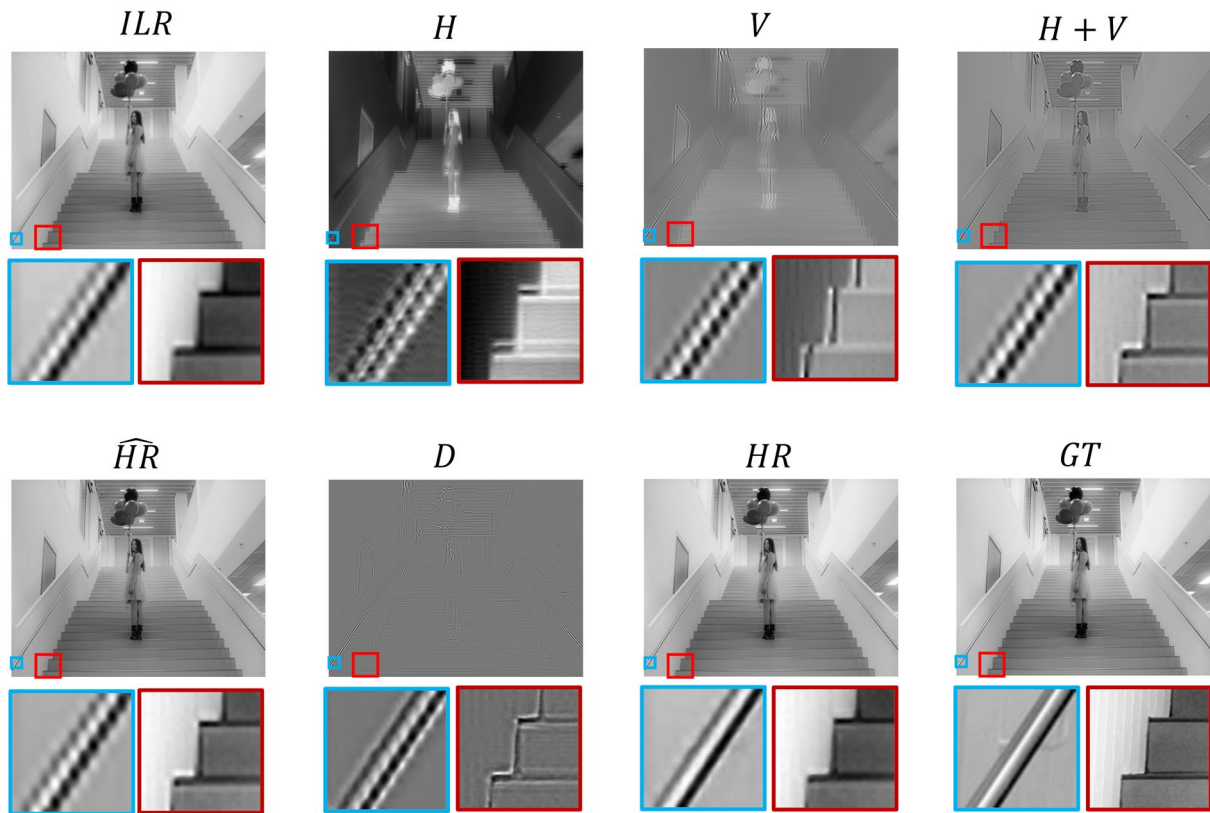
## 4 Experimental results and discussion

In this paper, we used Intel i7-6700 3.40 GHz CPU, 16 GB memory, and GTX 1080 Ti 11 GB GPU for the performance evaluation of the proposed and existing methods. The algorithms were implemented with Python 3.6 and TensorFlow [26] 1.11.0. The visual qualities were evaluated for bicubic, SRCNN [16], VDSR [17], and the proposed algorithms. The speed-up performance of VDSR and the proposed algorithm were also evaluated. VDSR in the literature [17] was implemented using the Matlab-based deep-learning library, MatConvNet [27]. We re-implemented and trained the VDSR using TensorFlow for fair performance comparison in our evaluation. Note that the data sets for VDSR for the learning stage are used for the proposed algorithm. Set5, Set14, B100, and Urban100 were used for performance comparison. The input images were converted to YCbCr and only the Y components were used. In Sect. 4.1, we compare the proposed method with the existing methods, using the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) as evaluation metrics of visual qualities. Section 4.2 evaluates the speed-up factor of the proposed method in CPU, GPU, and TPU environments and compares it with that of VDSR.

