IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 2, 2019, accepted September 16, 2019, date of publication September 23, 2019,
date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943155

Local Excitation Network for Restoring a JPEG-Compressed Image

SONGHYUN YU AND JECHANG JEONG, (Member, IEEE)
Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Jechang Jeong (jjeong@hanyang.ac.kr)

This work was supported in part by the Research Fund of Signal Intelligence Research Center supervised by the Defense Acquisition Program Administration, and in part by the Agency for Defense Development, South Korea.

ABSTRACT Joint photographic experts group (JPEG) compression is lossy compression, and degradation of image quality worsens at high compression ratios. Therefore, a reconstruction process is required for a visually pleasant image. In this paper, we propose an end-to-end deep learning architecture for restoring JPEG images with high compression ratios. The proposed architecture changes a core principle of the squeeze and excitation network for low-level vision tasks where pixel-level accuracy is important. Instead of extracting global features, our network extracts locally embedded features and fine-tunes each feature value by using depthwise convolution. To reduce the computational complexity and parameters with large receptive fields, we use a combination of the recursive structure and feature map down- and up-scaling processes. We also propose a compact version of the proposed model by decreasing the number of filters and simplifying the network, which has about one-twentieth of the parameters of the baseline model. Experimental results reveal that our network outperforms conventional networks quantitatively, and the restored images are clear with sharp edges and smooth blocking boundaries. Furthermore, the compact model shows higher objective results while maintaining a low number of parameters. In addition, at a high compression ratio, the overall information, including details in the blocks, are lost owing to high quantization errors. We apply a generative adversarial network structure to restore these highly damaged blocks, and the results reveal that the image produced has details similar to those of the ground truth.

INDEX TERMS Convolutional neural network, JPEG image restoration, generative adversarial network.

I. INTRODUCTION

JPEG compression is a popular standard for still image compression, and it is a lossy compression technique due to the quantization of the discrete cosine transformation (DCT) coefficients. Lossy compression has a much higher compression ratio than lossless compression; however, it damages image parts that are relatively less sensitive to the human eye. A compressed image with a low compression ratio may be difficult for the human eye to detect, but an image with a high compression ratio might be seriously distorted. JPEG compression causes a serious degradation in image quality when the quantization step size is increased. Because in JPEG compression, DCT and quantization are performed in a block unit, blocking artifacts occur at the block boundary and ringing artifacts appear at the edge of the object. Furthermore, at high compression ratios, posterization occurs in the flat region, block information is completely blurred, or directional patterns occur within the block (see Fig. 1). Many deblocking algorithms [1]–[5] have been proposed to reduce blocking artifacts; however, the ability of simple deblocking alone to restore damaged images with high compression ratios is limited. Some methods have been proposed to restore JPEG images. Liu *et al.* [6] proposed sparsity-based dual-domain DCT, and Goto *et al.* [7] reduced compression artifacts using total variation regularization. Nosratinia [8] proposed enhancement of the compressed image by re-application of JPEG compression, and Jancsary *et al.* [9] proposed a method using regression tree fields. Wang *et al.* [10] improved the performance of a JPEG image restoration using deep sparse coding networks.

In recent years, convolutional neural networks (CNNs) have been used successfully in image classification [11], [12], super resolution [13], [14], image denoising [15], [16],

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

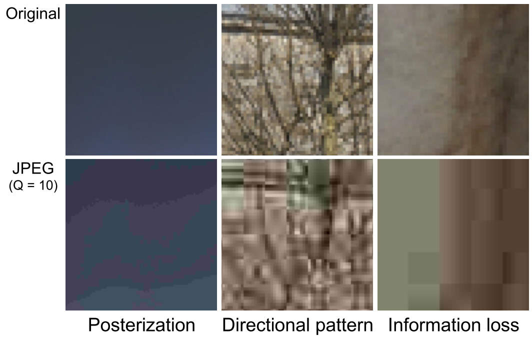


FIGURE 1. Examples of JPEG artifacts at a high compression ratio. At a high compression ratio, in addition to blocking artifacts, many kinds of artifacts occur due to the loss of detail information.

optical flow estimation [17], image dehazing [18] and other image restoration areas. CNNs have also significant improved JPEG image restoration. The ARCNN [19], a compact network with four convolution layers, was one of the initial networks for JPEG artifacts removal, and Svoboda *et al.* [20] used residual learning [12] and edge-loss to improve performance. The DnCNN [21] used global residual learning (GRL) and successfully trained a deeper network with a depth of 20 to enhance performance, and MemNet [15] adopts a recursive unit that reuses the weights of the convolution layer several times and a densely-connected block structure, thereby improving the performance while maintaining a few parameters. These methods train the network using a loss function based on the l_2 norm, which yeilds a high peak signal-to-noise ratio (PSNR); however, the resulting image tends to be blurred. SRGAN [22] applies a generative adversarial network (GAN) [23] composed of a generator and discriminator to super-resolution to avoid blurring and therefore, to obtain a more realistic super-resolved image.

In this paper, we propose a post-processing network that restores low-quality images with a high quantization error, focusing on JPEG compressed images with a high compression ratio. Inspired by [24] and [15], we adopt the recursive unit as the basic structure of the network that iteratively uses the same parameters. The structure of the proposed recursive unit is based on the block structure of the SE-ResNet proposed in the squeeze and excitation network (SENet) [25]. While SENet uses global pooling to extract globally embedded spatial information, our network uses the depthwise convolution [26] to extract locally embedded spatial information and perform point-wise multiplication, so that detailed information for each feature is stored in parameters. The proposed network is named a local excitation network for JPEG restoration (LEJR). We also propose a compact version of an LEJR by greatly reducing the number of parameters, which shows superior performance over the number of parameters compared to conventional networks. In addition, because the image we are interested in is a low-quality image with a significant amount of information lost, the result image would be blurred if the network is trained based on l_2 loss. To better restore the detail, we apply GAN structure to the JPEG restoration by adding a discriminator to the LEJR. GAN-based LEJR is trained using perceptual loss function resulting in more realistic images than LEJR trained on l_2 loss. The main contributions of this paper are summarized as follows:

- State-of-the-art performance by suggesting a new module that modifies SENet for low-level vision applications.
- A good trade-off between the memory usage and receptive field by using a combination of the recursive block and down-up sampling structure.
- Maximization of the efficiency between computational complexity and performance by proposing a compact version of the network.
- Application of the GAN structure to remove JPEG artifacts to restore more realistic images at a high compression ratio.

