IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2019, accepted January 12, 2020, date of publication February 14, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974001

Locally Adaptive Channel Attention-Based Network for Denoising Images

HAEYUN LEE[1], (Student Member, IEEE), AND SUNGHYUN CHO[2], (Member, IEEE)

[1]Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea
[2]Computer Science Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

Corresponding author: Sunghyun Cho (s.cho@postech.ac.kr)

This work was supported by the Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1801-05.

ABSTRACT Channel attention has recently been proposed and shown a great improvement in image classification accuracy. In this paper, we show that channel attention can greatly help a low-level vision task, image denoising, as well, and propose channel attention-based networks for image denoising. We provide a thorough analysis on the effect of channel attention on image denoising, which shows that channel attention boosts denoising performance by making the network to focus on informative channels more closely related to noise. We also show that channel attention has an adaptive nature to image contents and noise and propose locally adaptive channel attention for further improving image denoising quality. Experimental results show that our denoising network with global channel attention outperforms existing state-of-the-art methods in both blind and non-blind settings, and our locally adaptive channel attention substantially improves both image quality and computation time.

INDEX TERMS Image denoising, deep learning.

I. INTRODUCTION

Image denoising is one of the most fundamental problems in computer vision and image processing fields. A noisy image y is generally modeled as $y = x + n$ where x is a noise-free image, and n is noise, which is often assumed to be additive white Gaussian noise with a standard deviation σ. The goal of image denoising is to infer x from y, which is ill-posed due to the loss of information caused by noise. Image denoising has a wide range of applications such as consumer cameras, medical imaging, and other vision systems that take noisy images as input.

Recently, deep learning-based approaches have shown significant improvement over classical ones [1]–[5]. A key challenge in applying deep learning to denoising is to find an effective network architecture that maximizes denoising performance. As the representational power of a convolutional neural network (CNN) usually increases as does its depth, an intuitive solution would be to use more convolutional layers. However, simply stacking up more convolutional layers makes learning more difficult and causes over-fitting or performance saturation [6], [7].

To resolve this and to achieve higher denoising quality, we propose a channel attention-based network for denoising images (CANDI). Our network adopts channel attention, which was first proposed by Hu *et al.* [8] to enhance image classification accuracy. Specifically, Hu *et al.* compute channel-wise weights, or channel attention, from a given feature volume and recalibrate the channels in the feature volume using the weights. In this way, informative features can be emphasized and selectively used for more accurate image classification.

In our work, we adopt channel attention to emphasize informative features that help separate out noise from image contents. More specifically, extracting features from a noisy image using a CNN is analogous to extracting different frequency components of an image using a transform such as wavelet transforms. Among different frequencies, image noise is most distinguishable in high-frequency components, so classical denoising methods focus on such components [9]. In the case of a CNN, different feature channels correspond to different frequency components, so we may selectively use informative channels using channel attention to more effectively identify noise.

Based on this motivation, we first present CANDI and examine different network configurations to find an optimal architecture for CANDI (Sec. III). We then con-

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

duct an analysis of the effect of channel attention on image denoising with respect to different image contents and noise levels (Sec. IV). Through the analysis, we verify that channel attention does select informative feature channels with high-frequency components related to noise. We also show that channel attention has an adaptive nature to image contents and different noise levels. Based on the analysis, we also propose a locally adaptive channel attention-based network for denoising images (LACANDI) (Sec. V). Experimental results on natural images with Gaussian noise show that CANDI outperforms most state-of-the-art methods in both blind and non-blind settings, and LACANDI substantially improves image denoising quality as well as computation time (Sec. VI).

Our main contributions are summarized as follows:

- We propose novel channel attention-based networks for image denoising that outperform state-of-the-art methods in both non-blind and blind settings.
- We provide an analysis of the effect of channel attention on image denoising, which shows that channel attention boosts denoising performance in three aspects. 1) It makes a network to focus on informative channels more closely related to noise. 2) It adapts a network to image contents to faithfully restore a clear image. 3) It also adapts a network to different noise levels for effective blind denoising.
- We present locally adaptive channel attention for modeling locally different nature of natural images.

II. RELATED WORK

A. IMAGE DENOISING

Early approaches usually use explicit modeling of the characteristics of natural images. Filtering based approaches such as Gaussian filtering and bilateral filtering [10] assume that nearby pixels have similar values. Total variation [11], [12] assumes the magnitudes of image gradients follow a Laplacian distribution. Buades *et al.* [13] proposed to exploit the self-similarity property of natural images. This property has been widely applied in many following works such as [4], [5], [7], [14], [15].

To more faithfully capture the characteristics of natural images, learning-based approaches have been actively studied. Elad and Aharon [16] learn over-complete dictionaries for image denoising. Yang *et al.* [17] proposed coupled dictionaries learned from high-and low-resolution image patches for single-image super-resolution. Roth and Black [18] proposed the Fields of Experts framework that learns potential functions of Markov random fields. Zoran and Weiss [19] proposed an image prior based on a Gaussian mixture model of natural image patches. However, their performance is limited as they rely on relatively simple models compared to recent deep learning-based approaches.

For the last few years, deep learning has been actively applied to image denoising. Chen and Pock [2] proposed a non linear diffusion model called TNRD. Mao *et al.* [3] introduced a fully convolutional encoding-decoding framework for image denoising and super-resolution. Zhang *et al.* [1] proposed DnCNN that adopts residual learning [6] and batch normalization [20]. Yang and Sun [4] presented a BM3D-Net, which is inspired by BM3D [14]. Lefkimmiatis [21] developed a non-local operator to exploit self-similarity, and a deep network architecture consisting of several non-local operators. Plötz and Roth [5] introduced a deep learning architecture based on differential K-nearest neighbor selection called a neural nearest neighbors block. Liu *et al.* [7] presented a recurrent network based on non-local recurrent modules to exploit self-similarity. While deep learning-based approaches have shown superior results to classical ones, their performance is limited as they treat useful and less useful features in the same way.

