Multi-Category Image Super-Resolution with Convolutional Neural Network and Multi-Task Learning

URAZOE, Kazuya ; KUROKI, Nobutaka ; KATO, Yu ; OHTANI, Shinya ; HIROSE, Tetsuya ; NUMA, Masahiro

(Citation)
IEICE Transactions on Information and Systems,E104.D(1):183-193

(Issue Date)
2021-01-01

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90007755

PAPER

Multi-Category Image Super-Resolution with Convolutional Neural Network and Multi-Task Learning

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

SUMMARY This paper presents an image super-resolution technique using a convolutional neural network (CNN) and multi-task learning for multiple image categories. The image categories include natural, manga, and text images. Their features differ from each other. However, several CNNs for super-resolution are trained with a single category. If the input image category is different from that of the training images, the performance of super-resolution is degraded. There are two possible solutions to manage multi-categories with conventional CNNs. The first involves the preparation of the CNNs for every category. This solution, however, requires a category classifier to select an appropriate CNN. The second is to learn all categories with a single CNN. In this solution, the CNN cannot optimize its internal behavior for each category. Therefore, this paper presents a super-resolution CNN architecture for multiple image categories. The proposed CNN has two parallel outputs for a high-resolution image and a category label. The main CNN for the high-resolution image is a normal three convolutional layer-architecture, and the sub neural network for the category label is branched out from its middle layer and consists of two fully-connected layers. This architecture can simultaneously learn the high-resolution image and its category using multi-task learning. The category information is used for optimizing the super-resolution. In an applied setting, the proposed CNN can automatically estimate the input image category and change the internal behavior. Experimental results of 2× image magnification have shown that the average peak signal-to-noise ratio for the proposed method is approximately 0.22 dB higher than that for the conventional super-resolution with no difference in processing time and parameters. We have ensured that the proposed method is useful when the input image category is varying.

key words: *super-resolution, resolution enhancement, convolutional neural network, multi-task learning, deep learning*

1. Introduction

Image up-sampling techniques are used in several applications and devices such as digital cameras, smart phones, and televisions. When inputting a low-resolution image to a high-resolution display device, some resolution conversion is required. The image quality can become blurry and jagged using conventional interpolation methods such as Bicubic interpolation. According to the sampling theorem, signals that exceed the Nyquist frequency (hereinafter, referred to as high-frequency components) are not included in the original signal. Because the Bicubic interpolation is a technique to expand signals below the Nyquist frequency, high-frequency components are not generated.

In contrast, the method called "Super-Resolution" has been intensively investigated recently. The super-resolution method generates a high-resolution image by estimating high-frequency components not included in input signals. The convolutional neural network-based super-resolution (SRCNN), proposed by Dong et al. [1], [2], improves image qualities significantly. After SRCNN was proposed, several super-resolutions with convolutional neural networks (CNNs) have been proposed [1]–[12].

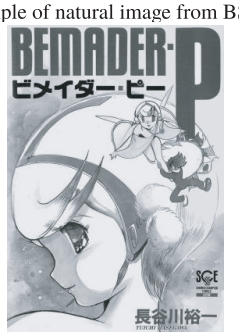
However, most CNN-based super-resolutions are trained for a limited image category. For example, the multi-channel convolutional neural network for image super-resolution (MCH) [5] was trained with the T91 dataset [13] of natural images and then applied to the Set5 [14] and Set14 [15] datasets in the same category. In practical use, however, users can input several types of images. Figure 1 shows examples of a natural image from BSDS100 [16], a manga image from Manga109 [17], [18], and a text image from Hradiš's dataset [19]. They have different features from each other. If a CNN is trained for a specific category, it often causes performance degradation for the other categories. Table 1 presents the image qualities obtained by the three MCHs that are trained with different categories. The MCH trained with natural images exhibits a good peak signal-to-noise ratio (PSNR) of 31.31 dB for natural testing images, whereas the MCH trained with manga images or text images yields unsatisfactory PSNRs of 31.20 dB or 28.08 dB for natural testing images, respectively. Similarly, MCH trained with manga images shows the highest PSNR of 36.65 dB for manga testing images, and MCH trained with text images shows the highest PSNR of 24.66 dB for text testing images. Therefore, we can determine if the CNN achieves a good result for a previously trained category. If we require a super-resolution for multiple image categories with conventional CNN, the possible solutions are as follows:

PlanA Train multiple CNNs with every image category;
PlanB Train single CNN with all image categories.

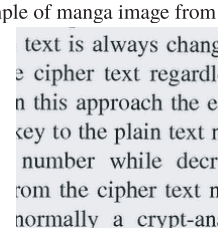
In PlanA, as shown in Fig. 2-(a), the CNNs are trained for each image category. In particular, super-resolution techniques for specific category have been studied [7]. However,

Manuscript received March 11, 2020.
Manuscript revised August 27, 2020.
Manuscript publicized October 2, 2020.
[†]The authors are with the Graduate School of Engineering, Kobe University, Kobe-shi, 657–8501 Japan.
[††]The author is with the Graduate School of Engineering, Osaka University, Suita-shi, 565–0871 Japan.
[*]Presently, with Panasonic Corporation.
[**]Presently, with EIZO Corporation.
[***]Presently, with Toyota Motor Corporation.
a) E-mail: kuroki@kobe-u.ac.jp
DOI: 10.1587/transinf.2020EDP7054

(a) Example of natural image from BSDS100.



(b) Example of manga image from Manga109.



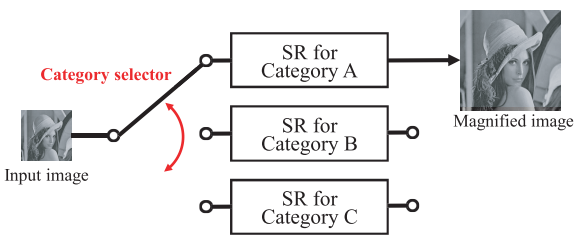
(c) Example of text image from Hradiš's dataset.

Fig. 1 Visual difference between multiple categories.

Table 1 Peak signal-to-noise ratios [dB] of 2× magnified images.

Training images	Testing images		
	Natural images	Manga images	Text images
Natural images	**31.31**	35.48	23.25
Manga images	31.20	**36.65**	24.53
Text images	28.08	29.77	**24.66**

This table lists the results of the super-resolution CNN trained with each image category. Red, bold numbers indicate the best score in each testing dataset. When the training and testing image categories are the same, the CNN yields the highest image quality in each image category.