II. RELATED WORKS

In this section, SENet [25], SRGAN [22], ARCNN [19], DnCNN [21], and MemNet [15] are briefly reviewed.

A. SENET

SENet is proposed for image classification. It upgrades the existing network by adding a squeeze and excitation (SE) unit to the block. The SE unit is connected to ensure that the conventional CNN does not have limited receptive fields according to the depth of each network. The SE unit is divided into two stages, squeeze and excitation. Squeeze uses global average pooling to create channel-wise descriptors. Excitation uses a fully connected layer, rectifier linear unit (ReLU), and sigmoid activation to extract the final feature, and each descriptor is multiplied by the input features. SENet enhances the performance of image classification with a small amount of additional computational complexity and fewer parameters compared to the existing network.

Cheng *et al.* [27] applied SENet to super-resolution and obsevered that it resulted in a performance improvement compared to the existing ResBlock. However, when SENet is applied to the proposed network, it has lower performance than ResBlock. Therefore, we propose a local excitation block (LEB) that modifies SE-ResBlock, improving its performance in JPEG artifact reduction.

B. SRGAN

SRGAN is a method that applies the GAN structure to single image super-resolution. To ensure that the resulting image is not blurred when the network is trained using the l_2 loss function, SRGAN uses perceptual loss function by training a generator with a discriminator which decides the reconstructed image is real or not. After pretraining the generator, the whole network is trained using perceptual loss, combining feature loss using VGGNet [11] and adversarial loss of discriminator with a weighting parameter. As a result, the PSNR value of the SRGAN is lower than that of the network trained with l_2 loss, but more visually pleasant images can be obtained.

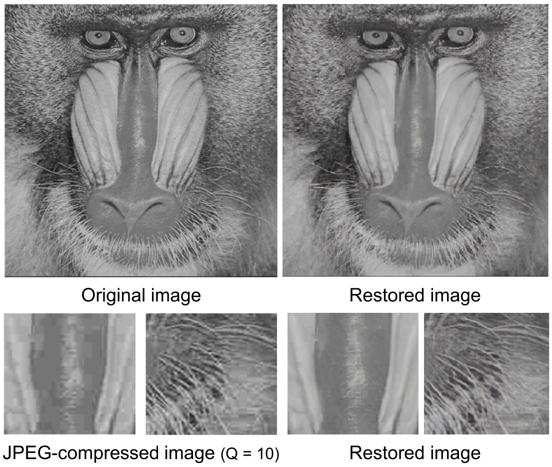

Original image | Restored image
JPEG-compressed image (Q = 10) | Restored image

FIGURE 2. Restoration result of the proposed method. Our GAN-based network not only eliminates bloc-king artifacts but also restores details at a high compression ratio.

Inspired by SRGAN, we apply the GAN structure to the restoration of JPEG compressed images. Unlike SRGAN, the structure of the discriminator and the weight in perceptual loss are changed. As can be seen in Fig. 2, the restored image is quite similar to the original image at low quality factors.

C. DEEP LEARNING FOR JPEG DEBLOCKING

In this subsection, we briefly introduce three deep-learning papers for JPEG artifacts removal.

ARCNN is one of the earliest researches using CNN for JPEG artifact removal, which consists of four convolution layers: feature extraction, feature enhancement, mapping, and reconstruction. The network is trained using l_2 loss. ARCNN has very few parameters, but performance is limited, because the network is very shallow.

DnCNN performs three tasks: Gaussian noise reduction, JPEG artifacts removal, and super-resolution, with the same network structure. It differs from ARCNN as it constructs a network with deeper depth (20 convolution layer) and uses GRL to train only the residuals of input and output, thereby improving both the training stability and performance of the deep model. It is also trained using l_2 loss.

MemNet utilizes recursive structure, local residual learning, and block-based densely connected structures to maximize the gradient flow of errors and reuse feature maps. Although the network depth is 80, the number of parameters is similar to that of DnCNN which uses 20 convolution layers owing to the recursive structure. In addition, the performance is further enhanced by using a multi-supervised loss function that takes into account not only the final output but also the local outputs of each block.

III. PROPOSED NETWORK

This section describes the structure and training method of the proposed network.

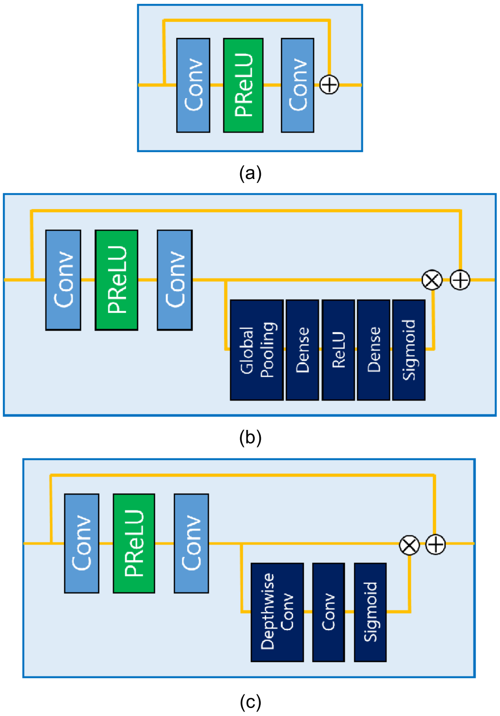


FIGURE 3. Comparison of three block structures. (a) ResBlock, (b) SE-ResBlock, and (c) Proposed block (LEB). Instead of using global pooling in SENet, our block uses depthwise convolution to extract local features.

A. NETWORK ARCHITECTURE

We propose two models: baseline model, LEJR, and LEJR_compact, a compact version of LEJR that reduces the number of parameters to about one-twentieth of the baseline model. Fig. 4 shows the overall structure of the proposed LEJR network. The proposed network consists of feature extraction & down-sampling, recursive block, reconstruction, and up-sampling.