In the Fig. 5, the output of the vertical 1D CNN structure,  $V$ , clearly restores the high-frequency signals in the vertical direction, and the output  $H$  of the horizontal 1D CNN

**Fig. 4** Generation of training image data





**Fig. 5** Intermediate output images of the proposed network for the “img009” image of Urban100 with a scale factor of four. ILR at the top of the figure is a low-resolution input image interpolated to the target size,  $V$  is a horizontal high-frequency signal,  $H$  is a vertical high-frequency signal, and  $H + V$  is the sum of horizontal and vertical

high-frequency signals.  $\widehat{HR}$  at the bottom of the figure is a high-resolution feature image resulting from  $ILR + H + V$ ,  $D$  is a diagonal high-frequency signal,  $HR$  is a high-resolution image reconstructed using the proposed structure, and  $GT$  is a high-resolution original image

structure restores the high-frequency signals in the horizontal direction. However, each of them shows strong horizontal and vertical high-frequency signals. We can hardly see the high-frequency signals in the diagonal directions in the output of the 1D CNN model, while the output  $D$  from the 2D CNN model shows a more detailed representation of the diagonal high-frequency signals.

#### 4.1 Comparison of visual quality

Table 1 shows a comparison of the existing and proposed algorithms in terms of the PSNR and SSIM for visual quality. The visual quality of the proposed algorithm is better than that of bicubic and SRCNN. Note that the visual quality of the proposed algorithm is comparable to that of VDSR.

Table 2 shows the differences in PSNR and SSIM between VDSR and the proposed method. The proposed method shows, on an average, a degradation of 0.19 dB in terms of the PSNR, compared to the VDSR method implemented

by TensorFlow. In the case of SSIM, the visual qualities of the proposed and VDSR techniques are approximately the same. Figure 6 shows the original and reconstructed images using the bicubic, SRCNN, VDSR, and the proposed algorithms for the “img090” image of the “Urban100” set and “Baboon” and “Barbara” images of “Set14”. The proposed method shows high visual quality at the boundaries, compared to the bicubic and SRCNN.

#### 4.2 Run-time comparison and complexity

The computational complexity of the VDSR and proposed algorithms were analyzed on GPU and CPU settings. Also, we calculated the number of multiplications required for VDSR and the proposed algorithm. For the run-time comparison, the time-saving ( $\Delta T$ ) was calculated, which can be defined as

**Table 1** Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) of the existing (bicubic, super-resolution convolutional neural network (SRCNN), very deep super-resolution (VDSR)) and proposed super-resolution algorithms

Dataset	Scale	Bicubic		SRCNN [16]		VDSR [17]		Proposed	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	×2	33.66	0.9299	36.66	0.9541	37.15	0.9578	37.08	0.9573
	×3	30.39	0.8682	32.75	0.9081	33.29	0.9186	32.94	0.9140
	×4	28.42	0.8104	30.48	0.8618	30.99	0.8783	30.56	0.8690
Set14	×2	30.24	0.8688	32.45	0.9066	32.75	0.9108	32.73	0.9099
	×3	27.55	0.7742	29.29	0.8217	29.65	0.8298	29.42	0.8253
	×4	26.00	0.7027	27.50	0.7518	27.83	0.7631	27.58	0.7566
B100	×2	29.56	0.8431	31.36	0.882	31.68	0.8944	31.63	0.8930
	×3	27.21	0.7385	28.41	0.7870	28.68	0.7960	28.55	0.7919
	×4	25.96	0.6675	26.90	0.7115	27.15	0.7228	27.01	0.7179
Urban100	×2	26.88	0.8403	29.51	0.8949	30.17	0.9069	30.06	0.9044
	×3	24.46	0.7349	26.24	0.7997	26.74	0.8171	26.46	0.8082
	×4	23.14	0.6577	24.52	0.7236	24.88	0.7399	24.59	0.7278