B. CONTENT-ADAPTIVE IMAGE RESTORATION

Our work is also closely related to content-adaptive image restoration techniques for reflecting different characteristics of different images. As different images may have different characteristics, adaptive techniques to image contents have been proposed. Saquib *et al.* [22] proposed to estimate parameters of a prior from a noisy image for image restoration. As a single image may have different characteristics in different local areas, locally adaptive approaches have also been proposed. Bishop *et al.* [23] introduced an image restoration approach that splits an image into regular grid cells and adapts a prior to each cell. Cho *et al.* [24] proposed to estimate parameters of a prior from local image regions. Sun *et al.* [25] showed that locally adapted priors can significantly improve the quality of non-blind deconvolution. On the other hand, recent deep learning-based image denoising approaches treat different channels representing different features in a fixed way regardless of image contents. Thus, they can be considered analogous to using one universal model or prior to all images in classical approaches, and bear the same limitations.

C. CHANNEL ATTENTION

Since Hu *et al.*'s work [8], a few following works that use channel attention have been introduced. Woo *et al.* [26] proposed a convolutional block attention module that combines channel attention and spatial attention for high-level vision tasks. Regarding low-level vision tasks, Zhang *et al.* [27] and Cheng *et al.* [28] recently proposed single image super-resolution approaches that utilize channel attention. However, both of them do not target at image denoising. Furthermore, we provide a careful analysis on the effect of channel attention on image denoising, and present locally adaptive channel attention based on it.

III. CHANNEL ATTENTION-BASED NETWORK FOR DENOISING IMAGES

In this section, we first review the channel attention module [8]. We then introduce the architecture of CANDI and explain its training. Finally, we examine different design options for CANDI. For brevity, we denote convolution, batch

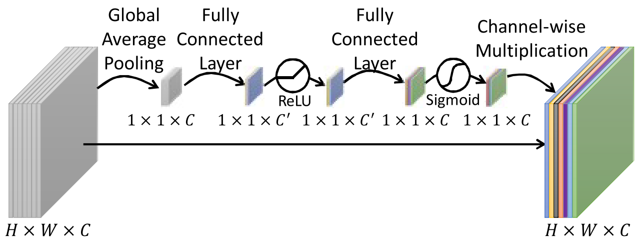


FIGURE 1. **A channel attention module [8].**

normalization [20] and rectified linear unit [29] by Conv, BN and ReLU, respectively in the rest of the paper.

A. CHANNEL ATTENTION MODULE

A channel attention module, or a squeeze-and-excitation block, was recently proposed by Hu *et al.* [8] for enhancing image classification accuracy. Fig. 1 illustrates the architecture of a channel attention module. Given a feature volume, the module computes channel-wise global statistics using global average pooling, and per-channel weights, or channel attention, ranging from 0 to 1 from the statistics. Each per-channel weight is then multiplied to its corresponding channel of the input feature volume to produce a re-scaled feature volume. In this way, informative channels can be emphasized while less useful ones are suppressed.

B. ARCHITECTURE OF CANDI

Fig. 2 shows the architecture of our denoising network, CANDI. CANDI takes a noisy grayscale image as input and predicts its noise map, which can be subtracted from the input image to produce a denoised result. We adopt the residual learning strategy that predicts a residual map, or a noise map in our case, as the strategy has consistently shown to outperform direct estimation of a restored image in recent image restoration methods [1], [7].

As shown in Fig. 2, CANDI consists of a series of residual blocks except for the first and last blocks. The first block extracts shallow features from an input image. It consists a Conv layer with 64 filters of size $3\times3\times1$ followed by a ReLU layer for non-linearity. In the middle, we have 20 residual blocks that perform denoising in the feature space. Each residual block consists of three Conv+BN+ReLU followed by one Conv and one channel attention module. Each block has a skip connection to ease the training of a deep network. Every Conv layer in the residual blocks has 64 filters of size $3 \times 3 \times 64$. We set the number of channels (C' in Fig. 1) in the middle of the channel attention modules as 4 using the reduction rate of 16 suggested by Hu *et al.* [8]. Finally, the last block reconstructs a noise map from features, and has a single Conv layer of size $3 \times 3 \times 1$.

The network architecture of CANDI is mainly inspired by DnCNN, which is a state-of-the-art CNN-based denoising method [1]. Removing the skip connections, Conv layers marked in yellow in Fig. 2, and channel attention modules, CANDI reduces to DnCNN. The effects of the additional components adopted to CANDI will be examined in Sec. III-D.

C. TRAINING

To evaluate the performance of CANDI, we train a few different models including models for known specific noise levels, and a blind model for unknown noise levels. We refer the models trained for known noise levels as CANDI, and the blind model as CANDI-B. In the following, we describe how we train both CANDI and CANDI-B.

1) LOSS FUNCTION

To train CANDI and CANDI-B, we use an L^2 loss function. Specifically, given a training dataset $D = \left\{\ldots, \left(I^{(i)}, J^{(i)}\right), \ldots\right\}$ where $I^{(i)}$ and $J^{(i)}$ are the i-th noisy image, and its corresponding ground truth noise-free image, respectively, we minimize the following loss function:

$$L\left(\Theta; D\right) = \sum_{i} \left\| \left(I^{(i)} - f(I^{(i)}; \Theta) \right) - J^{(i)} \right\|^2 \qquad (1)$$

where Θ is a set of network parameters, and $f(I^{(i)}; \Theta)$ is noise predicted by CANDI with parameters Θ.