1) FEATURE EXTRACTION & DOWN-SAMPLING

First, LEJR extracts 256 features through two convolution layers and two parametric rectifier liniear units (PReLU) [28], and the second convolution down-samples the feature map by 1/2 in each direction with stride 2. Down-sampling reduces the size of the feature map by a factor of four, thus reducing the computational complexity of the recursive block to one-fourth, which accounts for most of the network's overall computation volume, thus enabling much faster training. However, as down-sampling of feature maps may cause information loss, we use a global skip connection. It reduces information loss by predicting only the residual values while keeping the original input signal intact. When the input of the network is x_0, feature extraction & down-scaling can be expressed as the following equations:

$$H_0 = P_2\left(W_2 * P_1(W_1 * x_0 + B_1) + B_2\right), \tag{1}$$

$$P_i\left(x\right) = \max\left(x, 0\right) + a_i \cdot \min\left(0, x\right), \tag{2}$$

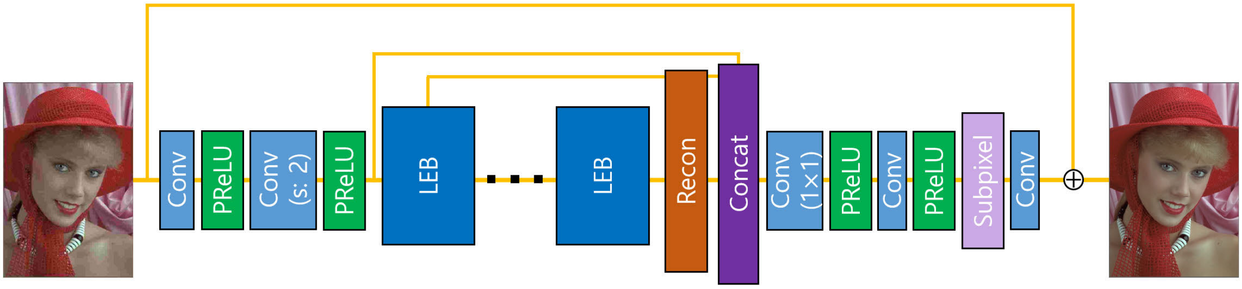


FIGURE 4. Architecture of the proposed network (baseline model).

where $P_i(\cdot)$ is the i-th PReLU activation function and a_i is a weight. W_i means the i-th weights of the convolution layer, B_i is bias. In the second convolution W_2, down-scaling is performed with the stride set to 2. The extracted feature H_0 becomes the input of the subsequent recursive block.

2) RECURSIVE BLOCK

Inspired by [24] and [27], we use a recursive structure to reuse weights to maximize performance efficiency. A recursive block is shown in Fig. 3(c). In the existing SE-ResBlock in Fig. 3 (b), instead of using global average pooling, which reduces the size of the feature map to 1×1, our network uses depthwise convolution to extract descriptors with the same size as the input feature map containing local information with a small number of additional computations and parameters. Then, the network considers the correlation between feature maps through a 1×1 convolution, and after sigmoid activation, which maps the signal range from 0 to 1, descriptors are point-wisely multiplied by input features. Each feature has its own descriptor, which implicates a local feature; thus it is possible to fine-tune the feature values. We named this process local excitation (LE), and it is formulated as follows:

$$L(x) = \sigma(W_c * (W_{DC} * x + B_{DC}) + B_c) \cdot x, \tag{3}$$

where $L(\cdot)$ means the LE function, x is the input feature map of the function, W_{DC} and B_{DC} are the weights and bias of depthwise convolution layer, respectively, W_c and B_c are the weights and bias of 1×1 convolution layer, respectively. $\sigma(\cdot)$ denotes the sigmoid activation function. Using Equation (3), the N-th recursive block is formulated as follows:

$$H_N = L(W_4 * P_3 (W_3 * H_{N-1} + B_3) + B_4), \tag{4}$$

where H_N is the output of the N-th recursive block, and it uses the previous output, H_{N-1}, as its input.

3) RECONSTRUCTION

To make full use of the local output, the outputs in each block go through a single reconstruction network, which is composed of one 3×3 convolution layer and PReLU. All the outputs of the reconstruction are concatenated, and the feature map is compressed through a 1×1 convolution layer, which acts as a bottleneck layer.

4) UPSAMPLING

The features down-scaled in the early part of the network are up-sampled. For the upsampling layer, we use a sub-pixel layer [29] to reduce network parameters, computational complexity, and information loss. The layer performs upscaling by rearranging the shape of the feature map without additional parameters. If the down-scaling factor is set to 2, the network can reduce the computational complexity to 1/4 of that of a network without down-up scaling. Then, the convolution layer adjusts the number of feature maps to three (RGB), and the final output image is obtained by adding the input image for GRL. Unlike existing networks, which handle only luminance images, the proposed network outputs an RGB image by receiving an RGB input.

5) COMPACT MODEL

Although the baseline model uses recursive blocks, as it uses 256 filters for each convolution layer, the total number of parameters is approximately 3,000K, which is larger than that of existing networks (Table 4). We propose a compact version of the LEJR, which has much fewer parameters. Fig. 5 shows the proposed compact network. It reduces the number of filters at all convolution layers to 64 and removes additional layers after the concatenation. Furthermore, the compact model does not use down-scaling of the feature map; this is covered in detail in Section 3.2. As a result, the network has one-twentieth of the parameters of the baseline model.

6) GAN STRUCTURE

Because a JPEG image that is compressed at a high compression ratio has serious degradation, based on SRGAN, we apply the GAN structure to JPEG image restoration. A baseline model is used as a generator and is trained with the discriminator in Fig. 6, which determines whether the image is a generated image or real image. The generator is first trained using the l_1 loss function and then further trained using perceptual loss, which combines the VGG-19 [11] based content loss with the adversarial loss of the discriminator, as in SRGAN. Detailed training methods are described

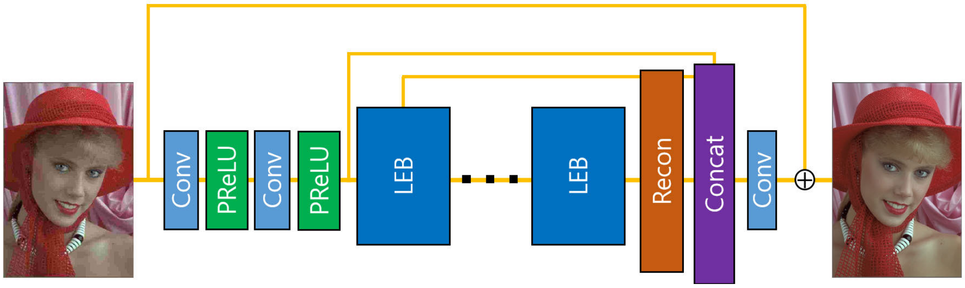


FIGURE 5. Architecture of the proposed network (compact model).

