Knowledge-Based Systems 231 (2021) 107384

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Learning Deformable and Attentive Network for image restoration ☆

Yuan Huang [a,1], Xingsong Hou [a,*], Yujie Dun [a,*], Jie Qin [b,*], Li Liu [c], Xueming Qian [a], Ling Shao [c]

[a] School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[b] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[c] Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Article history:
Received 1 March 2021
Received in revised form 30 July 2021
Accepted 10 August 2021
Available online 19 August 2021

Keywords:
Image restoration
Convolution neural network
Deformable convolution
Attention mechanism
Knowledge distillation
Image denoising
JPEG artifacts removal
Real-world super resolution

ABSTRACT

Image restoration (IR) aims to recover image quality from various degradations. Existing convolutional neural networks (CNN) based IR methods try to improve performance by enlarging the model receptive field with the sacrifice of fine spatial details and extra artifacts. This paper proposes a Deformable and Attentive Network (DANet) to address these problems. In DANet, we propose two novel blocks: Attentive DEformable-convolution Block (ADEB) and Attentive Recurrent Offset Block (AROB). In ADEB, deformable convolution is collaborated with various attention modules to generate more adaptive receptive fields. AROB transfers more attentive texture information among different scales during the encoding/decoding process for ADEB. To further refine DANet, we propose a knowledge distillation scheme to train a light-weighted DANet (DANet-S) with limited performance loss. Extensive experiments on several image benchmark datasets demonstrate that our method achieves state-of-the-art (SOTA) results for various IR tasks, including image denoising, JPEG artifacts removal, and real-world super resolution.

1. Introduction

During image acquisition, many factors can degrade the image quality, such as the limitation of image capturing devices, the environment noises, and the artifacts caused by image compression–decompression methods. Therefore, to restore the high-quality images from the low-quality observations, image restoration (IR) has been widely used in many applications, such as surveillance [1], medical imaging [2] and remote sensing [3]. Approaches for IR can be categorized into traditional model-based methods [4,5] and convolutional neural networks (CNN) based methods [6,7]. Traditional model-based approaches usually use manually designed prior and models. Recently, many CNN-based approaches have achieved remarkable performance in different IR tasks [8], such as image denoising, JPEG artifacts removal and single real image super resolution [9,10].

Most CNN-based IR methods focus on exploring deeper models or multi-scale inputs to enlarge the receptive field and learn enriched features. SRCNN [11] first introduced CNN for IR tasks. Based on SRCNN, VDSR [7] proposed a structure that cascades small filters several times to achieve good performance, and they also found that increasing the depth of the model can bring significant improvements.

While the depth of the models increasing, the gradients will gradually vanish during training. Researchers have developed various methods to tackle this problem. DRCN [12] proposed a deep model with recursive-supervision and gradient clipping method, which combines the predictions from each recursive layer to enhance the final results. DnCNN [6] introduced the residual connection to ease the feature flow in building deeper IR models. RCAN [13] introduced a residual structure combined with the channel-wise attention module to obtain attentive features in the deep model. To fully use the features of the deep model, RDN [14] employed dense connection, local feature fusing, and local residual learning to generate enriched features from different scales.

Employing deep models in IR also brings the over smoothing problem to restored images. Some work employed multi-scale (*e.g.*encoder–decoder structure) inputs to enlarge the receptive field instead of simply increasing the depth in IR models. In REDNet [19], an encoder–decoder structure using convolution and deconvolution layers was proposed with symmetric skip connections for faster training. MWCNN [20] employed wavelet transform instead of upsampling downsampling layers and multi-dilated convolution in an encoder–decoder model to enlarge the

☆ This work was supported in part by the National Key R&D Program of China under Grant 2017YFF0107700, NSFC, China under Grant 61872286, and the Key R&D Program of Shaanxi Province of China under Grant 2020ZDLGY04-05 and S2021-YF-YBSF-0094.

* Corresponding authors.
E-mail addresses: houxs@mail.xjtu.edu.cn (X. Hou), dunyj@mail.xjtu.edu.cn (Y. Dun), qinjiebuaa@gmail.com (J. Qin).

[1] This work was partially done when Yuan Huang was an Intern at IIAI.

https://doi.org/10.1016/j.knosys.2021.107384
0950-7051/

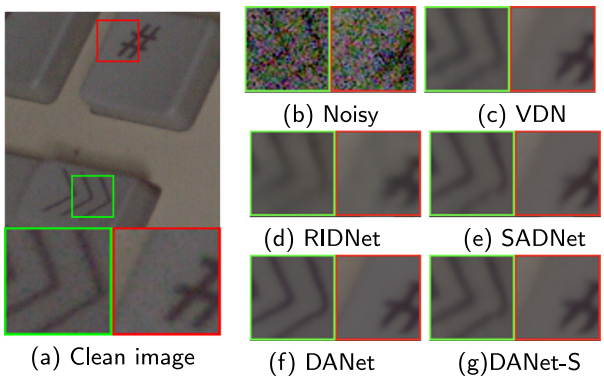


Fig. 1. Real image denoising results of VDN [15] (c), RIDnet [16] (d), SADNet [17] (e) and the propose DANet (f), DANet-S (g) for the image "*11_11.png* " on SIDD [18] dataset.

receptive field. Another way of employing wavelet decomposition in IR is the divide-and-conquer framework [21], which first decomposes images into different subspaces based on visual importance and exploits their prior differences using multiple models to preserve more contextual details. SADNet [17] employed deformable convolution in the decoder to acquire adaptive receptive field achieves better performance.

Though significant improvements have been made in the above methods, several issues still exist in developing a larger receptive field and learning enriched features in IR models. First of all, as the IR models become deeper, like in VDSR [7], DRCN [12], DnCNN [6], Memnet [22] and RCAN [13], the restored images tend to be over smoothing and lack texture details [16,23]. Secondly, although employment of multi-scale inputs and dilated convolution can enlarge the receptive fields, like in RDN [14], MWCNN [20], it also leads to spatially inaccuracy and checkerboard patterns in the restored images [17]. Furthermore, preserving more texture detail information and spatial accuracy at the same time remains to be a problem.