2) TRAINING DATA FOR CANDI

We generate training data following [2]. We use 400 images from the training and test sets of BSD500 dataset [30], each of which consists of 200 images. We downsample each image by scaling factors of 0.9, 0.8, and 0.7, and obtain four images including the original one. Each image is then randomly cropped into 180×180. From each cropped image, we extract 40×40-sized patches with stride of 10×10. We augment each patch by random horizontal and vertical flips and random rotation by 90°'s, and obtain two augmented versions. Through this process, we generate 476,800 patches. In our experiments, we consider three noise levels: $\sigma = 15$, 25, and 50. To train CANDI for each noise level, we add Gaussian noise of each noise level to the generated patches.

3) TRAINING DATA FOR CANDI-B

For CANDI-B, we follow the training strategy for the blind version of DnCNN [1]. We use the same 400 images from the BSD500 dataset, and perform the same procedure except for a couple of steps. First, we extract patches of 50×50 instead of 40×40 as done in [1]. Second, for each patch, we randomly sample a noise level σ from a uniform distribution defined on $[0, 55]$, and add Gaussian noise of σ to the patch.

4) TRAINING SETUP

For both CANDI and CANDI-B, we initialize the weights of all Conv layers by random normal initialization with zero-mean and a standard deviation of 0.0005. We use Adam optimizer [31] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We set the learning rate as 0.001 and reduce it by half every 30 epochs. We use a mini-batch size of 64, and train the models for 100 epochs. We used PyTorch [32] to implement and train CANDI and CANDI-B. The training of each model takes four days using an Intel Zeon E5-2620 @ 2.0 GHz and an NVIDIA TITAN RTX (24GB).

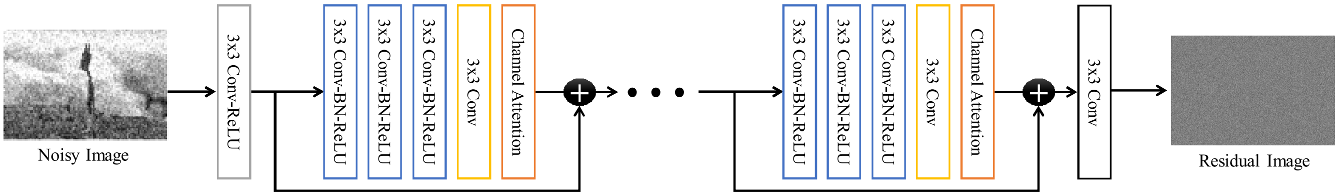


FIGURE 2. The architecture of CANDI.

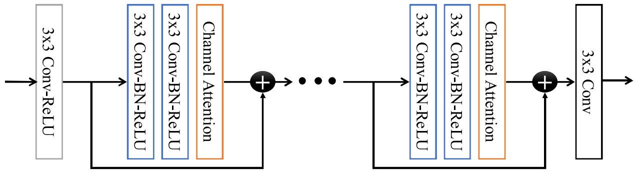


FIGURE 3. The architecture of CANDI$_{2\times8}$, which has two Conv layers in each residual block.

D. HYPERPARAMETERS AND NETWORK DESIGN

The network architecture is one of the most important factors for the performance of a neural network. In this section, we examine several different design options to find the optimal network architecture for CANDI. As our model is based on DnCNN [1], we begin with the architecture of DnCNN and examine different options one by one. In all the experiments, the performance of different models are evaluated using the Set12 dataset [1] for a noise level $\sigma = 50$. All models including DnCNN in this section were trained for 50 epochs using the setting described in Sec. III-C.

1) NUMBERS OF CONV LAYERS IN THE RESIDUAL BLOCK

We first conducted an experiment to find an optimal number of Conv layers in each residual block. Specifically, we prepared variants of CANDI with different numbers of Conv layers ranging from 1 to 5. We set the total number of layers of each variant as either 17 or 18 to compare them with DnCNN with 17 layers. We denote a model with y residual blocks with x Conv layers by CANDI$_{x\times y}$. CANDI$_{x\times y}$ has $(x\times y+2)$ Conv layers in total including the ones in the first and last blocks. The number of channel attention modules also varies across different models. For example, CANDI$_{1\times15}$ has 15 modules while CANDI$_{5\times3}$ has only three. Fig. 3 depicts a variant of CANDI (CANDI$_{2\times8}$) tested in this experiment.

Table 1 shows the experimental result. Among different versions of CANDI, CANDI$_{4\times4}$ achieves the highest PSNR. This suggests that simply using more channel attention modules does not improve denoising quality, but the number of modules should be carefully balanced.

2) NETWORK DEPTH

In the next experiment, we fix the number of Conv layers in each residual block as four, and gradually increase the number of residual blocks y to find out an optimal depth. Table 2 reports the result. It shows that the performance gradually increases until y reaches at 20, and it drops when y is 30. A possible reason of the performance drop for $y = 30$ is overfitting, which may be solved using a larger amount of training data. Based on this result, we fix the number of blocks as 20 in our final model.

TABLE 1. A comparison of different architectures. The best performance is in bold.

Method	PSNR (dB)
DnCNN	27.10
CANDI$_{1\times15}$	27.07
CANDI$_{2\times8}$	27.10
CANDI$_{3\times5}$	27.11
CANDI$_{4\times4}$	**27.15**
CANDI$_{5\times3}$	27.12

TABLE 2. A comparison of CANDI$_{4\times y}$ with different numbers of residual blocks. The best performance is in bold.

# residual blocks (y)	4	10	20	30
PSNR (dB)	27.15	27.18	**27.36**	27.32

3) THE STRUCTURE OF THE RESIDUAL BLOCK

Finally, we test a few different designs for the residual blocks. We compare six different designs shown in Fig. 4. Fig. 4(a) corresponds to a simple extension of DnCNN, which has 82 convolution layers. Fig. 4(b) corresponds to an extension of DnCNN with skip connections, each residual block of which consists of four Conv+BN+ReLU and a skip connection, but has no channel attention modules. Fig. 4(c) has a channel attention module, but no BNs. Fig. 4(d) has BNs as well as a channel attention module. Fig. 4(e) and (f) are obtained by removing ReLU and BN one by one from the last block before the channel attention module of Fig. 4(d). Fig. 4(f) corresponds to our final model presented in the main paper. Using these blocks, we prepared five variants of CANDI, each of which has 20 residual blocks.