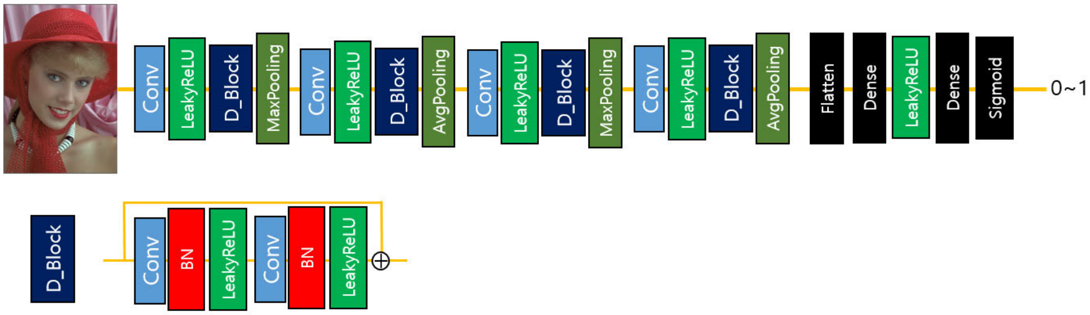


FIGURE 6. Architecture of the proposed discriminator.

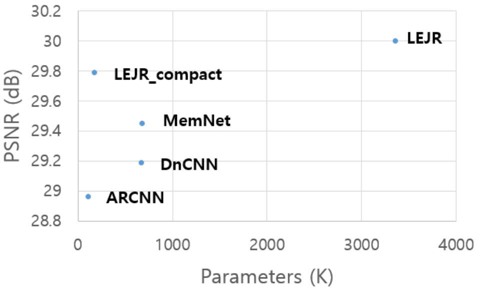


FIGURE 7. Parameters and PSNRs (dB) of each method (LIVE1 dataset is used with Q = 10).

in Section 3.3. Fig. 9 compares the output images of the GAN structure with the baseline model. In a low-quality image with Q = 10, the GAN-based model better restores the details of the image than the baseline model.

B. NETWORK ANALYSIS

In this section, experimental results on the structures of the network and the setting of recursion depth are explained.

Extracting a global feature and multiplying it to existing features is a core principle of SENet. Because our model directly links input and output images using a global skip connection and predicts only residual values, global tuning of feature values is not effective. Because of the characteristics of low-level vision tasks where pixel-level accuracy is important, our model extracts point-wise features through depth-wise convolution and multiplies them to the existing features to fine-tune each individual feature value. To demonstrate the superiority of our block structure, we experimented our network using three different block structures, and Fig. 3 shows the structures of each block. Figures 3(a), 3(b), and 3(c) are the basic ResBlock, SE-ResBlock, and proposed block, respectively. Table 1 compares the number of parameters and the PSNR using the three blocks. Compared to the ResBlock, the SE-ResBlock increases the number of parameters by 0.2%, but the PSNR is rather lower for JPEG artifact reduction application. However, compared with ResBlock, the proposed block has a 2% increase in the number of parameters and a 0.03 dB increase in the PSNR. These results indicate that our local excitation block is more effective in removing JPEG artifacts than are other existing blocks developed in high-level vision tasks.

TABLE 1. Block structure comparison (LIVE1 dataset is used with Q = 10).

Structure	PSNR (dB)	Parameters (ratio)
ResBlock	29.94	1.000
SE-ResBlock	29.92	1.002
Proposed	29.97	1.021

Table 2 shows the effect of GRL and feature map down-up scaling in the baseline model and compact model. For the baseline model, both GRL and down-scaling improve performance. The baseline models use 256 features in each convolution layer and use down-up scaling and GRL simultaneously to extract features at various scales like U-Net [30].