In this paper, to address the above issues, we aim to equip the encoder–decoder models with the capability of preserving spatial details and avoiding additional artifacts. To achieve this, we enhance a U-Net architecture to learn adaptive and attentive receptive fields that can preserve more local details. Specifically, motivated by the recent success of deformable convolution (DeConv) [24] and attention mechanism [25–27], we combine the DeConv with various attention units and propose Attentive DEformable-convolution Block (ADEB). In order to generate more attentive and adaptive receptive fields, we add attention modules in cooperating with deformable convolution. The attention modules can help to provide focus features, and generate more precise sampling locations according to the image content. Since the adaptively captured spatial details will still gradually vanish during downsampling when the network becomes deeper, we propose Attentive Recurrent Offset Block (AROB) as an attentive feature transfer module built upon ADEB. AROB is employed to propagate and maintain multi-scale features for ADEB throughout the network. More specifically, AROB learns from multi-high-resolution features and transfers them to further strengthen those features obtained from low-resolution ones. In this way, high-resolution details can still be preserved during reconstruction.

Moreover, we replace the upsampling and downsampling layers between two consecutive ADEB and AROB blocks by wavelet transform (decomposition/reconstruction), which captures both frequency and location information [20] to prevent the loss of detailed textures during encoding/decoding processes. The above-proposed modules constitute our final network, called the Deformable and Attentive Network (DANet). To further refine the DANet, we propose a knowledge distillation scheme to train a lightweight DANet Slim (DANet-S), which aims to preserve the performance of DANet with fewer parameters and smaller model size.

In another existing method that employs deformable convolution for IR, SADNet [17], the original DeConv was employed in the decoder of a U-net model. Unlike SADNet, we build a novel block, ADEB, based on the DeConv. We embed various attention modules in the block to propagate more attentive multi-scale features for DeConv to learn adaptive receptive fields. In SADNet, a context block is proposed to extract different receptive field features and preserve texture details. In DANet, a recurrent connection among all the AROBs is used to preserve more spatial details from multi-scale features. The attention modules and the recurrent connection in the DANet help to preserve more texture details in the restored images. Fig. 1 shows a visual comparison between SADNet and DANet for real image denoising on the SIDD dataset [18]. We can observe that DANet preserves more precise edges and more contextual details without introducing artifacts than the SADNet and other SOTA methods on real image denoising.

In summary, our main contributions include:

- We propose an encoder–decoder-based IR model named DANet. In DANet, we build two novel blocks (*i.e.*, ADEB and AROB) upon DeConv to learn contextual features with more adaptive receptive fields and preserve fine spatial details in the restored images.
- We propose a knowledge distillation scheme for DANet. With the proposed scheme, we can train a light-weighted version of DANet, DANet-S, which still preserves the performance of DANet.
- We apply DANet and DANet-S for synthetic and realistic noise removal, JPEG artifacts removal, and real image super resolution tasks, which achieve SOTA performance.

The rest of the paper is organized as follows. In Section 2, we introduce the related works. In Section 3, we present our proposed ADEB, AROB, DANet, and DANet-S for image restoration. Extensive experiments are conducted in Section 4 to evaluate the effectiveness of DANet and DANet-S on synthetic and realistic noise removal, JPEG artifacts removal, and real image super resolution tasks. Then ablation studies are presented in Section 5. The conclusion is given in Section 6.

2. Related work

In this section, we briefly describe the typical methods for several IR tasks, including synthetic and real image denoising, JPEG artifacts removal, and image super resolution. We mainly focus on several representative CNN-based approaches since they have achieved a significant improvement comparing to traditional methods.

2.1. Image denoising

In synthetic image denoising, Additive white Gaussian noise (AWGN) is widely used to create synthetic noisy images to evaluate the denoising method. One of the typical architecture of CNN models is based on the encoder–decoder structure (*e.g.*MWCNN [20], SADNet [17], DIDN [28]). Specifically, the high-resolution input or feature maps are downsampled to low-resolution ones of different scales, then upsampled to the original resolution. This procedure ensures that contextual information of different scales can be learned through encoding/decoding, which is more efficient than learning through single resolution models. However, there still exists a major issue in these deep models, *i.e.*, the loss

of important spatial details (*e.g.*, edges and textures) during the encode/decode reconstruction processes. To avoid this problem, another line of works (*e.g.*DnCNN [6], FFDNet [29], RNAN [30], RIDNet [16]) attempt to capture such details by directly operating on high-resolution images without any down-sampling steps. These models, which process only on high-resolution features, tend to have limited receptive fields and lose texture details in the restored results.

Unlike the synthetic image denoising, the realistic noisy images are captured with various camera types under different ISO levels and shutter speeds. Many approaches were recently developed, such as VDN [15], CBDNet [31], which used a two-step model to estimate and remove the denoises. RIDNet [16] and SADNet [17] proposed novel CNN structures to improve the denoising performance using attention mechanism and deformable convolution, respectively. Here we propose DANet to tackle the above problems, as shown in Sections 3 and 4.

2.2. JPEG artifacts removal

Compression artifacts are caused by the image compression method during compression and reconstruction. In CNN-based methods, Artifacts Reduction Convolutional Neural Network (AR-CNN) [32] first applied deep learning for JPEG artifacts removal. To explore deeper CNN models, DnCNN [6] employed the residual connection to the CNN model. Memnet [22] utilized the non-local information in deeper models and proposed an approach with a deep, persistent memory network. RCAN [30] introduced the dense connections and attention mechanism to the models. In the deep convolutional sparse coding (DCSC) [33] network, they sparse coded the feature maps with a CNN-based model instead of the raw images. Furthermore, they also designed multi-scale feature maps to broaden the receptive field of the model. To balance the trade-off between the receptive field size and the model's coefficient scale, MWCNN [20] applied the wavelet decomposition and reconstruction in the encoder–decoder-based model, which is used as a substitute for the downsampling and upsampling modules to improve the performance. We build a novel block using deformable convolution and various attention modules to learn from the attentive and adaptive receptive field. Based on the novel blocks, we propose DANet for JPEG artifacts removal, as shown in Sections 3 and 4.

2.3. Super resolution

Early super resolution approaches were developed based on the fixed sampling theory and interpolation methods. In CNN-based methods, SRCNN [11] first introduced the deep learning method to the single image super resolution problem. To use deeper CNN models, VDSR [7] proposed a deep model for single image super resolution. SRRESNET [34] added residual structure and generative adversarial training to the model. Furthermore, RCAN [13] introduced a channel-wise attention mechanism for single image super resolution models. Recently, more and more IR tasks tend to use real image data for practical applications. In LP-KPN [35], a novel real image super resolution dataset(RealSR) and a Laplacian pyramid-based kernel prediction network (LP-KPN) were proposed. Instead of using the synthetic sampling image as low-resolution images, they used paired LR-HR images of the same scene. Real-world low-resolution (LR) images are more complicated than stimulated ones. Thus models trained on synthetic data are not robust on real-world data. Targeting the real-world image super resolution dataset is more practical in real scenarios. Here we apply the proposed IR models on a real image super resolution dataset for comparison.