Table 3 reports the performance of the residual blocks. The simple extension of DnCNN without skip connection (Table 3(a)) did not converge during training possibly due to the increased difficulty of training as the depth of the network is much larger than the original DnCNN. Table 3(c) and (d) show that adding BNs significantly boosts denoising performance. Table 3(b) and (d) show that a channel attention module also considerably boosts the performance, proving its effectiveness on image denoising. Another interesting finding is that, as shown by (d), (e), and (f), removing ReLU and BN one by one from the block before the channel attention module gradually increases the

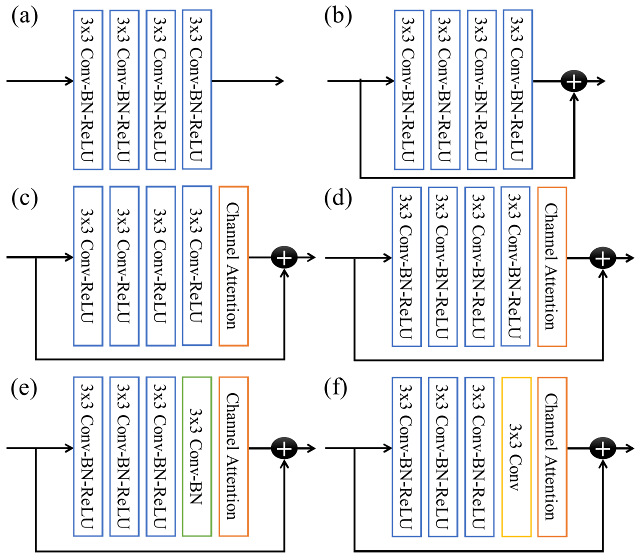


FIGURE 4. Different architectures to verify the effect of each component. (a) corresponds to a simple extension of DnCNN with 82 Conv layers. (b) is an extension of DnCNN with skip connections. (c) has a channel attention module but no BNs. (d) has BNs as well as a channel attention module. (e) and (f) are obtained by removing ReLU and BN one by one from (d).

TABLE 3. A comparison of the residual blocks in Fig. 4.

Model	(a)	(b)	(c)	(d)	(e)	(f)
PSNR (dB)	NA	27.30	26.70	27.36	27.38	**27.42**

performance. Although the reason is unclear, we conjecture that this is because features for image denoising are closely related to intensity values, and non-linear functions such as BN and ReLU can break the relationship between them.

IV. ANALYSIS ON CHANNEL ATTENTION

A. CHANNEL SELECTION

To investigate how channel attention helps image denoising, we first examine what channels are selected by the channel attention modules. To this end, we visualize channel attention weights computed from a noisy image and their corresponding feature channels at different depths (Fig. 5). The input image has Gaussian noise of a noise level $\sigma = 25$. We sampled channel attention values from the first and 11th residual blocks.

At the first residual block, features corresponding to the largest channel attention weights are less correlated with the structural content of the image, and show more random and high-frequency patterns. On the other hand, features corresponding to the smallest channel attention weights are more correlated with the structural content of the image. The features at the 11th residual block show a similar tendency too, while the tendency becomes less obvious. This verifies that channel attention emphasizes channels corresponding to high-frequency components closely related to noise.

B. CONTENT-ADAPTIVITY

A channel attention module aggregates information from different spatial locations, which may possibly encode the global context of an input image. Thus, in the next experiment, we investigate whether the channel attention modules reflect the content of an input image to better restore a clean image as done in content-adaptive image restoration techniques [22], [24], [25].

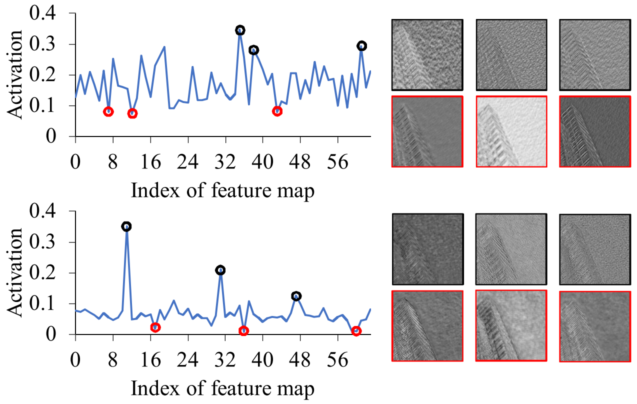


FIGURE 5. On the left, channel attention weights from the first and the 11th residual blocks are shown on the top and bottom rows, respectively. The largest and smallest channel weights are marked in black and red, respectively, and their corresponding feature channels are shown on the right.

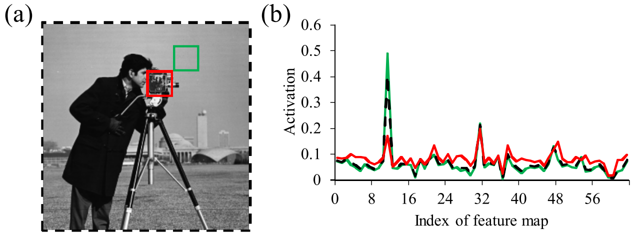


FIGURE 6. (a) An example image with other contents. (b) The channel attention weights at the 12th residual block.

To verify this, we first check whether channel attention modules produce different weights with respect to different image contents. Specifically, we added Gaussian noise of a noise level $\sigma = 25$ to an image (Fig. 6(a)). Then, we cropped two sub-images (marked in red and green in Fig. 6(a)) and fed them as well as the original noisy image to CANDI to capture their channel attention weights. We found that most channel attention modules produce almost identical weights regardless of image contents except for one or two modules. Fig. 6(b) visualizes the channel attention weights at the 12th residual block that produces different weights with respect to image contents. The channel attention weights of the three images are similar but clearly different to the weights of the other images, showing that they are adaptively determined to image contents.