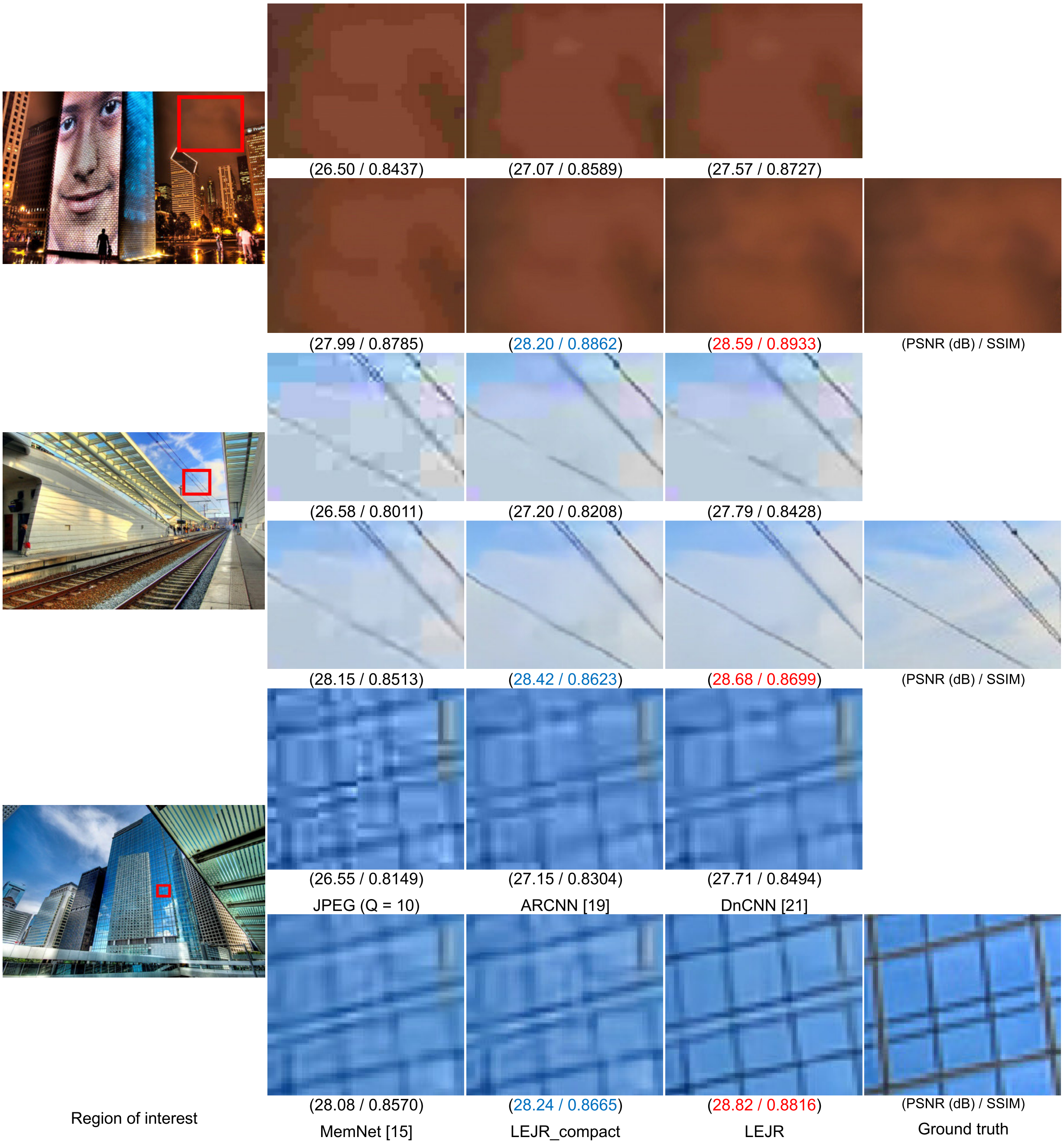


FIGURE 8. Restoration results of "img76", "img69", and "img61" in Urban100 with Q = 10. The image in the top row has a posterization artifact. Compared with other models, our model restores smooth continuity of the color, which leads to a large improvement in the PSNR.

It also prevents the features from blurring in the upscaling by using the subpixel layer instead of deconvolution layer. Furthermore, as in JPEG image restoration, similarity of input and output is very high. Like in SR, the network using GRL can predict only the residual values without storing the information of an input image. However, in the compact model, the performance degrades significantly with down-scaling (29.77 dB → 29.71 dB). This is probably because the compact model uses relatively small feature maps (64), and the process of enhancing the results after the subpixel upscaling layer is omitted.

Table 3 shows the PSNR and training time according to the recursion depth of the baseline model. The performance is 0.02 dB higher when the depth is 6 or 8 than when the depth is 4, but the training time increases 36% and 68%, respectively. Furthermore, although the proposed network uses a recursive block, it concatenates all outputs of reconstruction so that the number of network parameters gradually increases

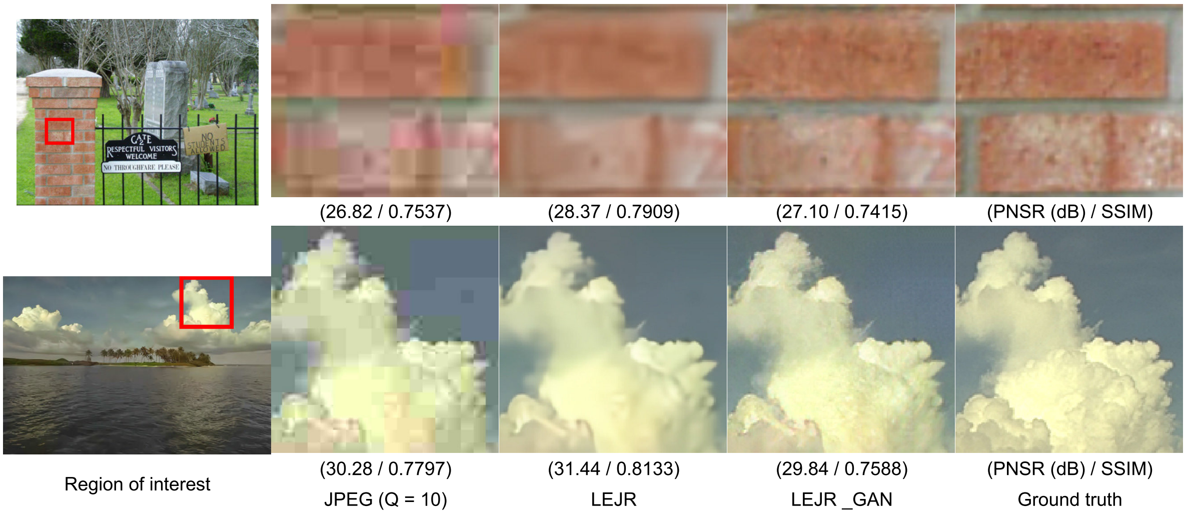


FIGURE 9. LEJR vs. LEJR_GAN ("cemetery" and "ocean" in LIVE1 with Q = 10). The GAN-based model has low PSNRs but restores the texture information to produce a more realistic image.

TABLE 2. Effect of GRL and down-up scaling (PSNR (dB)) (LIVE1 dataset is used with Q = 10).

		Down-up O	Down-up X
LEJR	GRL O	29.97	29.94
	GRL X	29.95	29.93
LEJR_ compact	GRL O	29.71	29.77

TABLE 3. Performance of the baseline model at each recursion depth (LIVE1 dataset is used with Q = 10).

Depth	PSNR (dB)	Training time (ratio)
4	29.97	1.00
6	29.99	1.36
8	29.99	1.68

TABLE 4. Performance of different training strategies (LIVE1 dataset is used with Q = 10).

Test / Train	RGB	YCbCr	Y
RGB	**27.58**	**33.46**	30.00
YCbCr	27.47	33.37	29.90
Y	-	-	**30.05**

as depth increases. Therefore, we set the recursion depth to 4 in the proposed baseline model.

There are many options to the select color channel for training. Table 4 lists the PSNR comparison of different color channels. We found that the performance of the luminance component is the highest when the model is trained only using luminance, and training together with YCbCr could reduce the performance. Additionally, in both RGB and YCbCr color spaces, the model trained with the RGB color channel outperforms the model trained with the YCbCr color channel also. The results reveal that although JPEG performs compression in the YCbCr color space, training together with YCbCr degrades the model performance because the characteristics of Y and Cb, Cr channels are significantly different. Meanwhile, RGB channels have high correlation among channels, which yields improved performance in color image processing. To improve the performance of the luminance component, it is better to train only the luminance images; however, this requires additional processing on the Cb and Cr channels and is impractical in the real world, where most images have color components. Therefore, we use the RGB space for practical color image processing.

C. TRAINING

In this section, we discuss the training loss for each proposed network.