**Table 2** Difference between very deep super-resolution (VDSR) and the proposed algorithm

Dataset	Scale	Proposed—VDSR [17]	
		PSNR	SSIM
Set5	×2	−0.06	−0.0005
	×3	−0.35	−0.0045
	×4	−0.42	−0.0092
Set14	×2	−0.02	−0.0009
	×3	−0.22	−0.0045
	×4	−0.25	−0.0065
B100	×2	−0.05	−0.0014
	×3	−0.13	−0.0040
	×4	−0.14	−0.0049
Urban100	×2	−0.11	−0.0026
	×3	−0.28	−0.0089
	×4	−0.29	−0.0121
Average		−0.19	−0.0050

$$\Delta T = \frac{T_{\text{ref}} - T_{\text{proposed}}}{T_{\text{ref}}}, \quad (9)$$

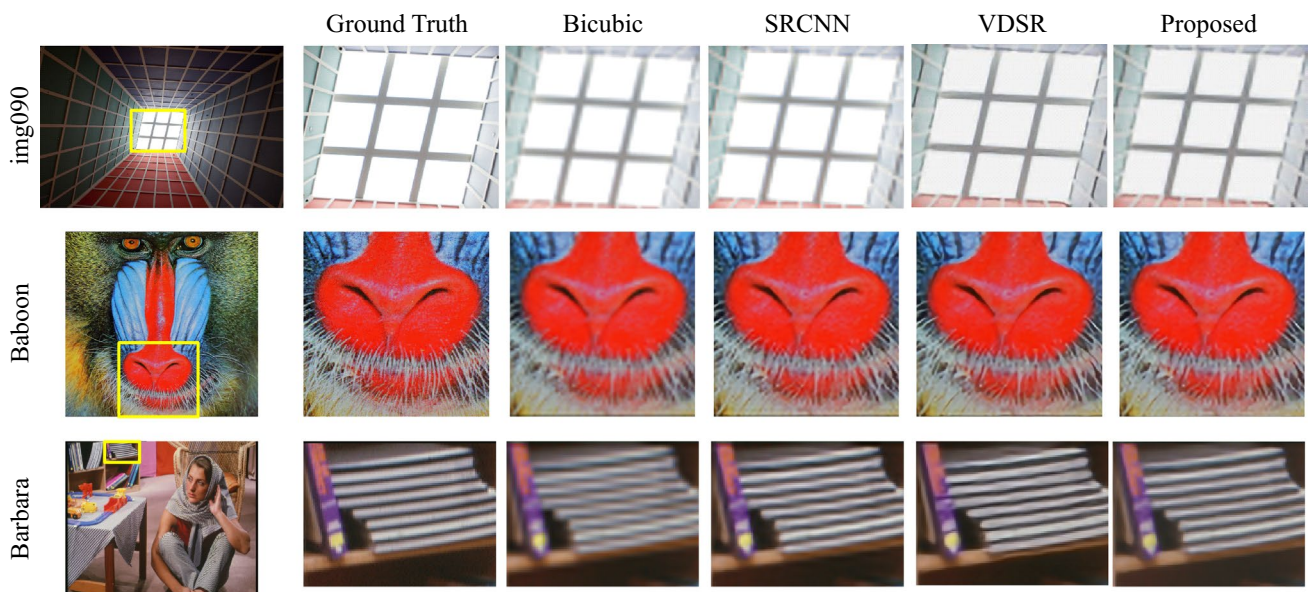
where  $T_{\text{ref}}$  is the run-time of the existing method and  $T_{\text{proposed}}$  is the run-time of the proposed method. Table 3 shows the run-times of SRCNN, VDSR and the proposed algorithms on GPU. The time-saving factors between the proposed algorithm and VDSR are also shown in the table. In the GPU environment, the VDSR requires 0.020 s, on average, whereas the proposed and SRCNN methods require 0.005 s and 0.011 s, on average, respectively. While SRCNN is faster than the proposed algorithm, its PSNR and SSIM are lower than the proposed algorithm, as shown in Table 1. SRCNN

has a smaller number of parameters, thus; it cannot reconstruct the high-frequency components. The average time-saving of the proposed algorithm, compared to VDSR, is approximately 50.79%. Table 4 shows the run-times of the SRCNN, VDSR and the proposed algorithms on Intel CPU. In the CPU environment, SRCNN, VDSR and the proposed methods take 1.04 s, 15.92, and 4.56 s, on average, respectively. We can state that the proposed algorithm can achieve high time-saving of approximately 71.3%.