To more clearly verify the content-adaptivity of channel attention, we examine the effect of adaptively computed channel attention weights. To this end, we applied CANDI to the sub-image marked in red in Fig. 6(a) using three different channel attention weights computed from the red sub-image, from the green sub-image, and from the entire image. Then, we measured their PSNR values. The PSNR values of the original noisy image and its denoising result with different

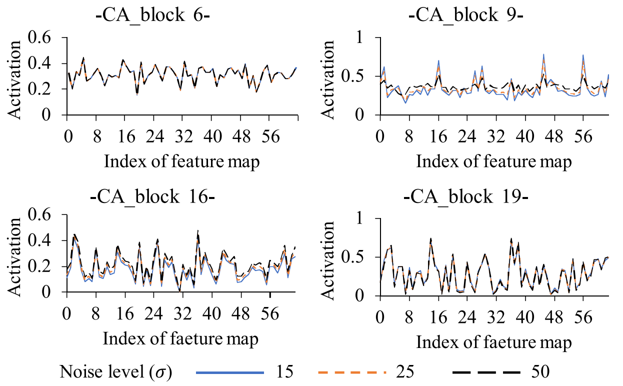


FIGURE 7. Channel attention weights of different noise levels (σ = 15, 25 and 50) at the 6th, 9th, 16th, and 19th residual blocks.

channel attention weights are 20.64, 25.27, 24.81, and 25.04 dB, respectively. The result shows that non-adaptive weights can still remove noise. Among the denoised images, the one obtained using channel attention weights from totally different contents has the lowest PSNR, while the one using channel attention weights from its own content has the highest value. This shows that the content-adaptivity of channel attention can improve denoising quality, analogously to previous content-aware priors [24], [25].

C. NOISE-ADAPTIVITY

We also investigate whether channel attention is adaptive to noise, and whether its adaptivity helps handle different levels of noise. To this end, we prepared three images of the same scene with different noise levels ($\sigma = 15, 25,$ and 50). Then, we fed them into CANDI-B and captured their channel attention weights. Fig. 7 visualizes the weights of the different noise levels at different depths. As shown in the figure, different noise levels produce different channel attention weights, indicating that the channel attention modules are adaptive to noise.

To verify whether the adaptivity of channel attention helps handle different levels of noise, we conducted another experiment using the same three images used earlier. Specifically, in this experiment, we remove noise from the image with $\sigma = 25$ using three different channel attention weights computed from the images with noise levels 15, 25, and 50. Fig. 8 shows the results. The PSNRs of the original noisy image with $\sigma = 25$, and its denoising results are 20.47, 24.66, 30.22, and 26.67 dB, respectively. Qualitatively speaking, the result using the attention weights of $\sigma = 15$ (Fig. 8(b)) has remaining noise, while the result using the attention weights of $\sigma = 50$ (Fig. 8(d)) has blurry details. On the other hand, the result using the attention weights of $\sigma = 25$ (Fig. 8(c)) has clearly restored details and no remaining noise. This behavior of channel attention is analogous to denoising strength parameters of traditional denoising algorithms such as the range sigma of the bilateral filter [10], and also shows that channel attention adapts the network to more effectively remove noise with respect to different noise levels.

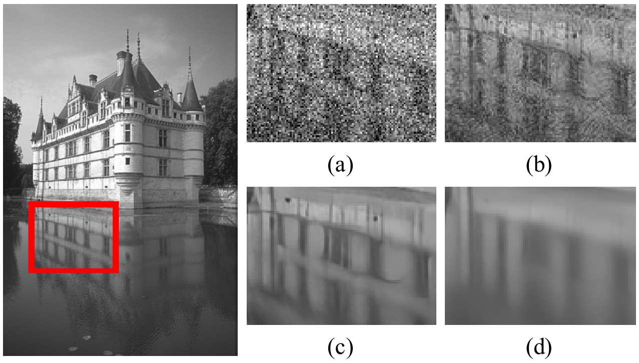


FIGURE 8. Left: A clean image, Right: (a) a noisy image with a noise level σ = 25, (b) a denoising result with the channel attention weights calculated from the image with a noise level σ = 15, (c) A denoising result with the channel attention weights calculated from the image with a noise level σ = 25 (d) a denoising result with the channel attention weights calculated from the image with a noise level σ = 50.

V. LOCALLY ADAPTIVE CHANNEL ATTENTION-BASED NETWORK FOR DENOISING IMAGES

The analysis in Sec. IV shows that channel attention has a content-adaptive property. While different images have different types of contents, even a single image may also have different types of contents on different local areas. However, the channel attention module cannot model such locally different nature of natural images due to the global average pooling operation. Inspired by this observation, in this section, we develop a locally adaptive channel attention module that allows us to compute spatially different channel attention. A locally adaptive CANDI (LACANDI) is then obtained by simply replacing all channel attention modules in CANDI by locally adaptive channel attention modules.

To compute locally adaptive channel attention, we split an input feature volume into a regular grid, and compute channel attention for each grid cell. To this end, we modify the channel attention module. We first replace the global average pooling by local average pooling that computes the average value for each grid cell. Specifically, for an input feature volume of size $k_wW' \times k_hH' \times C$ where k_w and k_h are the numbers of cells along the horizontal and vertical axis, respectively, and $W' \times H'$ is the size of each grid cell, we define a local average pooling operator as a combination of mean filtering and subsampling. Mathematically, the local average pooling operator LAP is defined as:

$$z_c = LAP(x_c) = D(f * x_c) \qquad (2)$$

where x_c is the c-th channel of an input feature volume x, f is a mean filter, $*$ is a convolution operator, and D is a decimation operator that subsamples the input feature value at the center of each grid cell. z is the output feature volume of the local average pooling operator, which has the size of $k_w \times k_h \times C$. In our experiments, we use a mean filter of $W' \times H'$, whose elements are $1/(W'H')$, for f. We also replace fully connected layers in the channel attention module by 1×1 Conv layers.