(a) PlanA (Train multiple CNNs with every image category).

(b) PlanB (Train single CNN with all image categories).

Fig. 2 How to improve the image qualities of multiple categories.

PlanA must estimate the category of the input image manually or automatically to select a suitable CNN. If users know the input image category, they need to select the optimal CNN manually and determine in advance the type of images used for learning the CNN. In addition, the solution requires maintaining the parameters of multiple CNNs. Therefore, PlanA is not suitable for low-end devices with a small memory capacity.

In PlanB, as shown in Fig. 2-(b), a single CNN is trained with images of all categories. However, because the CNN has difficulty in understanding the image categories, it operates uniformly for all inputs. In this case, although the CNN can show better results for all categories, it cannot yield the best result for each.

Moreover, [11], [12], [20] introduced some indicators for improving image quality performance after investigating recent super-resolution techniques. A promising strategy is to learn the potential correlations among image characteristics and reflect it in the loss function. Although the mean square error is widely used for the loss function between low-resolution and high-resolution images, it does not represent the object information of the image. Thus, the CNN is trained uniformly without understanding the object.

Moreover, multi-task learning has been studied in recent deep learning [12], [21]–[25]. Multi-task learning is a technique that manages multiple tasks with single machine learning model and improves main-task performance using the optional information contained in sub-tasks [21]. In CNN-based super-resolutions, multi-task learning is implemented by combining multiple loss functions [24], [25]. Shi et al. improved the image quality by adding the optional information about object boundaries on sub-networks [24]. Rad et al. proposed an architecture that can generates the super-resolution image and the semantic information of natural image [14], [15], [25], [26]. Reference [12] stated that because different tasks focus on different aspects of the data, combining related tasks with super-resolution models usually improves the performance of the super-resolution by providing additional information and knowledge.

Based on our survey, we considered the training for understanding image categories to be a promising solution for improving the image quality performance. Thus, we propose an image super-resolution method that is adaptable to multiple image categories, implemented with a single CNN and multi-task learning. The proposed architecture has two parallel outputs: a magnified image and its category. In the proposed architecture, the main CNN is equivalent to the conventional super-resolution CNN. Moreover, an optional neural network (NN) branches out from the middle layer of the main CNN. The branched NN is used to learn the image categories in the training phase. This is removed in the inference phase. This removable architecture can make an internal category classifier in the super-resolution CNN. Because the trained CNN has the category classification ability, users do not need to input the image category manually. Therefore, our method can address the problems of PlanA and PlanB.

The remainder of this paper is organized as follows. Section 2 details the outline of MCH proposed by Ohtani et al. Section 3 introduces the multi-category image super-resolution with CNN and multi-task learning. Finally, Sect. 4 presents experiments to show the usefulness of the proposed method.

2. Conventional Method

In this section, we provide an overview of the MCH, proposed by Ohtani et al. [4], [5].

2.1 Multi-Channel Convolutional Neural Network for Image Super-Resolution (MCH)

Figure 3 shows an overview of the MCH. Let K be the magnification ratio. To obtain a $K\times$ magnified image, the MCH generates $K \times K$ pixels per input pixel. In the case of $2\times$ magnification, there are four types of output pixel locations, as shown in Fig. 4: (A) upper left, (B) upper right, (C) lower left, and (D) lower right of the nearest input pixel. These output pixels are generated from K^2 output channels of the single CNN. Figure 5 shows the architecture of the MCH. Let $\boldsymbol{Y}$ be the low-resolution image. The outputs of the first convolutional layer are calculated as

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

where W_1 is the filter for convolution, B_1 is the bias, and "$*$" denotes the convolution operation. W_1 is 4-dimensional tensor, and its size is $1\times f_1\times f_1\times n_1$, where $f_1\times f_1$ is the filter size, and n_1 is the number of filters in the first convolutional layer. $\max(0, x)$ denotes the rectified linear unit [27]. The outputs of the second convolutional layer are calculated as

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

The size of W_2 is $n_1 \times f_2 \times f_2 \times n_2$, where $f_2 \times f_2$ is the filter size, and n_2 is the number of filters in the second convolutional layer. Finally, the K^2 pixels are calculated as

$$F_3(\boldsymbol{Y}) = W_3 * F_2(\boldsymbol{Y}) + B_3, \tag{3}$$

where $F_3(\boldsymbol{Y})$ has K^2 channels. The size of W_3 is $n_2\times f_3\times f_3\times K^2$. Let $F_{3.1}(\boldsymbol{Y})$, $F_{3.2}(\boldsymbol{Y})$, $\cdots$, $F_{3.K^2}(\boldsymbol{Y})$ be separate channels of $F_3(\boldsymbol{Y})$. Then, the super-resolution image, $F(\boldsymbol{Y})$, can be obtained using

$$F(\boldsymbol{Y}) = \uparrow \mathrm{A}(F_{3.1}(\boldsymbol{Y})) + \uparrow \mathrm{B}(F_{3.2}(\boldsymbol{Y})) + \cdots, \tag{4}$$

where $\uparrow$ A( ), $\uparrow$ B( ), $\cdots$ mean upsampling to their locations on the magnified image. Figure 6 shows an example of a 2×2 synthesis. In the MCH, all the convolutional layers have no padding to prevent image border effects.

2.2 Training Method

In the training phase, we prepare original high-resolution images and their down-sampled images by the Bicubic interpolation. The CNN is trained with the low-resolution image $\boldsymbol{Y}$ as input signal and the high-resolution image $\boldsymbol{T}$ as the teaching signal. The loss function of MCH is the mean squared error (MSE) given by

$$L_{\mathrm{MSE}}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|F(\boldsymbol{Y}_i;\Theta) - \boldsymbol{T}_i\|^2, \tag{5}$$

where N is the number of training data samples, and Θ is parameters to be determined by backpropagation technique such as W_1, W_2, W_3, B_1, B_2, and B_3. The learning rate is 10^{-4} for the first and second convolutional layers and 10^{-5} for the third convolutional layer in Ref. [5].