2.4. Knowledge distillation

Knowledge distillation (KD) was first introduced as a model compression method [36] and further explored with transferring prior knowledge from a larger model (teacher model) to a smaller model (student model) [37]. In KD, a larger teacher model is trained with a larger dataset and a soft-target loss function. With a transfer training dataset, the student model learns the mapping of the teacher model's output distribution to achieve comparable performance with smaller models [38,39]. Meanwhile, many definitions of prior knowledge can also be transferred from the teacher model to the student model. In SCFace [40], the student model learns a selective mapping between high-resolution faces (generated by teacher model) and low-resolution faces by solving a sparse graph optimization problem. SNSR [41] proposed a distillation scheme to boost the performance, where they built a larger model as the teacher model to transfer learned features and knowledge to the student model. Similarly, PISR [42] proposed a knowledge distillation scheme based on the baseline model (FSRCNN [43]). However, they cannot capture the full-scale prior knowledge for IR because it used an unbalanced encoder–decoder structure model in the teacher model. Unlike PISR, we propose an asymmetrical structure of the teacher model based on DANet to acquire full-scale prior knowledge. PISR used distillation loss to collaboratively train the student model and transfer knowledge from the teacher model, which is suitable for hardware-friendly methods. We initialized the student model with the parameters from the teacher model's decoder. Then we finetuned the student model instead of collaborative training with the teacher model. As shown in the experiment section, the DANet-S achieves comparable performance with fewer parameters. Because of the large size of DANet, finetuning is more friendly in a limited hardware situation.

3. Proposed network

This section presents an overview of the proposed DANet and DANet-S for IR, including the models for synthetic and real image denoising, JPEG artifacts removal, and real-world super resolution. Fig. 2 illustrates the overall architecture of DANet, which consists of two novel blocks (*i.e.*, ADEB and AROB) and wavelet transform (*i.e.*, decomposition and reconstruction) modules. More concretely, (1) ADEB provides enriched contextual features from informative spatial details learned by adaptive and attentive receptive fields; (2) AROB is an attentive recurrent module built upon ADEB, maintaining the offset information learned from feature maps of different resolutions; (3) with the proposed knowledge distillation scheme, DANet-S can preserve a comparable performance as DANet with fewer parameters.

Formally, DANet learns a mapping function f_θ with a set of parameters θ by taking the noisy images/low resolution images (LR) $\{\hat{I}_i\}_{i=1}^N$, $\hat{I}_i \in \mathbb{R}^{H\times W\times C}$ (H as the height, W as the width and C as the channel of the image) as inputs and outputting the corresponding clean images/high resolution images (HR) $\{I_i\}_{i=1}^N$, $I_i \in \mathbb{R}^{H\times W\times C}$, with an ℓ_2 loss function:

$$\mathcal{L} = \arg\min_\theta \sum_{i=1}^{N} \|f_\theta(\hat{I}_i) - I_i\|^2. \tag{1}$$

In the following, we first briefly introduce the deformable convolution mechanics and then discuss the critical components of the proposed DANet in detail.

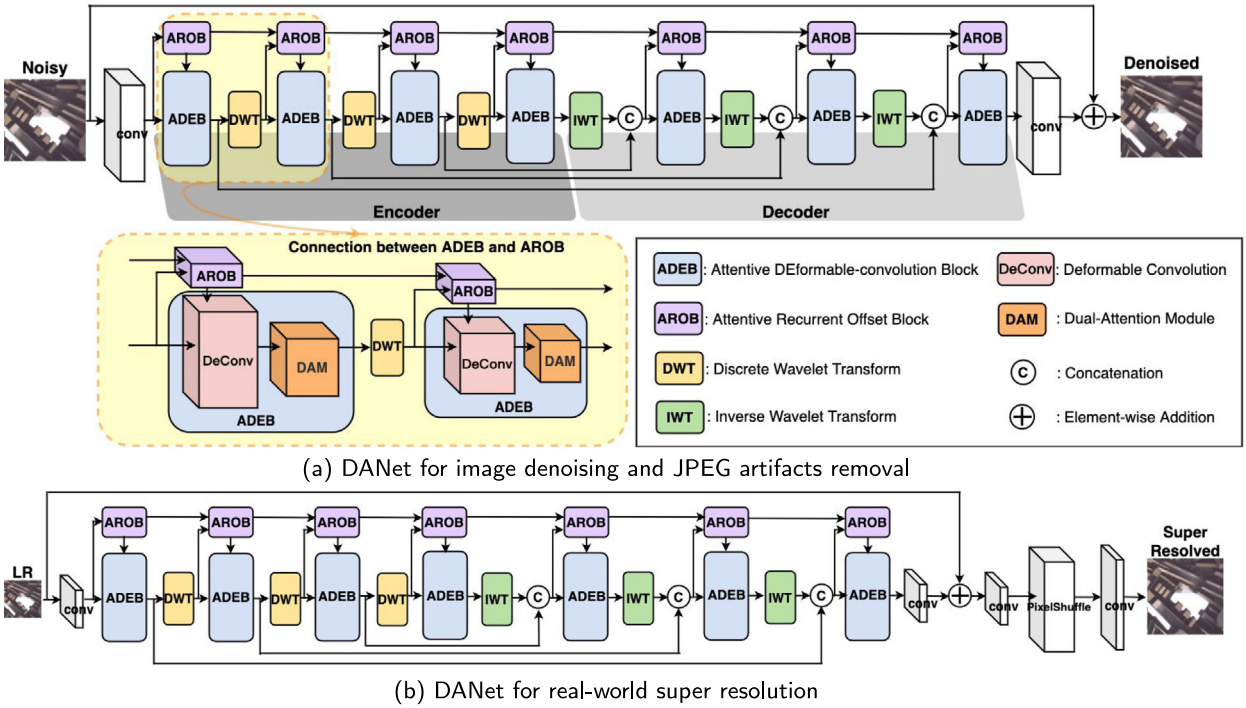


Fig. 2. An overview of the proposed DANet for image restoration.

3.1. Deformable convolution

Given an input feature map $x \in \mathbb{R}^{H\times W\times C}$, the standard convolution is typically performed on a regular grid $\mathcal{R}$ (which defines the receptive field size) over x:

$$y(p_0) = \sum_{p_n\in\mathcal{R}} w(p_n)\cdot x(p_0+p_n), \tag{2}$$

where y is the output feature map, p_0 indicates the location in the feature maps x and y, p_n indicates the location in the grid $\mathcal{R}$, and w is the weight.