Fig. 9 illustrates the network structure of a locally adaptive channel attention module. Using a locally adaptive channel attention module, we obtain a channel attention map

TABLE 4. A quantitative comparison of different methods. The first and the second best performance are in bold and underlined, respectively.

Dataset	Noise	BM3D	TNRD	DnCNN	N^{3}Net	LACANDI-B	CANDI	LACANDI
Set12	15	32.37/0.8952	32.50/0.8958	32.86/0.9031	-/-	32.86/0.9028	32.95/0.9047	**32.98/0.9052**
	25	29.97/0.8504	30.06/0.8512	30.44/0.8622	30.55/0.8640	30.56/0.8649	30.63/0.8657	**30.65/0.8669**
	50	26.72/0.7676	26.81/0.7680	27.18/0.7829	27.43/0.7935	**27.45**/0.7945	27.42/0.7936	27.45/**0.7947**
BSD68	15	31.07/0.8717	31.42/0.8769	31.73/0.8907	-/-	31.73/0.8908	31.79/0.8928	**31.82/0.8934**
	25	28.57/0.8013	28.92/0.8093	29.23/0.8278	29.30/0.8320	29.30/0.8304	29.32/0.8330	**29.34/0.8338**
	50	25.62/0.6864	25.97/0.6994	26.23/0.7189	26.39/0.7296	26.39/0.7270	26.38/0.7310	**26.40/0.7319**
Urban100	15	32.35/0.9220	31.86/0.9031	32.68/0.9255	-/-	32.52/0.9215	32.74/0.9293	**32.76/0.9298**
	25	29.70/0.8777	29.25/0.8473	29.97/0.8797	30.19/**0.8913**	30.13/0.8841	30.26/0.8899	**30.27**/0.8905
	50	25.95/0.7791	25.88/0.7563	26.28/0.7874	**26.82/0.8144**	26.80/0.8075	26.71/0.8060	26.74/0.8078

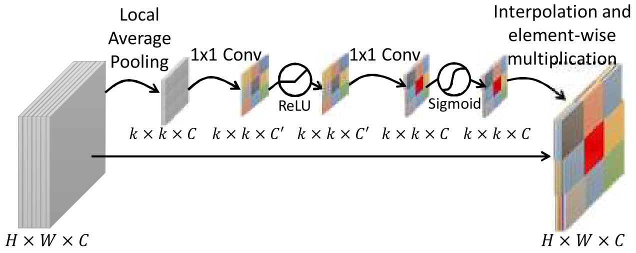


FIGURE 9. A locally adaptive channel attention module.

of size $k \times k \times C$. To re-scale the input feature volume, we upsample the map to the size of the input feature volume. For upsampling, we use bilinear interpolation to introduce smooth transition between grid cells and to avoid tiling artifacts. The upsampled channel attention map is then multiplied to the input feature volume element-wise to obtain a re-scaled feature volume.

A. TRAINING

In our experiments, we did not train LACANDI models separately, but simply reused the weights of CANDI models. This is possible because a fully connected layer in a channel attention module is equivalent to a 1×1 Conv layer in a locally adaptive channel attention module. Also, as we use small training images of 40×40, we can safely assume that the weights of CANDI models are already trained to be adaptive to small local areas. We also introduce a blind version of LACANDI, denoted by LACANDI-B. For LACANDI-B, we reused the weights of CANDI-B.

B. GRID SIZE

If we set $k_w = W$ and $k_h = H$ for an input image of size $W \times H$ and use a mean filter of size $W' \times H'$, we can compute locally adaptive channel attention weights for all pixels. Nonetheless, we empirically found that $k_w = k_h = 10$ works well in most cases. We compared these two options using the BSD68 dataset for a noise level $\sigma = 25$. Both LACANDI models with ($k_w = 10, k_h = 10$) and ($k_w = W$, $k_h = H$) achieve 29.34 dB, but their computation times are 0.09 seconds and 0.74 seconds, respectively, as the model with ($k_w = W, k_h = H$) needs a much larger amount of computation. In the rest of this paper, we use $k_w = k_h = 10$ for both LACANDI and LACANDI-B.

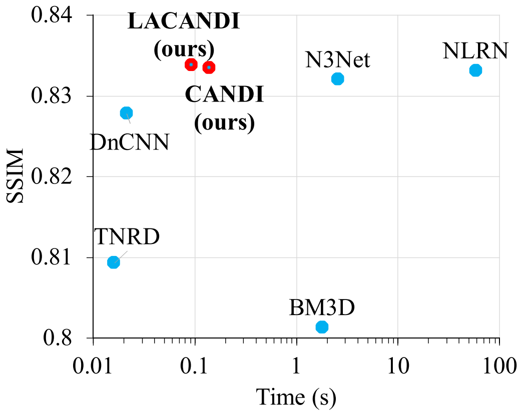


FIGURE 10. The average computation times and denoising qualities of different methods.

VI. EXPERIMENTS

In this section, we evaluate the performance of our final models: CANDI, LACANDI, CANDI-B, and LACANDI-B. For evaluation, we use three widely used benchmark datasets: BSD68 [33], Set12 [1], and Urban100 [34]. We compare our models with several state-of-the-art denoising methods: BM3D [14], TNRD [2], DnCNN [1], N^{3}Net [5], NLRN [7], DnCNN-B [1], and GCBD [35]. Except for BM3D, all the methods are learning-based. BM3D, N^{3}Net and NLRN exploit non-local self-similarity to effectively handle repeated structures. DnCNN-B is a blind version of DnCNN that shares the same architecture. GCBD is also a blind denoising method, which is based on generative adversarial networks. We refer the readers to our supplementary material for more details on the evaluation in this section and the analysis in Sec. IV.