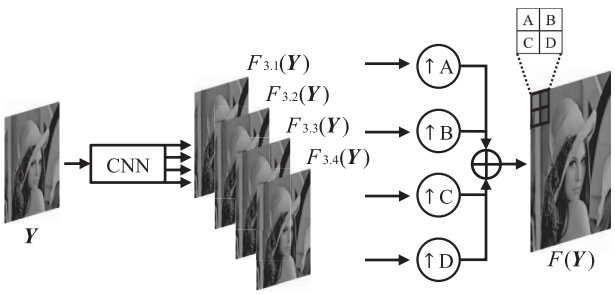


Fig. 3 Multi-channel CNN. ($\boldsymbol{Y}$ and $F(\boldsymbol{Y})$ represent the low-resolution image and the super-resolution image, respectively. $F_{3.1}(\boldsymbol{Y})$, $F_{3.2}(\boldsymbol{Y})$, $F_{3.3}(\boldsymbol{Y})$, and $F_{3.4}(\boldsymbol{Y})$ denote separate channels of $F_3(\boldsymbol{Y})$. ↑ A, ↑ B, ↑ C, and ↑ D indicate upsampling to their locations on the magnified image as shown in Fig. 6)

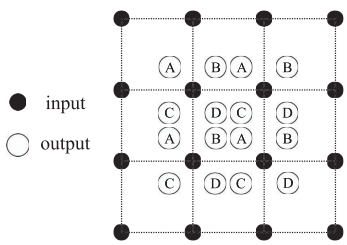


Fig. 4 Input/output pixel location ($2\times$ magnification).

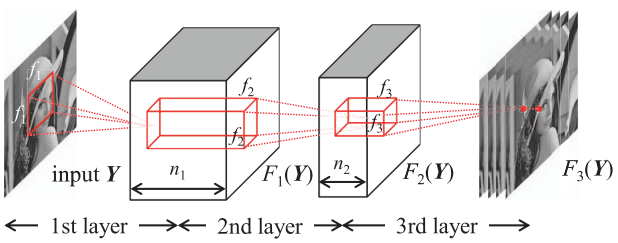


Fig. 5 Architecture of MCH.

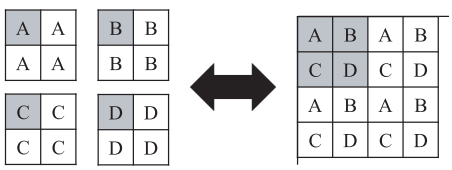


Fig. 6 Separation and synthesis of pixels ($2\times$ magnification).

3. Multi-Category Image Super-Resolution with Convolutional Neural Network and Multi-Task Learning

In this section, we propose an image super-resolution technique that is adaptable to multiple image categories, implemented with a single CNN and multi-task learning. Figure 7 shows the concept of the proposed method. This architecture has an internal category classifier; however, it has no input or output for the category information. The category information is estimated internally and only used for optimizing the super-resolution.

The remainder of this section is organized as follows. Section 3.1 presents the CNN architecture of the proposed method. Section 3.2 explains the training method of the proposed method. Finally, Sect. 3.3 details the behavior of the proposed method in inference. In the following, we refer to the proposed method as multi-channel convolutional neural network for multi-category (MCH-MC).

3.1 Proposed CNN Architecture

Figure 8 shows the proposed CNN architecture. Let $\boldsymbol{Y}$ be the low-resolution image. The output from the first to third convolutional layers is given by the same approach as MCH

$$F_1(\boldsymbol{Y}) = \max(0, W_1 * \boldsymbol{Y} + B_1), \tag{6}$$

$$F_2(\boldsymbol{Y}) = \max(0, W_2 * F_1(\boldsymbol{Y}) + B_2), \tag{7}$$

$$F_3(\boldsymbol{Y}) = W_3 * F_2(\boldsymbol{Y}) + B_3. \tag{8}$$

Thereafter, super-resolution image $F(\boldsymbol{Y})$ is generated by

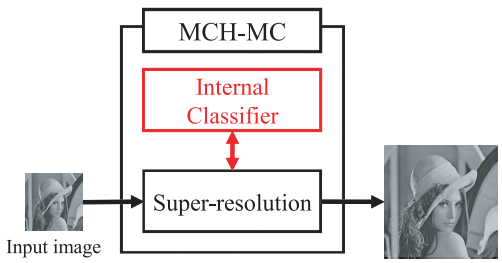


Fig. 7 Concept of the proposed architecture for multiple categories.

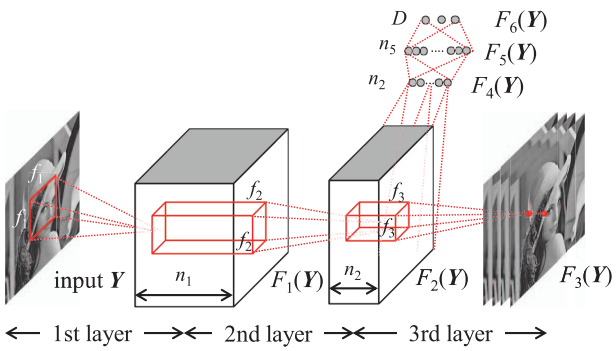


Fig. 8 The architecture of MCH-MC.

synthesizing each channel $F_{3.1}(\boldsymbol{Y})$, $F_{3.2}(\boldsymbol{Y})$, $\cdots$, $F_{3.K^2}(\boldsymbol{Y})$ as shown in Eq. (4).

In contrast, image classification NN is branched from the second feature map, $F_2(\boldsymbol{Y})$. Global average pooling (GAP) is a technique that outputs the average pixel value of each feature map [28]. Using GAP, one-dimensional vector $F_4(\boldsymbol{Y})$ of length n_2 is calculated as

$$F_4(\boldsymbol{Y}) = \mathrm{GAP}(F_2(\boldsymbol{Y})), \tag{9}$$

where $\mathrm{GAP}(F_2(\boldsymbol{Y}))$ outputs the pixel averages of every feature map, $F_2(\boldsymbol{Y})$. GAP reduces a 3-dimensional matrix to a 1-dimensional vector. The outputs of the first fully-connected layer are calculated as

$$F_5(\boldsymbol{Y}) = \max(0, W_5 \cdot F_4(\boldsymbol{Y}) + B_5), \tag{10}$$

where $F_5(\boldsymbol{Y})$ is an n_5-length vector. W_5 is a 2-dimensional matrix, and its size is $n_2 \times n_5$. The outputs of the second fully-connected layer are given by

$$F_6(\boldsymbol{Y}) = W_6 \cdot F_5(\boldsymbol{Y}) + B_6, \tag{11}$$

where $F_6(\boldsymbol{Y})$ is a D-length vector, and D is the number of image categories. Let $F_{6.1}(\boldsymbol{Y})$, $F_{6.2}(\boldsymbol{Y})$, $\cdots$ separate units of $F_6(\boldsymbol{Y})$. The ith output of SoftMax function is calculated as

$$F_{\mathrm{soft}.i}(\boldsymbol{Y}) = \frac{\exp(F_{6.i}(\boldsymbol{Y}))}{\sum_{j=1}^{D} \exp(F_{6.j}(\boldsymbol{Y}))}, \tag{12}$$

where $F_{\mathrm{soft}.i}$ means ith unit of F_{soft}, and i is from 1 to D. Each unit of F_{soft} stores the probability of the input image category.