On the other hand, the DeConv learns from an adaptive receptive field, with the grid $\mathcal{R}$ determined by a set of learn-able offsets. The DeConv [24] was first proposed for semantic segmentation and object detection, and then further improved [44]. In our work, we adopt the improved version of DeConv. Specifically, in addition to p_n in Eq. (4), we learn an extra set of offsets, *i.e.*, Δp_n $(n = 1,\ldots,N)$, where $N = |\mathcal{R}|$. In the meantime, we also learn a modulation scalar Δm_n for the nth location, which can modulate the input feature amplitudes from different spatial locations. Thus, the improved DeConv is formulated as follows:

$$y(p_0) = \sum_{p_n\in\mathcal{R}} w(p_n)\cdot x(p_0+p_n+\Delta p_n)\cdot \Delta m_n. \tag{3}$$

Based on the above formula, the DeConv layer can sample the feature map with a more spatially adaptive receptive field than the standard convolutional layer. In practice, two parallel convolutional layers are applied on the same input feature map to obtain the output feature map and the offset field, respectively.

In image denoising, since different pixels contribute differently to the output feature map [45], it is better to operate on more adaptive receptive fields to focus on the spatially meaningful features. To this end, we build two novel blocks based on DeConv, with ADEB extracting enriched and attentive features and AROB propagating offset fields throughout the whole network to preserve spatial details.

3.2. Attentive Deformable-Convolution Block (ADEB)

Based on the DeConv [44], we propose a novel block to generate more attentive features with an adaptive receptive field. Fig. 3 shows the proposed ADEB, which comprises a DeConv layer and a dual-attention module (DAM). Specifically, given an input feature map $x \in \mathbb{R}^{H\times W\times C}$, we first compute a new feature map $x_d \in \mathbb{R}^{H\times W\times C}$ as the input to the dual-attention module. As shown in Fig. 3, x_d is obtained after feeding x to a DeConv layer, several convolutional layers and the ReLU activation function. As a result, x_d represents more adaptive features from spatially important regions like edges. Subsequently, after respectively feeding x_d to the two distinct attention modules, we can obtain more informative features and suppress less useful features at both channel and spatial levels. Furthermore, several residual connections are also introduced to the ADEB to improve the information flow during the learning process.

The dual-attention module comprises channel-wise attention (CA) unit and spatial-wise attention (SA) unit, which exploits the inter-channel and inter-spatial dependencies of the convolutional feature maps. In practice, we adopt the attention mechanism similar to CA [46] and SA [47]. The detailed flow chart of these two units is also shown in Fig. 3. As can be seen, given the input feature map x_d, the CA unit performs average pooling over x_d to generate a channel-wise attention map $x'_{ca} \in \mathbb{R}^{1\times 1\times C}$. To further explore the channel-wise relationship, channel reduction and channel upsampling are applied (using several convolutional layers and the Sigmoid activation function). Finally, x_d is rescaled by this up-sampled attention map to yield the CA unit's final output. The SA unit, given x_d as the input feature map, first extracts the spatially dependent features using max/average pooling along the channel dimensions, followed by an unsqueeze operator (*i.e.*, several convolutional layers). Next, the outputs of these two branches are concatenated as a new feature map $x'_{sa1} \in \mathbb{R}^{H\times W\times 2}$. Finally, a spatial attention map $x'_{sa2} \in \mathbb{R}^{H\times W\times 1}$ is obtained by passing the above feature map through a convolutional layer and a Sigmoid activation, which rescale the input feature map x_d and generate the SA unit's final output.

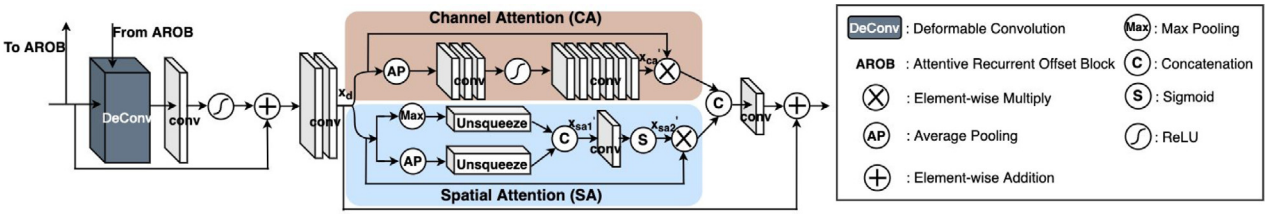


Fig. 3. An overview of the proposed Attentive DEformable-convolution Block (ADEB). ADEB combines the deformable convolution with spatial wise and channel wise attention module to generate more adaptive and more spatial accurate receptive fields in the IR model.

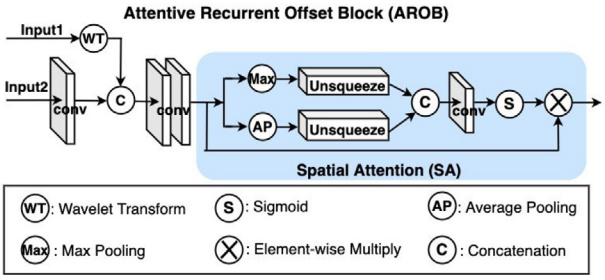
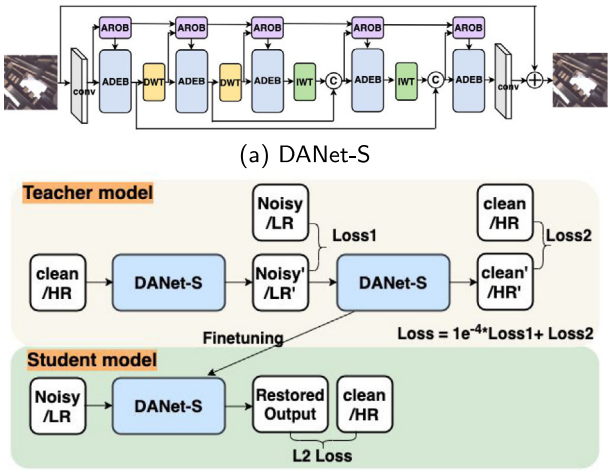


Fig. 4. An overview of Attentive Recurrent Offset Block (AROB). AROB connects and transfers multi-scale features between two consecutive ADEB blocks in DANet. The input1 represents the offset mask from the former AROB (for the first AROB in the model, there is no input1). The input2 represents the input feature map for the corresponding ADEB. The sizes of two inputs are aligned using wavelet transform (WT).