A. QUANTITATIVE COMPARISON

We first quantitatively compare our models with state-of-the-art non-blind methods in terms of PSNR and structural similarity index (SSIM) [36], which is another widely used measure for image quality assessment. A higher SSIM value means that an image is more similar to the ground truth one. The PSNR and SSIM values of all the other methods are from their papers except for the SSIM values of N^{3}Net, which are not reported in [5]. For the SSIM values of N^{3}Net, we measured them using the trained models provided by the authors.

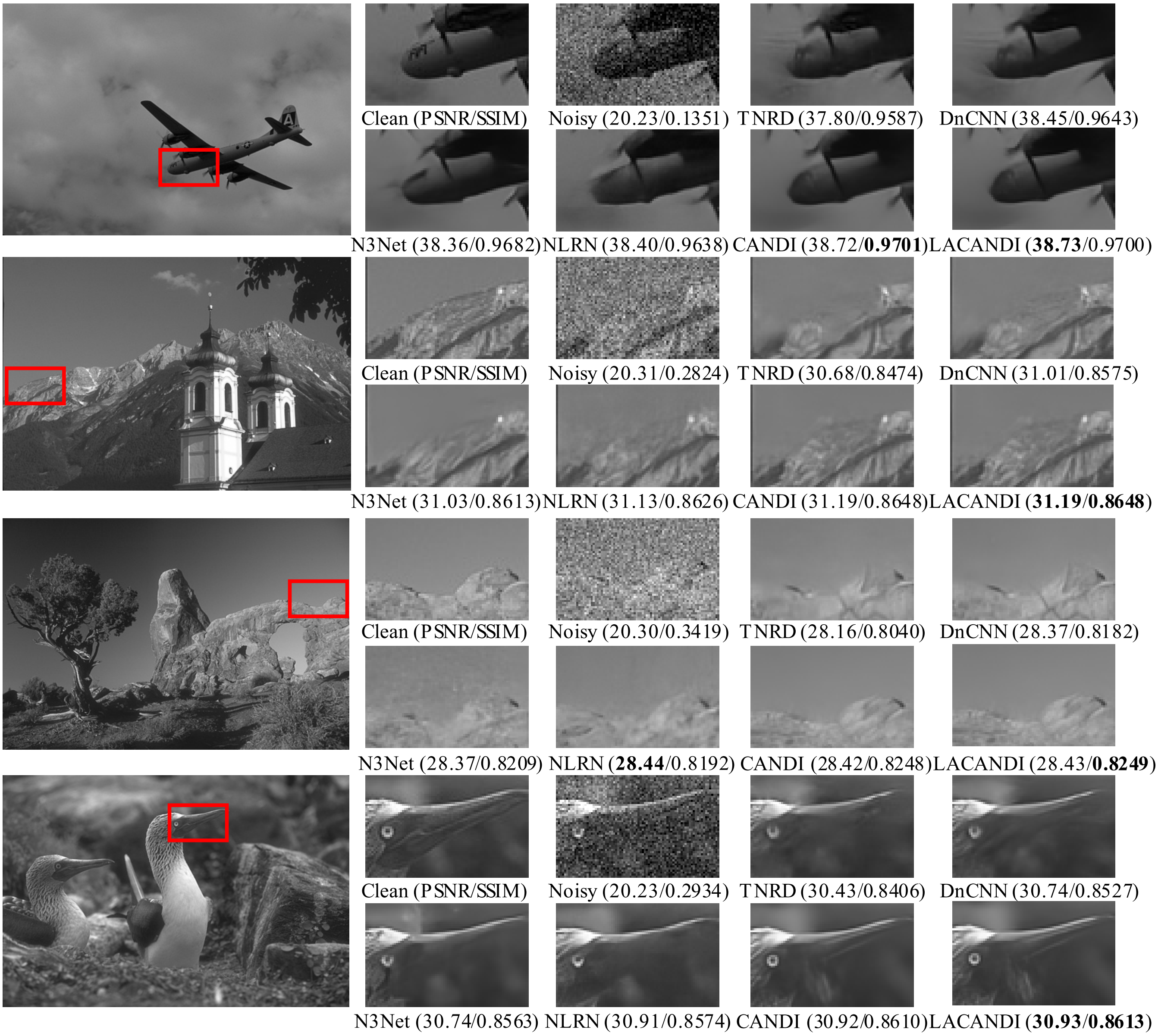


FIGURE 11. A qualitative comparison for a noise level $\sigma = 25$. Left: ground-truth images. Right: magnified views of different image denoising results.

Table 4 shows a quantitative comparison. It is shown that LACANDI outperforms CANDI in all cases in terms of both PSNR and SSIM, which validates the effectiveness of the locally adaptive channel attention modules, and their content-adaptivity. It is also shown that CANDI and LACANDI outperform the other methods in most cases. Even LACANDI-B, a blind version of LACANDI, shows similar performance to N^{3}Net, which is non-blind, on the Set12 and BSD68 datasets thanks to its noise-adaptivity. NLRN, which is another state-of-the-art method, is not included in this comparison because it is orders-of-magnitude slower than our models as will be discussed later. Table 5 shows a quantitative comparison against state-of-the-art blind denoising methods. It shows that LACANDI-B outperforms all the other methods by a large margin.

TABLE 5. A quantitative comparison of blind denoising methods on the BSD68 dataset. The best performance is in bold.

Noise	DnCNN-B	GCBD	CANDI-B	LACANDI-B
15	31.61	31.59	31.73	**31.73**
25	29.16	29.15	29.30	**29.30**
50	26.23	-	26.38	**26.39**

B. QUALITATIVE COMPARISON

Fig. 11 shows a qualitative comparison against state-of-the-art methods. The figure shows that our models produce less artifacts than the other ones especially on the first, third and last rows. The second and third rows show that our models preserve more details than the others. More examples can be found in the supplementary material.