1) TRAINING CNN MODEL

In the image restoration task, it is suggested that the model trained with the l_1 loss function shows a better performance in terms of PSNR than with the l_2 loss. Moreover, there are additional performance enhancements when using the two loss functions together [31]. As the removal of JPEG artifacts is part of image restoration leads to fast convergence and improved performance than that tasks, using l_1 loss in our experiment on JPEG restoration by using l_2 loss. Therefore, our CNN-based models, the baseline model and compact model, are trained with the l_1 loss function and then re-trained with the l_2 loss function. These loss functions are given by:

$$l_1(\theta) = \frac{1}{N}\sum\nolimits_{i=1}^{N} |F(x_i;\theta) - y_i|, \tag{5}$$

$$l_2(\theta) = \frac{1}{N}\sum\nolimits_{i=1}^{N} |F(x_i;\theta) - y_i|^2, \tag{6}$$

where N is batch size, $F(\cdot)$ is the network function with learnable parameters θ, and x_i and y_i denote patch pairs of the JPEG image and ground truth in the training data. When the network is fine-tuned with the l_2 loss function, the PSNR increases by a maximum of 0.03 dB compared to training with only the l_1 loss function.

2) TRAINING THE GAN MODEL.

To train the GAN-based model, the perceptual loss function, which combines the content loss and adversarial loss functions, is used, as described in Section 3.1. The loss functions are

$$l_{content}(\theta) = \frac{1}{N}\sum_{i=1}^{N} |\varphi_{5,4}(F(x_i;\theta)) - \varphi_{5,4}(y_i)|^2, \quad (7)$$

$$l_{adversarial}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log D(F(x_i;\theta);\theta_D), \quad (8)$$

$$l_{perceptual}(\theta) = l_{content}(\theta) + 0.1 \cdot l_{adversarial}(\theta), \quad (9)$$

where $l_{content}(\cdot)$ and $l_{adversarial}(\cdot)$ are content loss and adversarial loss functions, respectively, and $l_{perceptual}(\cdot)$ is the perceptual loss function. As in [22], the feature map after the fifth convolution before the fourth maxpooling of VGG-19 is used for content loss, and it is expressed as $\varphi_{5,4}(\cdot)$ in Equation (7). $D(\cdot)$ in Equation (8) represents the discriminator function. After the generator (baseline model) is pre-trained with the l_1 and l_2 loss functions, it is trained with a discriminator.

IV. EXPERIMENTAL RESULTS

This section explains the dataset and implementation details, and compares the performance of the proposed model with those of the state-of-the-art models.

A. DATASETS

We used the DIV2K dataset [32] consisting of 800 2K resolution training images and 100 validation images for training. For comparison, LIVE1, B100 [33], and Urban100 [34] datasets were used for the test. The original images were compressed with quality factors 10, 20, 30, and 40 using the MATLAB JPEG encoder. A low quality factor means a high compression ratio.

B. IMPLEMENTATION DETAILS

Our baseline model has four recursions and 256 filters at each convolution layer. All convolution layers except the bottleneck layer and the convolution layer of LE using a 1×1 kernel use 3×3 kernels. The Compact model has 10 recursion depths, and uses 64 filters with a 3×3 kernel. Our networks accept RGB images as inputs and output RGB images, and because the network is fully convolutional, arbitrary image sizes can be processed when testing. For training, the image is divided into 80×80 size patches, and 16 patch pairs constitute one batch. As the JPEG encoder uses a transform of 8×8 pixels, we set the stride to 79 to consider the various cases of block boundaries when making training data. The training data is augmented by random flip and rotation, and the Adam is used as the optimizer. The CNN-based model is initially trained at an initial learning rate of $1e-4$ using the l_1 loss function and re-trained using the l_2 loss function with an initial learning rate of $1e-5$. When training the GAN model, both the generator and discriminator are trained with the Adam optimizer with an initial learning rate of $1e-4$. One epoch is made up of approximately 43,000 batches, and the CNN-based model is trained for 500,000 iterations with the l_1 loss function, and re-trained for 100,000 iterations with the l_2 loss function. The proposed GAN model is trained for 100,000 iterations by alternately training the generator and discriminator. The proposed model was implemented using Keras [35], and it took approximately 30 h to train the baseline model using Geforce GTX 1080Ti. Each model was trained for each quality factor.

C. COMPARISONS WITH STATE-OF-THE-ART MODELS

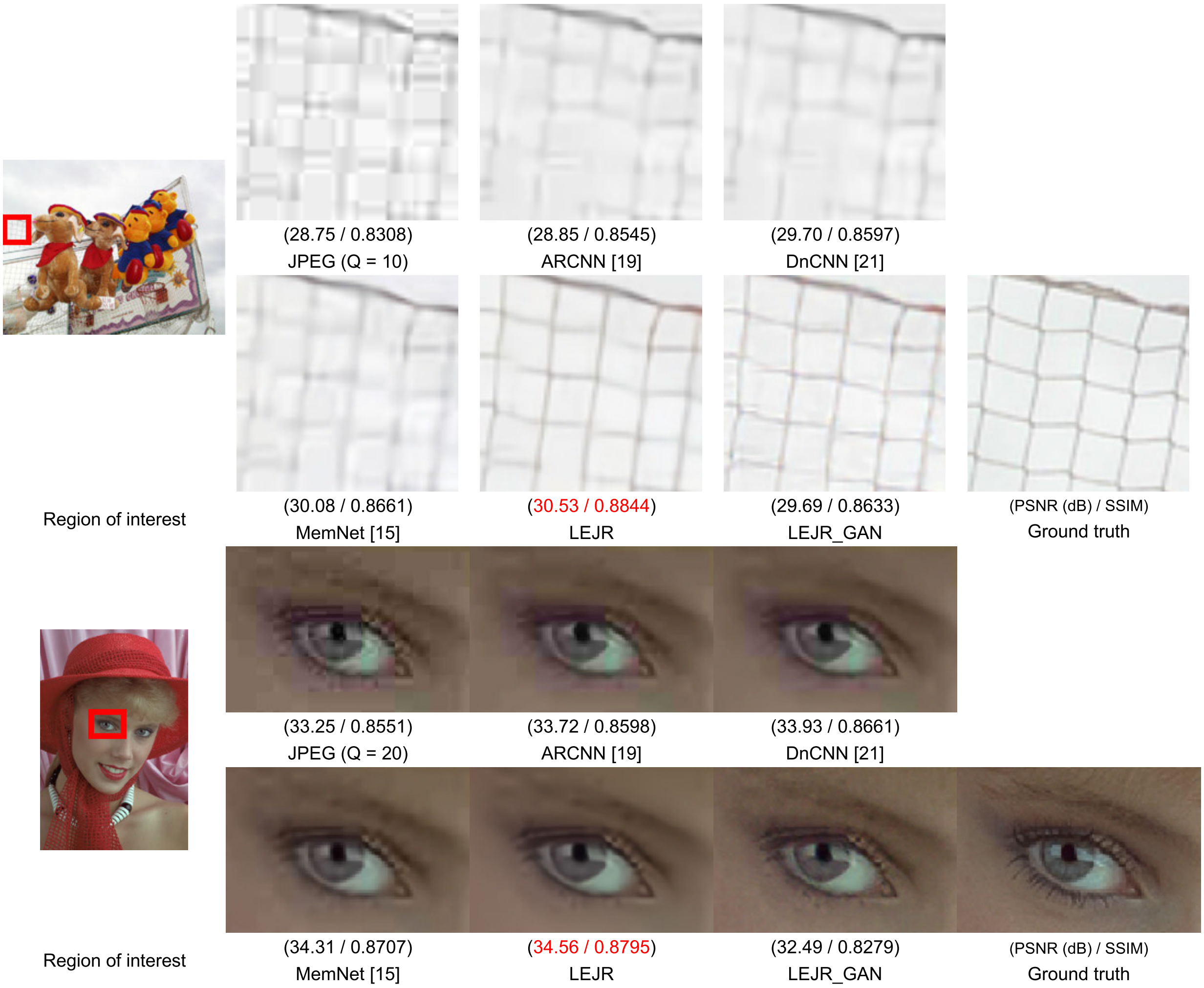
The proposed network was compared with an ARCNN [19], a DnCNN [21], and a MemNet [15]. The ARCNN and DnCNN were experimented on using MATLAB public code, and MemNet was reproduced by us using Keras. The network structure of the reproduction version is the same as that of MemNet, but ours uses an RGB input and output instead of a gray-scale image, and it was trained using an l_1 loss function instead of an l_2 loss function.