Table 5 shows the number of multiplications required for VDSR and the proposed algorithm. In CNN frameworks, the number of weights used by the model is the number of multiplications required to process one pixel. Therefore, the theoretical computational complexity is compared based on the number of weights used in the neural network models. From the numerical number of weights, it can be inferred that the proposed method can save 81.22% multiplications.

Table 6 shows the time-saving factors of the proposed method against VDSR on Google Coral TPU. Because two-dimensional (1D filter × # channels) memory access and convolutions are executed by the proposed 1D structure, the proposed network is efficient to achieve high-speed parallelism for the Google TPU. We found that the proposed method with the GPU in Table 3 reduced the average processing time by 50.78%, compared to the previous method. Furthermore, it can reduce the average processing time by 61.82% on the Google Coral Edge TPU. To evaluate the performance of the proposed algorithm in terms of the number channels, we evaluated PSNR, SSIM, and run-time of the proposed network by varying the number of channels (32, 64, and 128) with the number of parameters in Table 7. The proposed method having 128 channels yields visual performance similar to VDSR with lower computational





**Fig. 6** Comparison of super-resolution images reconstructed by bicubic, super-resolution convolutional neural network (SRCNN), very deep super-resolution (VDSR), and proposed algorithm

**Table 3** Run-times of the SRCNN, VDSR, and proposed algorithm on GPU

Dataset	Scale	SRCNN (s)	VDSR (s)	Proposed (s)	Time-saving of proposed-VDSR ( $\Delta T$ ) (%)
Set5	$\times 2$	0.002	0.016	0.008	48.80
	$\times 3$	0.002	0.017	0.008	49.42
	$\times 4$	0.002	0.016	0.008	50.62
Set14	$\times 2$	0.004	0.031	0.016	48.96
	$\times 3$	0.004	0.033	0.016	51.86
	$\times 4$	0.005	0.031	0.016	48.96
B100	$\times 2$	0.003	0.020	0.010	52.84
	$\times 3$	0.003	0.021	0.010	53.48
	$\times 4$	0.003	0.020	0.010	52.98
Urban100	$\times 2$	0.012	0.092	0.045	51.37
	$\times 3$	0.013	0.104	0.052	50.31
	$\times 4$	0.012	0.091	0.045	50.68
Average		0.005	0.041	0.020	50.79

complexity than VDSR. When 32 channels are used for the proposed algorithm, it is faster than that with 64 channels but yields similar visual performance to SRCNN. Therefore, we found that the proposed method is a reasonable trade-off between complexity and visual performance. Depending on the requirements of an application, the number of channels can be selected.

## 5 Conclusion

In this paper, we proposed a low-complexity neural network with 1D and 2D filters. The existing CNN-based SR technologies reconstructed high-resolution images with deep and wide layers, using 2D filters. However,

**Table 4** Run-times of SRCNN, VDSR, and the proposed algorithm on Intel CPU

Dataset	Scale	SRCNN (s)	VDSR (s)	Proposed (s)	Time-saving of proposed-VDSR ( $\Delta T$ ) (%)
Set5	×2	0.37	6.79	1.80	73.54
	×3	0.37	4.81	1.47	69.38
	×4	0.38	6.78	1.85	72.69
Set14	×2	0.75	12.80	3.51	72.57
	×3	0.74	9.44	2.95	68.76
	×4	0.74	12.55	3.56	71.66
B100	×2	0.50	6.47	1.99	69.31
	×3	0.49	6.24	1.94	68.96
	×4	0.50	6.43	1.98	69.30
Urban100	×2	2.54	42.57	11.70	72.52
	×3	2.51	31.53	9.79	68.95
	×4	2.54	44.59	12.16	72.73
Average		1.04	15.92	4.56	71.37

**Table 5** Comparison of the number of multiplications between very deep super-resolution (VDSR) and the proposed algorithm