3.2 Training Method

This subsection presents details of the training method of the proposed CNN architecture. MCH-MC has two training phases for the internal classifying system, as shown in Fig. 9-(a) and (b). First, the CNN is pre-trained to classify input image categories in Phase1. Next, the CNN is trained to improve image quality performance for super-resolution in Phase2.

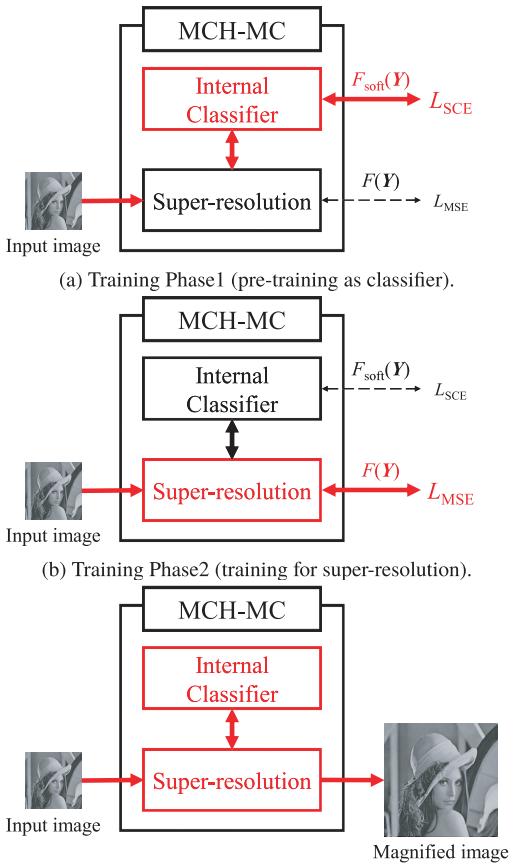
MCH-MC has two outputs: $F(\boldsymbol{Y})$ and $F_{\mathrm{soft}}(\boldsymbol{Y})$; therefore, MCH-MC is trained with multi-task learning [21]. MCH-MC requires two teaching signals of $\boldsymbol{T}$ and $\boldsymbol{t}$. $\boldsymbol{T}$ is an original high-resolution image corresponding to a low-resolution image $\boldsymbol{Y}$. The loss function is calculated using Eq. (5). In contrast, $\boldsymbol{t}$ is a D-length vector with accurate category labels. The loss function is calculated with SoftMax cross entropy (SCE) by

$$L_{\mathrm{SCE}}(\Theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{D} \boldsymbol{t}_{ij} \log F_{\mathrm{soft}.j}(\boldsymbol{Y}_i; \Theta), \tag{13}$$

where N is the number of training data samples, and Θ is parameters to be determined by backpropagation technique such as W_1, W_2, W_5, W_6, B_1, B_2, B_5, and B_6.

Table 2 Training conditions of MCH-MC in each phase.

Phase	Loss ratio		Learning rate		Dropout ratio
	r_{MSE}	r_{SCE}	Other layers	Third convolutional layer	
Phase1	0.1	1	10^{-3}	10^{-4}	0.5
Phase2	1	0.1	10^{-4}	10^{-5}	None



(a) Training Phase1 (pre-training as classifier).

(b) Training Phase2 (training for super-resolution).

(c) Inference Phase (super-resolution).

Fig. 9 Two training phases for the internal classifying system and the inference phase.

After calculating each loss, the total loss $L(\Theta)$ is defined as

$$L(\Theta) = r_{\mathrm{MSE}} L_{\mathrm{MSE}}(\Theta) + r_{\mathrm{SCE}} L_{\mathrm{SCE}}(\Theta), \tag{14}$$

where r_{MSE} and r_{SCE} are composite ratios. This training method is a kind of multi-task learning [21]. The balance between r_{MSE} and r_{SCE} is changed, as presented in Table 2. We explain the details of each training phase.

3.2.1 Phase1: Pre-Training as Classifier

In Phase1, the CNN is pre-trained to improve the image classification ability. Therefore, r_{SCE} is set ten times larger than r_{MSE}, as presented in Table 2.

In addition, Dropout [29] is inserted after the first and second convolutional layers to prevent the CNN from over-optimized for image classification tasks. Dropout is a technique that randomly disables neuron of intermediate layer at the ratio D_{ratio} and improves the generalization performance by preventing overfitting. In MCH-MC, D_{ratio} is 0.5.

3.2.2 Phase2: Training for Super-Resolution

In Phase 2, the CNN is trained for the super-resolution. All weighted layers are inherited from the pre-trained model obtained from Phase1 [30]. Thereafter, r_{MSE} and r_{SCE} were set to 1 and 0.1 as presented in Table 2, respectively. Herein, $L(\Theta)$ operates to improve the super-resolution ability. r_{SCE} is set to a small value for maintaining image classification ability. Dropout layers used in Phase1 are removed from the CNN architecture. The learning rate of Phase2 are reduced from Phase1 and is set to the same values as used in Ref. [5] for a fair comparison.