(b) A pipeline of the proposed knowledge distillation scheme

Fig. 5. An overview of the DANet-S and the proposed knowledge distillation scheme for training DANet-S.

3.3. Attentive Recurrent Offset Block (AROB)

As discussed above, the encoder–decoder structure tends to omit spatial details. Even with the help of DeConv, the details still gradually vanish because of downsampling. Especially when the network becomes deeper and deeper, this will lead to over smoothed results. We propose Attentive Recurrent Offset Block (AROB) to preserve more spatial details, which provides a consistent flow for the offset field across feature maps of various resolutions.

The original DeConv obtains the offset feature by applying a convolutional layer over the current input feature map. Therefore, if we adopt DeConv directly, the offset information learned from the preceding layers cannot be accessed by the subsequent layers. To address this limitation, we propose AROB to recurrently connect the ADEB block and transfer the offset feature across different resolution features.

More concretely, AROB receives two kinds of feature maps as inputs, *i.e.*, the offset field from the preceding AROB module and the feature map to be passed to the DeConv from the corresponding ADEB, as shown in Fig. 4. Then, we adjust the size of the offset feature (*i.e.*, "Input1" in Fig. 4) by using either discrete wavelet transform (DWT) during encoding or inverse wavelet transform (IWT) during decoding. We adopt DWT/IWT other than bilinear interpolation [24], considering its ability to preserve location information. After concatenating the two feature maps, we apply a spatial-attention module to obtain more spatially significant offsets. As shown in Fig. 4, DWT and IWT are also employed for decomposition/upsampling and reconstruction/downsampling in our main encoder–decoder structure. As an additional benefit, DWT/IWT can prevent the loss of details to some extent as it captures both the frequency and location information of feature maps.

3.4. Knowledge distillation scheme

To further reduce the parameters of the DANet with minor performance loss, we propose a simple distillation scheme to promote the efficiency of the model. DANet has three scales of modules embedded with ADEB and AROB. We constructed a lightweight version of DANet, DANet Slim (DANet-S), by reducing downsampling and upsampling modules. Since the performance of DANet-S falls due to parameter reduction, we propose a KD scheme to boost the DANet-S's performance. Fig. 5 shows the proposed KD pipeline. We employ two DANet-S cooperating as a degraded reconstructed structure as the teacher model. The teacher model takes a clean/high-resolution image as the input, and the first DANet-S performs as a degradation model to generate the noisy/low-resolution images. The generated noisy/low-resolution images are used as input for the second DANet-S to restore the clean/high-resolution image. The teacher model takes the constraint loss during training, which combines the two DANet-S to learn the prior knowledge within the degradation and the restoration process. We adopt this symmetric structure to capture the full-scale prior knowledge in different image restoration tasks.

The student model shares the same structure as the restoration model in the teacher model. The student model is initialized using the teacher model's restoration model to transfer the teacher's prior knowledge to the student. The student model is trained using the noisy/low-resolution and clean/high-resolution image pairs with L2 loss. While another knowledge distillation scheme PISR, which aims to boost hardware-friendly super resolution method, employs similar student model initialization and then collaboratively trains the student model with the distillation loss. The added distillation loss contributes a relatively small part of the loss, formalized as below:

$$Loss = L_{reconstruction} + \lambda \times L_{distillation}, \; where \; \lambda = 1e^{-6} \tag{4}$$

Table 1
Average PSNR (dB) and SSIM [48] results of different methods for image super resolution, tested on *Set5* dataset with scale factors scale = {2, 3, 4}.

Training dataset	*Scale*	BL	BL+PISR	BL+PISR_NC
DIV2K	×2	37.15 0.9568	37.33 0.9576	37.28 0.9573
	×3	33.15 0.9157	33.31 0.9179	33.24 0.9168
	×4	30.89 0.8748	30.95 0.8759	30.92 0.8755

The reconstruction loss employs the L2 loss, while the distillation loss measures the average mean loss of the intermediate features from the teacher model and student model. In comparison, our way of training the student model can not only bring performance gain but is also more friendly to a large-sized model like DANet in a limited hardware situation.

Furthermore, we remove the collaborative training (distillation loss) during the second phase of training the baseline model [42] (represent as BL) with PISR (represents as BL+PISR_NC, 'PISR_NC' represents the PISR scheme without collaborative training) to demonstrate the performance difference in Table 1. We can observe that removing the collaborative training only caused acceptable performance loss, but it also makes the scheme much more friendly to large-size models.

4. Experiments

In this section, we perform extensive experiments on diverse benchmark datasets to evaluate the proposed scheme's effectiveness on different tasks. Firstly, we give a brief description of training datasets and then provide the implementation details for (a) synthetic and realistic noise removal, (b) JPEG artifacts removal, and (c) image super resolution. Secondly, we compare the results with the SOTA methods in each IR task and provide a visual comparison. The trained models will be released along with the source code.

4.1. Implementation details

We implement the DANet as an end-to-end model using the PyTorch library [51]. During training, the initial learning rate is 1×10^{-4} and gradually decreased to 1×10^{-6}. The model is trained with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) for 100 epochs. We set the batch size to 12 and perform random horizontal and vertical flipping for data augmentation. The same settings are used in each IR task. During training DANet-S with the KD, the batch size is set to 5 and 12 for training the teacher model and the student model, respectively. The rest of the settings are the same as training DANet. The experiments are conducted on a PC with a single NVIDIA Tesla V100 GPU.

4.2. Synthetic and realistic noise removal

4.2.1. Synthetic noise removal

This section demonstrates the effectiveness of the proposed DANet for synthetic AWGN image denoising in gray- scale and RGB scale. We train our network only on the training set of the DIV2K [56], which consists of 800 high-resolution images. Image patches of 256 × 256 with an interval of 130 pixels were cropped and used during training.

On Grayscale Images. We first demonstrate the effectiveness of our DANet for synthetic image denoising on grayscale images corrupted by AWGN. Our trained models are evaluated on Set12, BSD68 [49], and Urban100 [50] datasets. To fully validate different methods' denoising abilities, we adopt AWGN with different noise levels (standard deviation σ), *e.g.*, σ=15, 25, 50, and 75. Table 2 summarized quantitative comparisons in terms of PSNR and SSIM [48], where we can see that our DANet performs favorably against the traditional as well as recent SOTA CNN based denoising algorithms at all noise levels. Specifically, compared to the previously best methods SADNet, our algorithm achieves a performance gain of 0.1 to 0.2 dB in PSNR at each noise level on different test sets. Moreover, our DANet demonstrates superior performance, especially in the Urban100 test set, where the images contain more structural texture details. Furthermore, with the proposed knowledge distillation scheme, the DANet-S shows comparable performance with fewer parameters than DANet.