TABLE 6. The average computation times of different denoising methods on the BSD68 dataset.

Methods	TNRD	DnCNN	LACANDI	CANDI	BM3D	N^3Net	NLRN
Time (s)	0.016	0.02	0.09	0.13	1.8	2.56	67.95

C. COMPUTATION TIME

Finally, we compare the computation times of state-of-the-art methods and ours. The computation times were measured using the authors' code in the same environment as the training environment (an Intel Zeon E5-2620 @ 2.0 GHz and an NVIDIA TITAN RTX). Table 6 reports the average computation times on the BSD68 dataset that has images of either 321 × 481 or 481 × 321. Both our models take about 0.1 seconds to denoise a single image, which shows that our models can be used in practical applications. Compared to N^3Net and BM3D, our models are an order-of-magnitude faster. Compared to NLRN, LACANDI is more than 700 times faster. N^3Net, BM3D, and NLRN perform feature matching to exploit non-local self-similarity, so they require relatively large computation times. Especially, NLRN repeatedly performs feature matching, which causes a significant amount of computation. Interestingly, LACANDI is faster than CANDI even though LACANDI computes a larger number of channel attention weights. This is because local average pooling is more GPU-friendly than global average pooling.

Fig. 10 visually compares the computation times and denoising qualities of different methods measured on the BSD68 dataset for a noise level $\sigma = 25$. The figure shows that our methods outperform all the other methods except for NLRN in terms of both PSNR and SSIM, although they require relatively small amounts of computation. In terms of PSNR, our methods are worse than NLRN. Specifically, the PSNR and SSIM of NLRL are 29.41 dB and 0.8331, respectively, while those of LACANDI are 29.34 dB and 0.8338. However, ours are orders-of-magnitude faster.

VII. CONCLUSION

In this paper, we proposed CANDI, a novel channel attention-based network for denoising images. Then, we analyzed the effect of channel attention on image denoising, and showed that channel attention has an adaptive nature to image contents and noise. Based on this, we proposed a locally adaptive channel attention module and an image denoising network, LACANDI, based on it. Experimental results showed that both CANDI and LACANDI and their blind versions outperform state-of-the-art methods.

We believe that locally adaptive channel attention can also benefit other problems such as super-resolution, deblurring, and high-level vision problems such as segmentation. The performance of image denoising may depend on the complexity of image contents, and an analysis on this may help design a more effective network structure for denoising. LACANDI splits an image into a uniform grid, which may hinder fully exploiting locally different characteristics of natural images. To resolve this, we may adopt semantic segmentation. Exploring such possibilities would be an interesting future direction.

REFERENCES

[1] 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.

[2] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[3] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[4] D. Yang and J. Sun, "BM3D-Net: A convolutional neural network for transform-domain collaborative filtering," *IEEE Signal Process. Lett.*, vol. 25, no. 1, pp. 55–59, Jan. 2018.

[5] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1095–1106.

[6] 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.

[7] D. Liu, B. Wen, Y. Fan, C. Change Loy, and T. S. Huang, "Non-local recurrent network for image restoration," 2018, *arXiv:1806.02919*. [Online]. Available: http://arxiv.org/abs/1806.02919

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[9] R. C. González and R. E. Woods, *Digital Image Processing*, 3rd ed. London, U.K.: Pearson, 2008. [Online]. Available: http://www.worldcat.org/oclc/241057034

[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Nov. 2002, pp. 839–846.

[11] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.

[12] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–489, Jan. 2005.

[13] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2005, pp. 60–65.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[15] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[17] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[18] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2005, pp. 860–867.

[19] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[21] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3587–3596.

[22] S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields with applications to Bayesian tomography," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1029–1044, Jul. 1998.

[23] T. E. Bishop, R. Molina, and J. R. Hopgood, "Nonstationary blind image restoration using variational methods," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2007, pp. I-125–I-128.

[24] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman, "A content-aware image prior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 169–176.

[25] L. Sun, S. Cho, J. Wang, and J. Hays, "Good image priors for non-blind deconvolution," in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 231–246.

[26] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[27] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 294–310.

[28] X. Cheng, X. Li, J. Yang, and Y. Tai, "SESR: Single image super resolution with recursive squeeze and excitation networks," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 147–152.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[30] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS) Autodiff Workshop*, Dec. 2017.

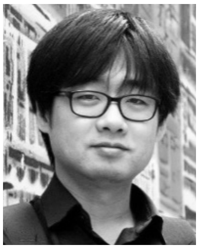
[33] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

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

[35] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

HAEYUN LEE (Student Member, IEEE) was born in Iksan, South Korea. He received the B.S. degree in mathematics from Chonbuk National University, Jeonju, South Korea, in 2016, and the M.S. degree in information and communication engineering from the Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea, in 2018, where he is currently pursuing the Ph.D. degree in information and communication engineering. His current research interest includes the image restoration and medical image analysis based on deep learning.



SUNGHYUN CHO (Member, IEEE) received the B.S. degrees in computer science and in mathematics from Pohang University of Science and Technology (POSTECH), in 2005, and the Ph.D. degree in computer science from POSTECH, in February 2012. He spent six months as an Intern at Microsoft Research Asia, Beijing, in 2006, and four months as an Intern at Adobe Research, Seattle, in 2010. From March 2012 to March 2014, he was a Research Scientist with Adobe Research, Seattle. From April 2014 to April 2017, he was a Senior Engineer with Samsung Electronics. From April 2017 to August 2019, he was an Assistant Professor with the Daegu Gyeongbuk Institute of Science and Technology (DGIST). He is currently an Assistant Professor with POSTECH. His research interests include computational photography, image/video processing, computer vision, and computer graphics. In 2008, he was a recipient of the Microsoft Research Asia 2008/09 Graduate Research Fellowship Award.