3.2.3 Approach for Unbalanced Training Dataset

If the number of training images between each category is not always the same. Then, SoftMax Cross Entropy is significantly influenced by the principal category that has more images than others. Consequently, although the CNN operates with higher classification accuracies for the principal category, it has lower accuracies for the remaining category. To address this problem, the MCH-MC adopts Weighted SoftMax Cross Entropy to correct the unbalance of data samples between each category. The weighted SoftMax cross entropy is given by

$$L_{\mathrm{WSCE}}(\Theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{D} \alpha_j t_{ij} \log F_{\mathrm{soft}.j}(\boldsymbol{Y}_i; \Theta), \tag{15}$$

where the weight α_j is an inversely proportional value to the quantity of data samples of category j. Consequently, the CNN can be trained appropriately, if the training dataset is nonuniform for each category. In Sect. 4, MCH-MC adopts L_{WSCE} alternate to L_{SCE}.

3.3 Inference Phase for Super-Resolution

After training the CNN, super-resolution processing is performed with the trained model. In this phase, the GAP and fully-connected layers used for image classification are removed, as shown in Fig. 9-(c). Consequently, the inference CNN is the same architecture as MCH as shown in Fig. 5. Therefore, there are no differences in the processing time and the number of parameters to be stored in the device.

However, the MCH-MC has an image classification ability in the first and second convolutional layers. We will show this ability in later experiments (Sect. 4.3.4). Therefore, MCH-MC can respond to the image category and

change the internal behavior automatically. We do not require specifying the input image category or determine the kind of images the CNN learned in advance. This is an advantage of the proposed method.

4. Experiments and Results

In this section, we show the effectiveness of the proposed method. The remainder of this section is organized as follows: Sect. 4.1 and 4.2 explains the experimental conditions and descriptions, respectively. Section 4.3 evaluates the effectiveness of the proposed method from five viewpoints. Section 4.3.1 evaluates the image qualities and processing times quantitatively. Section 4.3.2 shows the transition in image quality performance and backpropagations. Section 4.3.3 shows transition in classification accuracy and image quality. Section 4.3.4 visualizes the importance areas for the classification judgement with Grad-CAM. Section 4.3.5 shows the correlation between image quality and classification results.

4.1 Experimental Conditions

Our experimental environment is as follows—OS: Ubuntu 16.04 LTS, CPU: Intel Core i7-8700 3.20GHz, Memory: 16.00GB, GPU: NVIDIA GeForce GTX 1080 8GB, and IDE: MATLAB.

All CNNs are implemented with a Caffe package [31]. In all experiments, we focus only on the luminance channel in the YCrCb space because several studies of super-resolution are evaluated only on the luminance channel [2]–[10], [24]. We evaluate 2× image magnification. PSNR and structural similarity (SSIM) are calculated only at the center of the luminance channel to prevent the influence of the image boundaries.

4.2 Experimental Descriptions

In our experiment, we evaluate the performances by K-fold cross-validation. The number of divisions is five.

The training and testing images are three datasets, as presented in Table 3. We define BSDS100, which is a dataset of natural images, Manga109, which is a dataset of comic images, and Hradiš's dataset, which is a dataset of text images, as "Nature," "Manga," and "Text," respectively.

In the training phase, we prepare 60×60 pixel teaching signals, $\boldsymbol{T}$, cropped from original images with a stride s pixel. The sizes of stride s are five for "Nature" and "Text," and 10 for "Manga." The input signals, $\boldsymbol{Y}$, are prepared by downsampling these teaching signals with the Bicubic interpolation of $1/2$ scaling factor. The MSE loss function is optimized with the central pixels of $\boldsymbol{T}$ because all convolutional layers have no padding. The teaching signals of image category $\boldsymbol{t}$ are vectors of three elements corresponding to "Nature," "Manga," and "Text." We do not use any data augmentation for the training and testing images. According to [5], the filter sizes of CNN are as shown in Table 4. The filter

Table 3 Training and testing datasets.

Dataset	Amount	Ave.Pixels	Category name
BSDS100 [16]	100	154,401	"Nature"
Manga109 [17], [18]	109	966,011	"Manga"
Hradiš's Dataset [19]	100	40,000	"Text"

Table 4 Parameter setting of CNN for 2× magnification.

MCH-MCH-MC	MCH-MC
$\{f_1, f_2, f_3\}$={5, 5, 3}	n_5=256
$\{n_1, n_2\}$={64, 32}	D=3

Table 5 Setting of weighted SoftMax cross entropy parameter, "$\boldsymbol{\alpha}$," on MCH-MC.

K-fold	α_{Nature}	α_{Manga}	α_{Text}
1	0.4760	0.0888	1
2	0.4760	0.0888	1
3	0.4760	0.0888	1
4	0.4751	0.0886	1
5	0.4995	0.0931	1

Table 6 Training datasets and the CNN architecture of each method.

Method	Training dataset			Architecture
	Nature	Manga	Text	
SRforNature	✓			MCH
SRforManga		✓		MCH
SRforText			✓	MCH
SRforAll	✓	✓	✓	MCH
SRforMC	✓	✓	✓	MCH-MC

weights of each layer are initialized by random values from a Gaussian distribution with a mean and standard deviation of 0 and 0.001, respectively, and 0 for biases. The Adam optimizer is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [32]. The aforementioned parameters are common for all the methods. We set the weight, $\boldsymbol{\alpha}$, of the MCH-MC as presented in Table 5. The number of backpropagations of Phase1 in the MCH-MC is 1 million (32 batch size). The number of backpropagations of the MCH and MCH-MC in Phase2 are 10 million (32 batch size) at the maximum. We examine the following five methods:

- SRforNature
- SRforManga
- SRforText
- SRforAll
- SRforMC (SRforMulti-Category)

These methods have different CNN architectures and training images as shown in Table 6.

4.3 Experimental Result and Discussion

4.3.1 Performance of Image Qualities and Processing Times

Table 7 lists the PSNRs and SSIMs for 2× magnified images of testing datasets. Red bold and underlined blue numbers indicate the highest and second highest values, respectively. First, we focus on each category. On category "Nature,"

Table 7 PSNRs and SSIMs of 2× magnified images.

Method	PSNR[dB] / SSIM			
	Nature	Manga	Text	Average
SRforNature	**31.31** / **0.8883**	35.48 / 0.9656	23.25 / 0.9203	30.17 / 0.9259
SRforManga	31.20 / 0.8868	**36.65** / **0.9698**	24.53 / 0.9423	30.96 / 0.9340
SRforText	28.08 / 0.7772	29.77 / 0.8711	24.66 / 0.9426	27.57 / 0.8639
SRforAll	31.22 / 0.8870	36.25 / 0.9683	25.28 / 0.9510	31.07 / 0.9364
SRforMC	**31.31** / **0.8883**	36.46 / 0.9694	**25.57** / **0.9565**	**31.29** / **0.9389**

Red bold and underlined blue numbers indicate the best and second-best scores in each testing dataset, respectively.