VDSR	Proposed	$\Delta\#param.$
664,704	124,800	81.22%

**Table 6** Run-times of VDSR, and the proposed algorithm on Raspberry pi 4 with Google edge TPU

Dataset	Scale	VDSR (ms)	The proposed algorithm (ms)	Time-saving of proposed-VDSR ( $\Delta T$ ) (%)
Set5	×2	4.21	1.64	61.1
	×3	4.21	1.62	61.4
	×4	4.33	1.63	62.3
Set14	×2	4.72	1.85	60.8
	×3	4.46	1.77	60.4
	×4	4.51	1.68	62.7
B100	×2	4.77	1.63	65.8
	×3	4.81	1.77	63.3
	×4	4.72	2.02	57.1
Urban100	×2	4.78	1.71	64.3
	×3	4.80	1.78	63.0
	×4	4.79	1.93	59.7
Average		4.59	1.75	61.8

the VDSR model required a higher number of computations and used more memory. For low-complexity CNN-based SR, the proposed method consisted of a 1D feature-extraction layer and 2D image-restoration layers. The first feature-extraction layer was designed to extract the horizontal and vertical high-frequency signals using 1D filters. The second high-resolution image-restoration layers were developed for extracting high-frequency signals in the diagonal directions using 2D filters. Also, we proposed a loss function to optimize the 1D and 2D filters, simultaneously, using regularization terms. In comparison with the TensorFlow-based VDSR model, the proposed method showed a negligible loss in visual quality with significant computational load reduction. Future research will be conducted on the development of neural network structures and learning methods that are robust to various types of distortions such as coding losses and so on.

**Table 7** Performance comparison in terms of PSNR, SSIM, and run-time of the proposed network by varying the number of channels (32, 64, and 128) with the number of parameters on GPU

Dataset	Scale	32 channels (32 K)			64 channels (125 K)			128 channels (495 K)		
		PSNR (dB)	SSIM	Time (ms)	PSNR (dB)	SSIM	Time (ms)	PSNR	SSIM	Time (ms)
Set5	x2	36.99	0.9565	4.579	37.08	0.9573	8.378	37.31	0.9580	17.754
	x3	32.72	0.9093	4.388	32.94	0.9140	8.376	33.08	0.9142	18.536
	x4	30.40	0.8620	4.383	30.56	0.8690	7.976	30.70	0.8692	17.547
Set14	x2	32.68	0.9094	7.763	32.73	0.9099	15.600	32.88	0.9112	34.476
	x3	29.28	0.8220	8.480	29.42	0.8253	15.672	29.52	0.8271	35.267
	x4	27.45	0.7513	7.759	27.58	0.7566	15.671	27.67	0.7581	33.905
B100	x2	31.56	0.8921	4.939	31.63	0.8930	9.622	31.75	0.8944	22.35
	x3	28.45	0.7885	4.989	28.55	0.7919	9.853	28.61	0.7934	22.94
	x4	26.92	0.7127	4.953	27.01	0.7179	9.545	27.05	0.7181	22.30
Urban100	x2	29.87	0.9023	21.90	30.06	0.9044	44.53	30.33	0.9083	111.8
	x3	26.21	0.8003	24.65	26.46	0.8082	51.59	26.57	0.8118	123.2
	x4	24.44	0.7213	21.82	24.59	0.7278	44.87	24.70	0.7325	111.5
Average		29.75	0.8356	10.05	29.88	0.84	20.14	30.01	0.8414	47.63

**Acknowledgements** This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2020-2016-0-00288) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2018R1A2B2008238).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**References**

1. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag.* **20**, 21–36 (2003)
2. Tipping, M.E., Bishop, C.M.: Bayesian image super-resolution. *Adv. Neural Inf. Process. Syst.* **15**, 1303–1310 (2003)
3. Tian, J., Ma, K.-K.: An MCMC approach for Bayesian super-resolution image reconstruction. In: *IEEE Int. Conf. Image Process.*, Genova (2005)
4. Shimizu, M., Yoshimura, S., Tanaka, M., Okutomi, M.: Super-resolution from image sequence under influence of hot-air optical turbulence. In: *2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage (2008)
5. Lee, O.Y., Park, S.J., Kim, J.W., Kim, J.O.: Multi-frame super-resolution of high frequency with spatially weighted bilateral total variance regularization. *IEIE Trans. Smart Process. Comput.* **3**, 271–274 (2014)
6. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. *IEEE Comput. Graph. Appl.* **22**, 56–65 (2002)
7. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **19**, 2861–2873 (2010)
8. Timofte, R., Smet, V.D., Gool, L.V.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: *Asian Conf. Comput. Vis. (ACCV 2014)*, Singapore (2014)
9. Huang, J.-B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: *2015 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston (2015)
10. Jun, J.H., Choi, J.H., Lee, D.Y., Jeong, S., Cho, S.H., Kim, H.Y., Kim, J.O.: Accelerating self-similarity-based image super-resolution using OpenCL. *IEIE Trans. Smart Process. Comput.* **4**, 10–15 (2015)
11. Yuan, Y., Yang, X., Wu, W., Li, H., Liu, Y., Liu, K.: A fast single-image super-resolution method implemented with CUDA. *J. Real Time Image Process.* **16**, 81–97 (2019)
12. Jung, C., Ke, P., Sun, Z., Gu, A.: A fast deconvolution-based approach for single image super-resolution with GPU acceleration. *J. Real Time Image Process.* **14**, 501–512 (2018)
13. Russakovsky, O., Jia, D., Hao, S., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C.,