Additionally, Fig. 6 shows the visual comparison of DANet with recent leading denoising methods. We can observe that BM3D and DnCNN lose fine details, MWCNN, RIDNet, and SADNet obtain blurred edges, while our DANet preserves a clear line similar to the ground truth. Therefore, DANet successfully reconstructs the structural content and fine details of the corrupted image. Especially in textual abundant images, DANet shows superior performance. As shown in Fig. 7, we can observe that the DANet still keeps an outstanding performance among the comparison methods in restoring the subtle line in the image.

On RGB Images. In addition to grayscale images, we also evaluate the performance of DANet on RGB ones. Quantitative comparisons in terms of PSNR and SSIM are summarized in Table 3. As can be seen, DANet achieves the highest results compared to SOTA competitors. Specifically, on Koda24 and CBSD68, DANet obtains 0.1 to 0.3 dB higher in PSNR than the second-best method, *i.e.*, SADNet, which also adopts the DeConv for image denoising but in a more straightforward way. The performance gain on Urban100 is even higher (∼0.5 dB in PSNR *w.r.t.* different noise levels). As shown in Fig. 8, the DANet still outperforms other comparison methods in visual quality, in which both DANet and DANet-S can restore cleaner image texture details without any artifacts.

4.2.2. Realistic noise removal

We compare the proposed DANet with several SOTA realistic noise removal approaches. We train on SIDD medium [18] training set, and test on SIDD [18], DND [53] test sets. SIDD [18], mainly collected with smartphone cameras, contains 320 high-resolution image pairs for training and 1280 patches of 512 × 512 for testing. DND [53] consists of 50 high-resolution images, which are cropped into patches of 512 × 512 for testing. Without ground truth images (the ground truth images are not publicly available), the PSNR and SSIM are acquired by submitting the denoised images to the official website of DND dataset. Quantitative comparisons in terms of PSNR and SSIM [48] metrics are summarized in Table 4. As can be observed, our DANet outperforms the existing methods. For instance, compared with VDN and RIDNet, the PSNR results of DANet are 0.5 to 0.6 dB higher on SIDD and 0.2 to 0.3 dB higher on DND.

Moreover, we also visually depict some denoising results on these two real image datasets, as shown in Figs. 9 and 10. It can easily be observed from Fig. 9 that the proposed DANet recovers much cleaner edges and preserves finer image details than other competitors. In Fig. 10, compared to several SOTA methods, DANet preserves clearer textural details (such as the edges) and structural content.

4.3. JPEG artifacts removal

This section demonstrates our algorithm's effectiveness on JPEG artifacts removal. To train the model, we use only DIV2K [56] (the same settings as synthetic noise removal task) training set and test on CLASSIC5 and LIVE1 [57] datasets. Quantitative comparisons in terms of PSNR and SSIM [48] metrics are summarized

Table 2
Average PSNR (dB) and SSIM [48] results of different denoising methods for AWGN denoising on gray-scale images, tested on *Set12*, *BSD68* [49], *Urban100* [50] datasets with AWGN levels $\sigma = \{15, 25, 50, 75\}$. The best and the second best results are highlighted in red and blue, respectively.

Dataset	σ	Method / metric	BM3D [4]	DnCNN [6]	FFDNet [29]	MWCNN [20]	SADNet [17]	RIDNet [16]	DANet	DANet-S
Set12	*15*	PSNR↑	32.37	32.86	32.75	33.15	33.18	32.91	**33.29**	**33.26**
		SSIM↑	0.8952	0.9027	0.9024	0.9088	0.9127	0.9085	**0.9141**	**0.9137**
	25	PSNR↑	29.97	30.43	30.43	30.79	30.88	30.60	**31.01**	**30.99**
		SSIM↑	0.8505	0.8618	0.8631	0.8711	0.8752	0.8694	**0.8772**	**0.8767**
	50	PSNR↑	26.72	27.18	27.32	27.74	27.80	27.43	**27.97**	**27.92**
		SSIM↑	0.7676	0.7827	0.7899	0.8056	0.8069	0.7944	**0.8109**	**0.8093**
	75	PSNR↑	24.91	25.20	25.49	25.88	26.02	25.72	**26.14**	**26.09**
		SSIM↑	0.6950	0.7095	0.7355	0.7487	0.7542	0.7406	**0.7579**	**0.7561**
BSD68	*15*	PSNR↑	31.08	31.72	31.64	31.86	31.89	31.81	**31.95**	**31.93**
		SSIM↑	0.8722	0.8906	0.8902	0.8947	0.9008	0.8982	**0.9015**	**0.9013**
	25	PSNR↑	28.57	29.23	29.19	29.41	29.46	29.34	**29.52**	**29.51**
		SSIM↑	0.8017	0.8278	0.8288	0.8360	0.8431	0.8381	**0.8448**	**0.8443**
	50	PSNR↑	25.62	26.23	26.29	26.48	26.50	26.40	**26.62**	**26.59**
		SSIM↑	0.6869	0.7189	0.7239	0.7366	0.7382	0.7314	**0.7436**	**0.7427**
	75	PSNR↑	24.21	24.64	24.79	24.98	25.05	24.89	**25.10**	**25.08**
		SSIM↑	0.6139	0.6401	0.6577	0.6707	0.6742	0.6639	**0.6775**	**0.6750**
Urban100	*15*	PSNR↑	32.34	32.67	32.42	33.17	33.21	33.09	**33.70**	**33.65**
		SSIM↑	0.9220	0.9250	0.9273	0.9088	0.9104	0.9364	**0.9425**	**0.9422**
	25	PSNR↑	29.70	29.97	29.92	30.66	30.71	30.53	**30.92**	**31.40**
		SSIM↑	0.8777	0.8792	0.8887	0.9026	0.9033	0.9009	**0.9132**	**0.9145**
	50	PSNR↑	25.94	26.28	27.65	27.42	27.75	27.05	**28.29**	**28.17**
		SSIM↑	0.7791	0.7869	0.8057	0.8371	0.8380	0.8242	**0.8569**	**0.8545**
	75	PSNR↑	23.91	23.94	24.51	25.52	25.95	25.22	**26.38**	**26.22**
		SSIM↑	0.6950	0.6989	0.7367	0.7810	0.7958	0.7639	**0.8070**	**0.8013**

Table 3
Average PSNR (dB) and SSIM [48] results of different denoising methods for AWGN denoising on RGB-scale images, tested on *Koda24*, *CBSD68* [49], *Urban100* [50] datasets with AWGN levels $\sigma = \{15, 25, 50, 75\}$. The best and the second best results are highlighted in red and blue, respectively.