Fig. 10 Visual comparison on 2× magnified images. (Red bold and underlined blue numbers indicate the best and second-best scores in each testing image, respectively.)

SRforNature and SRforMC have almost the same image quality performances. On category "Manga," SRforManga achieved the highest image quality performances. On category "Text," however, SRforText did not achieve the highest image quality performance, because the number of training images used for SRforText is significantly fewer than that for other methods. From these results, we observe that CNN-based super-resolution achieves a good image quality if sufficient number of images of the same category is learned previously. However, if there are few training images of the same category, the average PSNR and SSIM are degraded.

Next, we focus on SRforAll and SRforMC. These methods have the same training datasets; however, they have different CNN architecture. SRforMC achieved a 0.22pt higher average PSNR and a 0.0025pt higher average SSIM than SRforAll. This is because MCH-MC is trained with a category information. Thus, SRforMC is the best solution if the input image category is variable.

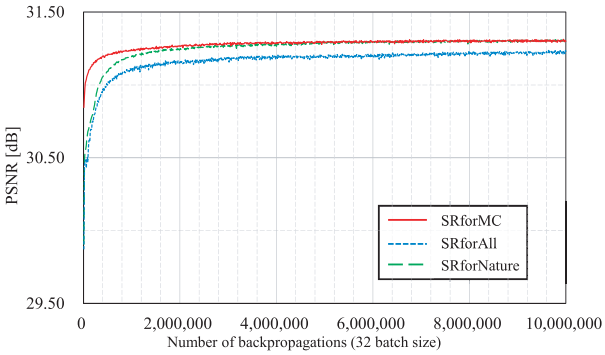
Furthermore, we evaluate the image quality subjectively. Figure 10 shows examples of magnified images.

Table 8 Processing time of 2× magnified images in all categories.

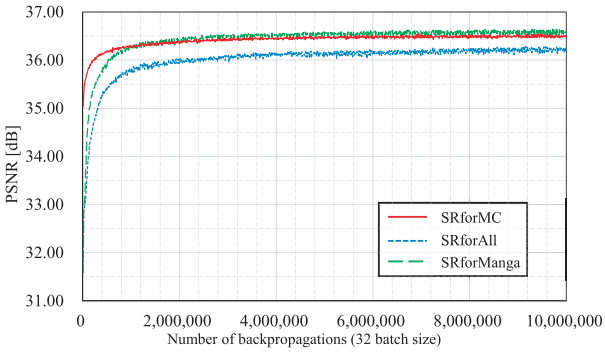
Processing time[s]	
MCH	MCH-MC
0.8003	0.8003

There are significant subjective degradations in "108005" by SRforText, "DualJustice" by SRforText, and "0000001" by SRforNature. The reason is an inconsistency between the input and training image categories. Whereas SRforText enhances the edges, SRforNature smoothens them. Therefore, we must consider the training image category of the CNN before using it.

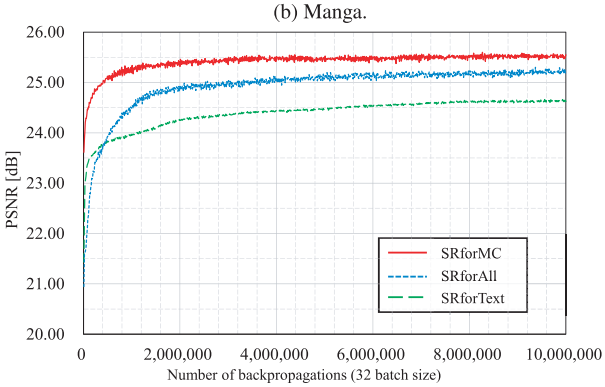
Finally, we evaluate the processing time on a CPU. GPU acceleration is not used in this evaluation because MCH is developed for low complexity and high-speed processing [5]. Table 8 shows the processing time for each architecture at inference. There was no difference in processing time because the MCH-MC of the inference mode has the same CNN architecture as the MCH shown in Fig. 5. Consequently, the processing times of all methods were the same.



(a) Nature.



(b) Manga.



(c) Text.

Fig. 11 The average PSNR curves for the number of backpropagations. (2× magnification)

However, a comparison of super-resolutions is basically evaluated with standard benchmark training and testing datasets including T91 [13], BSDS200 [16], Set5 [14], Set14 [15], and BSDS100 [16]. The comparison of MCH and other super-resolutions is stated in [5]; therefore, we do not consider it in this study.

4.3.2 PSNR Convergence Curves

Figure 11 shows the PSNR convergence curves. In the figure, the PSNR curves of SRforMC shows those of Phase2. The horizontal and vertical axes of these curves indicate the number of backpropagation and the average PSNR, respectively. In general, the image quality performance of CNN improves with the number of backpropagations. SRforMC achieved higher PSNRs than SRforAll at the same number of backpropagation points. This advantage will may not change after 10 million backpropagations.

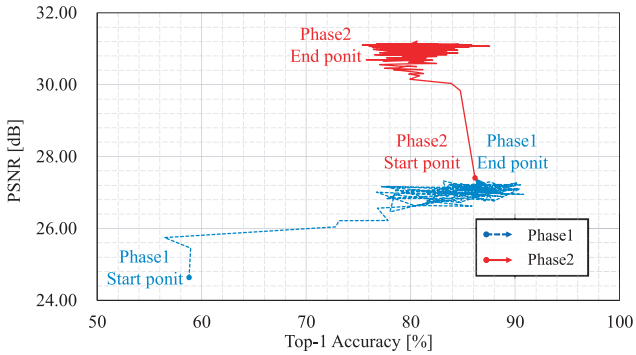


Fig. 12 Transitions in classification accuracy and image quality on SRforMC.

Table 9 Top-1 accuracy of image classification in SRforMC.