Table 5 shows comparison results using the PSNR, structural similarity (SSIM), and the number of parameters, where $*$ indicates our reproduction results. The LIVE1, B100, and Urban100 datasets were used with quality factors 10, 20, 30, and 40. In all datasets and for all quality factors, the proposed baseline model (LEJR) shows the highest objective indicator values, and the proposed compact model has the second best result. Although other methods process only the luminance component and our model processes the RGB image with a single model, LEJR and LEJR_compact models significantly outperform the existing models in respect of the luminance PSNR. Especially, LEJR shows superior results in Urban100, which consists of images with a lot of complex regions and edges. The LIVE1 dataset consists of 29 diverse RGB images, and the LEJR_compact model shows an improved PSNR over 0.3 dB than MemNet with about four times less parameters. The LEJR_compact model has a higher number of parameters than the ARCNN, but it demonstrates good performance while maintaining a small number of parameters compared to the latest very deep models. Fig. 7 shows the number of parameters and PSNR values for each method.

Fig. 8 compares the result images of LEJR and LEJR_compactc with those of the conventional methods. The first row in Fig. 8 depicts the restored results of severely damaged image by posterization. JPEG images are subject to severe posterization at a low quality factor, and conventional methods cannot restore them well. By contrast, our LEJR model reconstructs a smooth image, which is hard to distinguish from the ground truth. In second and third rows, compared to other methods where the images are severely

TABLE 5. Average PSNR/SSIM for quality 10, 20, 30, and 40 on datasets LIVE1, B100, and Urban100. The last row represents the number of parameters of each network. Red font indicates the best performance, blue font indicates the second-best performance, and ∗ indicates our reproduction results.

Dataset	Q	JPEG		ARCNN [19]		DnCNN [21]		MemNet [15]		LEJR		LEJR _compact		LEJR_GAN	
LIVE1	10	28.36	0.7937	28.96	0.8076	29.19	0.8123	29.45 (29.55*)	0.8193 (0.8229*)	29.98	0.8361	29.78	0.8306	28.74	0.7958
	20	30.62	0.8653	31.29	0.8733	31.59	0.8802	31.83 (31.90*)	0.8846 (0.8853*)	32.33	0.8954	32.11	0.8917	30.41	0.8501
	30	31.41	0.9000	32.67	0.9043	32.98	0.9090	33.29*	0.9128*	33.72	0.9202	33.56	0.9183	32.25	0.8930
	40	32.35	0.9173	33.63	0.9198	33.96	0.9247	34.30*	0.9273*	34.71	0.9339	34.57	0.9318	32.07	0.8888
B100	10	28.13	0.7701	28.74	0.7781	28.84	0.7826	29.13*	0.7939*	29.52	0.8045	29.34	0.8017	28.21	0.7724
	20	30.25	0.8475	30.82	0.8504	31.05	0.8573	31.31*	0.8633*	31.66	0.8721	31.50	0.8701	30.27	0.8438
	30	31.52	0.8814	32.14	0.8857	32.36	0.8905	32.62*	0.8948*	32.96	0.9021	32.85	0.9006	31.45	0.8692
	40	32.43	0.9010	33.00	0.9041	33.27	0.9090	33.58*	0.9122*	33.89	0.9189	33.74	0.9171	31.36	0.8667
Urban100	10	26.99	0.8102	28.06	0.8373	28.54	0.8487	28.83*	0.8548*	29.74	0.8746	29.29	0.8656	28.55	0.8420
	20	29.22	0.8725	30.30	0.8899	31.01	0.9022	31.35*	0.9055*	32.17	0.9179	31.75	0.9127	30.12	0.8756
	30	30.69	0.9014	31.94	0.9174	32.47	0.9248	32.93*	0.9288*	33.76	0.9380	33.37	0.9342	32.16	0.9173
	40	31.81	0.9186	32.80	0.9297	33.50	0.9376	34.12*	0.9412*	34.83	0.9483	34.34	0.9445	32.46	0.9195
Parameters		-		107K		671K		677K		3,357K		173K		-	



(28.75 / 0.8308)
JPEG (Q = 10)

(28.85 / 0.8545)
ARCNN [19]

(29.70 / 0.8597)
DnCNN [21]

Region of interest

(30.08 / 0.8661)
MemNet [15]