Dataset	σ	Method / metric	CBM3D [52]	CDnCNN [6]	FFDNet [29]	MWCNN [20]	SADNet [17]	RIDNet [16]	DANet	DANet-S
Koda24	*15*	PSNR↑	34.44	34.85	34.73	35.03	35.11	34.91	**35.31**	**35.24**
		SSIM↑	0.9192	0.9233	0.9224	0.9293	0.9303	0.9272	**0.9321**	**0.9315**
	25	PSNR↑	31.86	32.35	32.24	32.58	32.69	32.32	**32.91**	**32.84**
		SSIM↑	0.8700	0.8812	0.8799	0.8898	0.8922	0.8842	**0.8952**	**0.8942**
	50	PSNR↑	28.65	29.16	29.08	29.55	29.65	29.27	**29.90**	**29.81**
		SSIM↑	0.7762	0.7985	0.7971	0.8143	0.8169	0.8034	**0.8225**	**0.8214**
	75	PSNR↑	26.90	27.05	27.33	27.97	28.06	27.68	**28.26**	**28.20**
		SSIM↑	0.7109	0.7186	0.7386	0.7617	0.7643	0.7505	**0.7701**	**0.7684**
CBSD68	*15*	PSNR↑	33.33	33.99	33.84	33.87	33.93	33.79	**34.04**	**34.01**
		SSIM↑	0.9238	0.9303	0.9288	0.9330	0.9338	0.9315	**0.9348**	**0.9344**
	25	PSNR↑	30.63	31.31	31.20	31.30	31.39	31.11	**31.51**	**31.47**
		SSIM↑	0.8698	0.8848	0.8825	0.8896	0.8921	0.8856	**0.8934**	**0.8928**
	50	PSNR↑	27.35	28.01	27.96	28.17	28.25	27.98	**28.39**	**28.34**
		SSIM↑	0.7647	0.7925	0.7910	0.8044	0.8072	0.7969	**0.8101**	**0.8103**
	75	PSNR↑	25.63	26.02	26.23	26.56	26.63	26.37	**26.73**	**26.70**
		SSIM↑	0.6903	0.7115	0.7240	0.7425	0.7453	0.7352	**0.7493**	**0.7472**
Urban100	*15*	PSNR↑	33.90	34.12	33.94	34.30	34.49	34.28	**34.90**	**34.79**
		SSIM↑	0.9425	0.9436	0.9429	0.9488	0.9503	0.9477	**0.9529**	**0.9522**
	25	PSNR↑	31.44	31.66	31.50	31.94	32.19	31.66	**32.74**	**32.61**
		SSIM↑	0.9111	0.9145	0.9136	0.9232	0.9265	0.9180	**0.9315**	**0.9302**
	50	PSNR↑	28.05	28.16	28.10	28.79	29.10	28.41	**29.80**	**29.55**
		SSIM↑	0.8401	0.8490	0.8485	0.8710	0.8769	0.8589	**0.8888**	**0.8840**
	75	PSNR↑	25.97	25.29	26.05	27.12	27.40	26.73	**28.03**	**27.85**
		SSIM↑	0.7754	0.7637	0.7907	0.8315	0.8379	0.8168	**0.8517**	**0.8477**

Table 4
Average PSNR (dB) and SSIM [48] of different methods on the *SIDD* test set [18] and the *DND* dataset [53]. The best and the second best results are highlighted in red and blue, respectively. "NA" means "Not Available" due to unavailable code or model.

Dataset	Method / metric	CBM3D [52]	TNRD [54]	MLP [55]	DnCNN [6]	FFDNet [29]	CBDNet [31]	SADNet [17]	RIDNet [16]	VDN [15]	DANet	DANet-S
SIDD [18]	PSNR↑	25.65	24.73	24.71	23.66	NA	30.78	39.36	38.71	39.28	**39.87**	**39.70**
	SSIM↑	0.685	0.643	0.641	0.583	NA	0.754	NA	0.914	0.909	**0.915**	**0.913**
DND [53]	PSNR↑	34.51	33.65	34.23	37.90	37.61	38.06	39.37	39.26	39.38	**39.50**	**39.39**
	SSIM↑	0.851	0.831	0.833	0.943	0.942	0.942	**0.954**	0.952	0.952	**0.953**	0.951