Top-1 Accuracy[%]			
Nature	Manga	Text	Average
72.0	96.4	89.0	85.8

4.3.3 Transition in Classification Accuracy and Image Quality

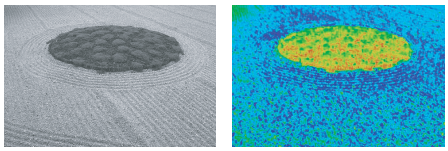
Figure 12 shows the transition in classification accuracy and image qualities in SRforMC for all testing images. The horizontal and vertical axes in this figure indicate the average accuracy of image classification and the average PSNR, respectively. The Phase1 start point is at the lower left of this figure and is obtained after 1,000 backpropagations. Next, the point moves to the right to indicate that the CNN is trained to improve image classification accuracy. Subsequently, the point moves to the top, indicating that CNN is trained to improve the PSNRs while maintaining the classification accuracy.

Table 9 presents the final accuracies of image classification in SRforMC. The average accuracy is 85.8%. Therefore, we can ensure that SRforMC has both the super-resolution and image classification abilities.

4.3.4 Visualization of Classification Ability

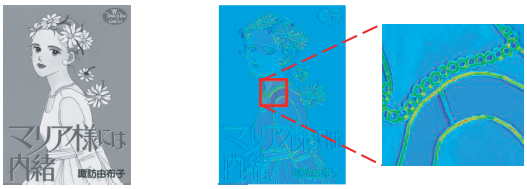
We subjectively evaluate the CNN's internal behavior for each category with gradient-weighted class activation mapping (Grad-CAM) [33]. Grad-CAM is a technique that considers the important area for assessments as a heat map. In this experiment, Grad-CAM shows which area is recognized as "Nature," "Manga," or "Text" in the second feature map, $F_2(\boldsymbol{Y})$.

Figure 13 shows some examples of Grad-CAM outputs for each category. By observing the red and yellow areas, we subjectively found that "Nature," "Manga," and "Text" are based on uneven, flat, and edge areas, respectively. Thus, the first and second convolutional layers extract important characteristics for the classification.



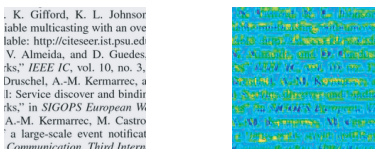
Input image (Nature: “86016”) Output of Grad-CAM

(a) Important areas classified as “Nature.”



Input image (Manga: “MariaSamaNihaNaisyo”) Output of Grad-CAM

(b) Important areas classified as “Manga.”



Input image (Text: “0000099”) Output of Grad-CAM

(c) Important areas classified as “Text.”

Fig. 13 Importance of judgment grounds in each category classification.

Table 10 Increase in image qualities from SRforAll to SRforMC.

Increase of PSNR[dB] / Increase of SSIM	
True	False
+0.20 / +0.0027	+0.14 / +0.0017

4.3.5 Correlation between Image Quality and Classification

In this subsection, we evaluate whether the image quality depends on the classification accuracy. Table 10 shows the increase in the average PSNRs from SRforAll to SRforMC according to the success or failure of the category classification. The increase in the PSNRs and SSIMs is larger in the correctly classified image group than in the misclassified image group.

In contrast, some image samples improved in image quality despite misclassification; therefore, we discuss this result. Figure 14 shows the increase in PSNR from SRforAll to SRforMC for each testing image. The horizontal and vertical axes in this figure indicate the PSNR increase from SRforAll to SRforMC and their distributions, respectively. The distribution of correct classification spreads to the right compared to that of misclassification. This result indicates that these misclassified images insignificantly improved in image quality than the correctly classified images. However, some misclassified images achieved significant image quality improvement.

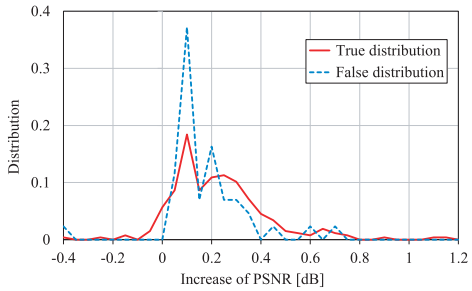
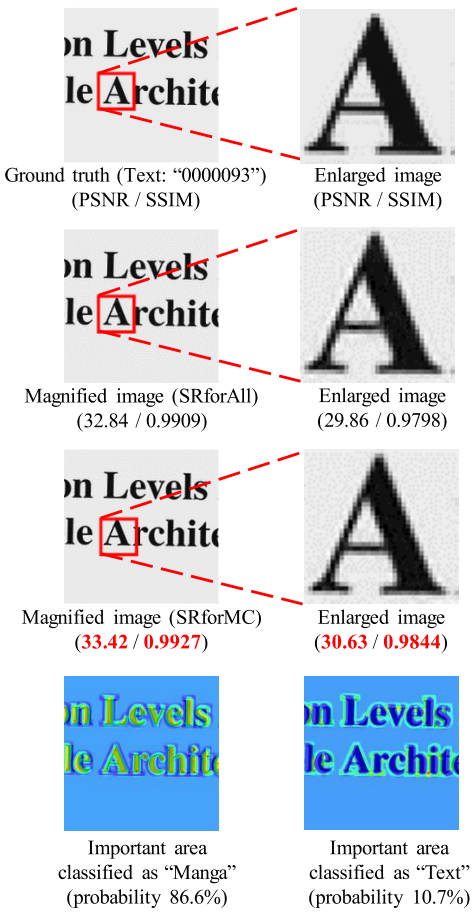


Fig. 14 Increase in PSNR from SRforAll to SRforMC depending on the classification result.



Ground truth (Text: “0000093”) (PSNR / SSIM) Enlarged image (PSNR / SSIM)

Magnified image (SRforAll) (32.84 / 0.9909) Enlarged image (29.86 / 0.9798)

Magnified image (SRforMC) (**33.42 / 0.9927**) Enlarged image (**30.63 / 0.9844**)

Important area classified as “Manga” (probability 86.6%) Important area classified as “Text” (probability 10.7%)

Fig. 15 Image sample with a large quality improvement despite misclassification in SRforMC.

Figure 15 shows an example that significantly improved the PSNR despite the misclassification. This sample is a “Text” with a large font size. Then, the CNN estimated it as a “Manga” with a probability of 86.8% or a “Text” with a probability of 10.7%. Grad-CAM shows that the black flat parts caused the “Manga” probability, and the edge parts caused the “Text” probability. Because the former probability is larger than the latter, the CNN seems to estimate it as a “Manga.” However, SRforMC improved the 0.58pt PSNR and reduced the overshoots around the edge area, as shown in the enlarged images. Therefore, the pro-

posed method made accurate decisions locally, leading to the improved image quality. Therefore, SRforMC can perform appropriate processing based on the image characteristics, although image categories are not accurately classified.