(30.53 / 0.8844)
LEJR

(29.69 / 0.8633)
LEJR_GAN

(PSNR (dB) / SSIM)
Ground truth

(33.25 / 0.8551)
JPEG (Q = 20)

(33.72 / 0.8598)
ARCNN [19]

(33.93 / 0.8661)
DnCNN [21]

Region of interest

(34.31 / 0.8707)
MemNet [15]

(34.56 / 0.8795)
LEJR

(32.49 / 0.8279)
LEJR_GAN

(PSNR (dB) / SSIM)
Ground truth

FIGURE 10. Restoration results of "carnivaldolls and womanhat" in LIVE1 dataset with Q = 10 and 20.

damaged, the LEJR_compact restores the edges well, and the restored images obtained by applying the LEJR are highly close to the ground truth. The CNN-based models trained using the l_1 loss function have high PSNR and SSIM values; however, the resulting image is blurred. However, although the proposed GAN-based network has low objective indicators, it shows clearer images that have more detail information (see Fig. 9). Fig. 10 compares the LEJR and LEJR_GAN models with conventional models. The LEJR model demonstrates a good deblocking performance while preserving sharp edges, and the LEJR_GAN model gives more realistic restored images.

As our LEJR utilizes a combination of the recursive structure and down-up strategy, it has a large receptive field while maintaining the number of parameters and GPU memory consumption low. Conventional networks use relatively small patch sizes for training: 24×24 in ARCNN [19], 40×40 in DnCNN [21], and 31×31 in MemNet [15]. By contrast, our LEJR uses sufficiently large size patches (80×80) to take advantage of the large receptive field. Consequently, owing to the efficient structure of LEJR, the large receptive field and large size patch contributed to reduce posterization artiracts, thereby resulting in a significant improvement in objective scores.

V. CONCLUSION

In this paper, deep convolutional neural networks were proposed for restoring JPEG-compressed images. We have proposed a baseline model (LEJR) that maximizes objective performances, a compact model (LEJR_compact) that greatly reduces the number of parameters, and successfully applied a GAN-based model (LEJR_GAN) that gives restored images which are more realistic. A recursive structure was used to improve performance against the number of parameters, and a local excitation unit was developed by modifying the existing SE-ResBlock to fit the JPEG artifact reduction. The LEJR used the feature map down-up scaling strategy to speed up the training and testing time apart from enlarging the receptive field by reducing the computational complexity and memory usage. For practical usage, the proposed network trains RGB images end-to-end to obtain RGB images directly from the network output. Consequently, the proposed baseline model had significantly increased PSNR and SSIM values compared to the existing models, and the proposed compact model recorded higher objective values than the existing models with fewer parameters. In addition, this study suggests that GAN can be used to reconstruct realistic images that have many details in JPEG image restoration with high compression ratios. The proposed network can be used in other image restoration fields such as image dehazing, image denoising, and super-resolution.

REFERENCES

[1] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[2] H. C. Reeve, III and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, no. 1, 1984, Art. no. 230134.

[3] C. Wang, J. Zhou, and S. Liu, "Adaptive non-local means filter for image deblocking," *Signal Process., Image Commun.*, vol. 28, no. 5, pp. 522–530, 2013.

[4] Z. Xiong, M. T. Orchard, and Y.-Q. Zhang, "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 433–437, Apr. 1997.

[5] K. Lee, D. S. Kim, and T. Kim, "Regression-based prediction for blocking artifact reduction in JPEG-compressed images," *IEEE Trans. Image Process.*, vol. 14, no. 1, pp. 36–48, Jan. 2005.

[6] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1395–1411.

[7] T. Goto, Y. Kato, S. Hirano, M. Sakurai, and T. Q. Nguyen, "Compression artifact reduction based on total variation regularization method for MPEG-2," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 253–259, Feb. 2011.

[8] A. Nosratinia, "Enhancement of JPEG-compressed images by reapplication of JPEG," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 27, no. 1, pp. 69–79, Feb. 2001.

[9] J. Jancsary, S. Nowozin, and C. Rother, "Loss-specific training of nonparametric image restoration models: A new state of the art," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 112–125.

[10] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of JPEG-compressed images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 2764–2772.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, Apr. 2015, pp. 1–14.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

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

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

[15] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4539–4547.

[16] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[17] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4161–4170.

[18] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "AOD-Net: All-in-one dehazing network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4770–4778.

[19] C. Dong, Y. Deng, C. C. Loy, and X. Tnag, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 576–584.

[20] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," 2016, *arXiv:1605.00366*. [Online]. Available: https://arxiv.org/abs/1605.00366

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

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

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

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

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018, pp. 7132–7142.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[27] X. Cheng, X. Li, Y. Tai, and J. Yang, "SESR: Single image super resolution with recursive squeeze and excitation networks," 2018, *arXiv:1801.10319*. [Online]. Available: https://arxiv.org/abs/1801.10319

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[29] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1874–1883.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*. [Online]. Available: https://arxiv.org/abs/1505.04597

[31] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

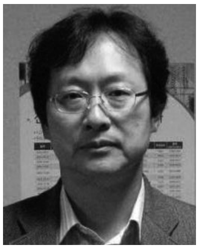
[32] R. Timofte *et al.*, "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1110–1121.

[33] 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," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2015, pp. 416–423.

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

[35] F. Chollet. *Keras (Version 2.0.8)*. (2015). [Online]. Available: https://github.com/fchollet/keras

SONGHYUN YU received the B.S. degree in electronic engineering from Hanyang University, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electronics and computer engineering. His research interests include image processing, deep learning, and image/video compression.



JECHANG JEONG received the B.S. degree in electronic engineering from Seoul National University, South Korea, in 1980, the M.S. degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology, in 1982, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1990. From 1982 to 1986, he was with the Korean Broadcasting System, where he helped to develop teletext systems. From 1990 to 1991, he was a Postdoctoral Research Associate with the University of Michigan, where he helped to develop various signal-processing algorithms. From 1991 to 1995, he was with Samsung Electronics Company, South Korea, where he was involved in the development of HDTV, digital broadcasting receivers, and other multimedia systems. Since 1995, he has been conducting research with Hanyang University, Seoul, South Korea. He has published numerous technical articles. His research interests include digital signal processing, digital communication, and image and audio compression for HDTV and multimedia applications. He received the Scientist of the Month Award from the Ministry of Information and Communication, South Korea, in 1998.