- Li, F.-F.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015)
14. McCann, M.T., Jin, K.H., Unser, M.: Convolutional neural networks for inverse problems in imaging: a review. *IEEE Sig. Process. Mag.* **34**, 85–95 (2017)
  15. Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K.: Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Process. Mag.* **35**, 20–36 (2018)
  16. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 295–307 (2016)
  17. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas (2016)
  18. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: European Conf. on Comput. Vis. (ECCV). Springer (2016)
  19. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas (2016)
  20. Shamsolmoali, P., Zareapoor, M., Zhang, J., Yang, J.: Image super resolution by dilated dense progressive network. *Image Vis. Comput.* **88**, 9–18 (2019)
  21. Shamsolmoali, P., Li, X., Wang, R.: Single image resolution enhancement by efficient dilated densely connected residual network. *Signal Process. Image Commun.* **79**, 13–23 (2019)
  22. Shamsolmoali, P., Zareapoor, M., Wang, R., Jain, D.K., Yang, J.: G-GANISR: gradual generative adversarial network for image super resolution. *Neurocomputing.* **366**, 140–153 (2019)
  23. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. Eighth IEEE Int. Conf. Comput. Vis. ICCV 2001, Vancouver (2001)
  24. Kingma, D.P., Adam, J.L.B.: A method for stochastic optimization. <http://arxiv.org/abs/1412.6980> (2014)
  25. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proc. Thirteenth Int. Conf. Artif. Intell. Stat. (AISTATS), Chia Laguna (2010)
  26. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems. In: 12th Symp. Oper. Syst. Des. Implementation. (OSDI 16), Savannah (2015)
  27. Lenc, A., Karel, V.: MatConvNet: convolutional neural networks for MATLAB. In: Proc. 23rd ACM Int. Conf. Multimed., Brisbane (2015)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



machine learning.



**Jangsoo Park** received the B.S. and M.S. degrees in Computer Engineering from Kwangwoon University, Korea, in 2016 and 2019, respectively. He was a trainee in software maestro, The Ministry of Science, ICT and Future Planning, Korea from 2015 to 2016. He was a part time researcher in Lee Gil Ya Cancer and Diabetes Institute, Gachon University, Incheon, Korea from 2014 to 2015. His research interests are lightweight neural networks, model based image processing, computer vision and

**Jongseok Lee** received the B.S. and M.S. degrees in Electronic Engineering from Kwangwoon University, Korea, in 2016 and 2018, respectively. He is currently working towards a PhD in computer engineering at the same university. His current research interests are video coding, video processing, computer vision, spiking neural network, and deep learning.



**Donggyu Sim** received the B.S. and M.S. degrees in Electronic Engineering from Sogang University, Korea, in 1993 and 1995, respectively. He also received the Ph.D. degree at the same University, in 1999. He was with the Hyundai Electronics Co., Ltd. From 1999 to 2000, being involved in MPEG-7 standardization. He was a senior research engineer at Varo Vision Co., Ltd, working on MPEG-4 wireless applications from 2000 to 2002. He worked for the Image Computing Systems Lab. (ICSL) at

the University of Washington as a senior research engineer from 2002 to 2005. He researched on ultrasound image analysis and parametric video coding. Since 2005, he has been with the Department of Computer Engineering at Kwangwoon University, Seoul, Korea. In 2011, he joined the Simon Fraser University, as a visiting scholar. He was elevated to an IEEE Senior Member in 2004. He is one of main inventors in many essential patents licensed to MPEG-LA for HEVC standard. His current research interests are video coding, video processing, computer vision, and video communication