5. Conclusion

In this study, we propose an image super-resolution for multiple image categories with a single CNN and multi-task learning. The proposed CNN does not require an external category information; however, it has an internal category-classifying ability. In our experiments, the average PSNR of the proposed method was approximately 0.22 dB higher than that of the conventional method with the same training dataset and processing time. The proposed method is useful when the input image category is varying. Implementation for a single image containing multi-category will be considered in our future studies.

References

[1] C. Dong, C.C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," Computer Vision – ECCV 2014, vol.8692, pp.184–199, 2014.
[2] C. Dong, C.C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.38, no.2, pp.295–307, 2016.
[3] C. Dong, C.C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," Computer Vision – ECCV 2016, vol.9906, pp.391–407, 2016.
[4] Y. Kato, S. Ohtani, N. Kuroki, T. Hirose, and M. Numa, "Image super-resolution with multi-channel convolutional neural networks," 14th IEEE International New Circuits and Systems Conference (NEWCAS), 2016.
[5] S. Ohtani, Y. Kato, N. Kuroki, T. Hirose, and M. Numa, "Multi-channel convolutional neural networks for image super-resolution," IEICE Trans. Fundamentals, vol.E100-A, no.2, pp.572–580, 2017.
[6] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[7] H.T. Tran and T. Ho-Phuoc, "Deep laplacian pyramid network for text images super-resolution," 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), 2019.
[8] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[9] K. Urazoe, N. Kuroki, Y. Kato, S. Ohtani, T. Hirose, and M. Numa, "Super-resolution with multi-path convolutional neural networks," IEEJ Transactions on Electronics, Information and Systems, vol.140, no.6, pp.638–650, 2020. (in japanese).
[10] K. Urazoe, N. Kuroki, Y. Kato, S. Ohtani, T. Hirose, and M. Numa, "Improvement of luminance isotropy for convolutional neural networks-based image super-resolution," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E103-A, no.7, pp.955–958, 2020.
[11] V.K. Ha, J.-C. Ren, X.-Y. Xu, S. Zhao, G. Xie, V. Masero, and A. Hussain, "Deep learning based single image super-resolution: A survey," International Journal of Automation and Computing, vol.16, no.4, pp.413–426, 2019.
[12] Z. Wang, J. Chen, and S.C.H. Hoi, "Deep learning for image super-resolution: A survey," arXiv: abs/1902.06068, 2019.
[13] J. Yang, J. Wright, T.S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing, vol.19, no.11, pp.2861–2873, 2010.
[14] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L.A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," Procedings of the British Machine Vision Conference, pp.135.1–135.10, 2012.
[15] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," Proceedings of the 7th International Conference on Curves and Surfaces, vol.6920, pp.711–730, 2012.
[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," Proceedings Eighth IEEE International Conference on Computer Vision (ICCV), 2001.
[17] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," Multimedia Tools Applications, vol.76, no.20, pp.21811–21838, 2017.
[18] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa, "Object detection for comics using manga109 annotations," arXiv: abs/1803.08670, 2018.
[19] M. Hradiš, J. Kotera, P. Zemčík, and F. Šroubek, "Convolutional neural networks for direct text deblurring," Proceedings of the British Machine Vision Conference, pp.6.1–6.13, 2015.
[20] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," IEEE Transactions on Multimedia, vol.21, no.12, pp.3106–3121, 2019.
[21] R. Caruana, "Multitask learning," Machine Learning, vol.28, no.1, pp.41–75, 1997.
[22] Y. Zhang and Q. Yang, "A survey on multi-task learning," arXiv: abs/1707.08114, 2017.
[23] A.H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task CNN model for attribute prediction," IEEE Transactions on Multimedia, vol.17, no.11, pp.1949–1959, 2015.
[24] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, "Structure-preserving image super-resolution via contextualized multitask learning," IEEE Transactions on Multimedia, vol.19, no.12, pp.2804–2815, 2017.
[25] M.S. Rad, B. Bozorgtabar, C. Musat, U.V. Marti, M. Basler, H.K. Ekenel, and J.-P. Thiran, "Benefiting from multitask learning to improve single image super-resolution," Neurocomputing, vol.398, pp.304–313, 2020.
[26] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-stuff: Thing and stuff classes in context," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[27] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," Proceedings of the 27th International Conference on International Conference on Machine Learning, pp.807–814, 2010.
[28] M. Lin, Q. Chen, and S. Yan, "Network in network," 2nd International Conference on Learning Representations, ICLR, 2014.
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol.15, pp.1929–1958, 2014.
[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems 27, pp.3320–3328, 2014.
[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," Proceedings of the 22nd ACM International Conference on Multimedia, pp.675–678, 2014.
[32] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," International Conference on Learning Representations (ICLR), 2015.
[33] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

Kazuya Urazoe received the B.E. and M.E. degrees in Electrical and Electronic Engineering from Kobe University in 2018 and 2020, respectively. He is now with Panasonic Corporation. His research interests include digital image processing and machine learning.

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

Yu Kato received the B.E. and M.E. degrees in Electrical and Electronic Engineering from Kobe University in 2015 and 2017, respectively. He joined EIZO Corporation in 2017. He received the Ph.D. degree from Kobe University in 2019. His research interests include digital image processing.

Shinya Ohtani received the B.E. and M.E. degrees in Electrical and Electronic Engineering from Kobe University in 2014 and 2016, respectively. He joined Toyota Motor Corporation in 2016. His research interests include robot vision.

Tetsuya Hirose received the B.S., M.S., and Ph.D. degrees from Osaka University in 2000, 2002, and 2005, respectively. From 2005 to 2008, he was a Research Associate with the Department of Electrical Engineering, Hokkaido University. From 2008 to 2019, he was an Associate Professor with the Department of Electrical and Electronics Engineering, Kobe University. Since 2019, he has been a Professor with the Division of Electrical, Electronic and Information Engineering, Graduate School of Engineering, Osaka University. His current research interests include extremely low-voltage and low-power analog/digital mixed-signal integrated circuit design and smart sensor systems. Dr. Hirose is a member of the IEICE, JSAP, and IEEE.

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