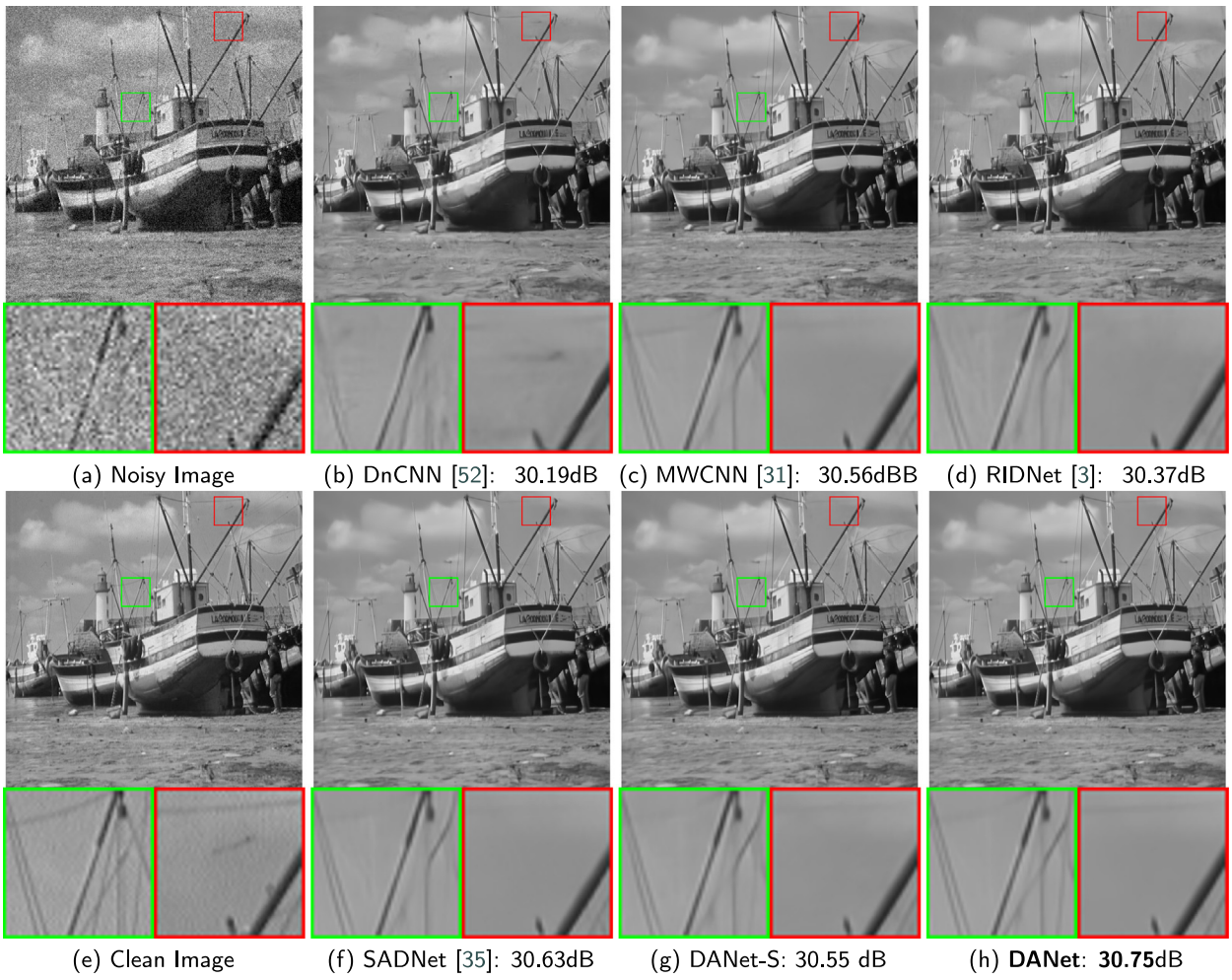


Fig. 6. Denoising results of a typical image ("boats.png" from *set12* test set) using different denoising methods corrupted by AWGN noise ($\sigma = 25$). The best result is highlighted in **bold**.

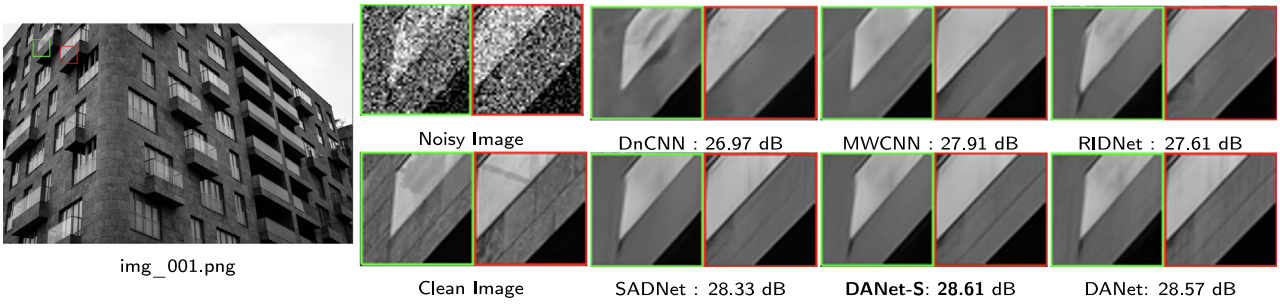


Fig. 7. Synthetic denoising results of a typical grayscale image ("img_001.png" from *Urban100* test set) using different denoising methods corrupted by AWGN (σ = 50). The best result is highlighted in **bold**.

in Table 5. We can see that our proposed method achieves significant improvements over SOTA approaches. For instance, compare with RNAN [30] and MWCNN [20], DANet achieves superior performance on both test sets. Visual comparison in Fig. 11 shows that MWCNN and RNAN have the over smoothing problem while DANet can preserve more subtle texture details.

4.4. Image super resolution

In this section we compare our proposed method against the SOTA SR algorithms (VDSR [7], SRResNet [34], RCAN [13], LP-KPN [35]) on the RealSR testing images with upscaling factors of $\times 2$, $\times 3$ and $\times 4$. Note that all the benchmark algorithms are trained on the RealSR [35] dataset for a fair comparison (the comparing results are also provided from RealSR [35]). RealSR [35] contains real-world LR-HR image pairs of the same scene captured by adjusting the cameras' focal length. In RealSR, the number of training image pairs for scale factors $\times 2$, $\times 3$ and $\times 4$ are 183, 234, and 178, respectively. Moreover, the dataset also provides 30 additional test images for each scale factor. Here, we compute the PSNR and SSIM [48] using the Y channel (in YCbCr color space), as it is a common practice in existing SR methods [7,13,34]. The results in Table 6 show that DANet achieves a clear advantage over the other compared methods. The

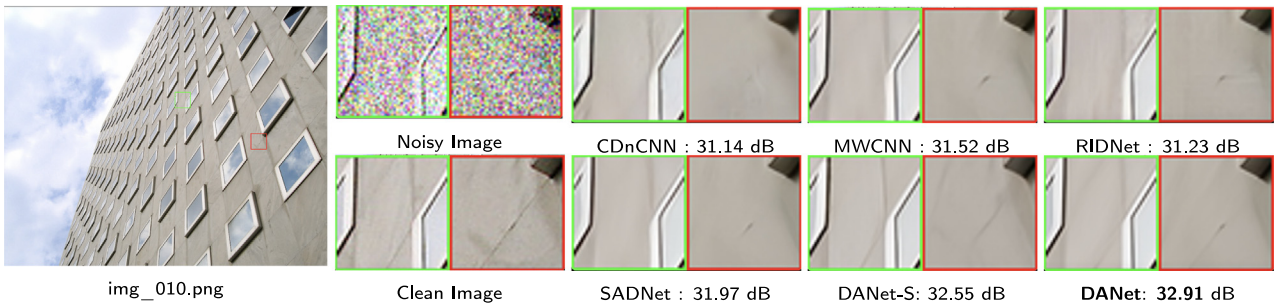


Fig. 8. Synthetic denoising results of a typical RGB image ("img_010.png" from *Urban100* test set) using different denoising methods, corrupted by AWGN (σ = 50). The best result is highlighted in **bold**.

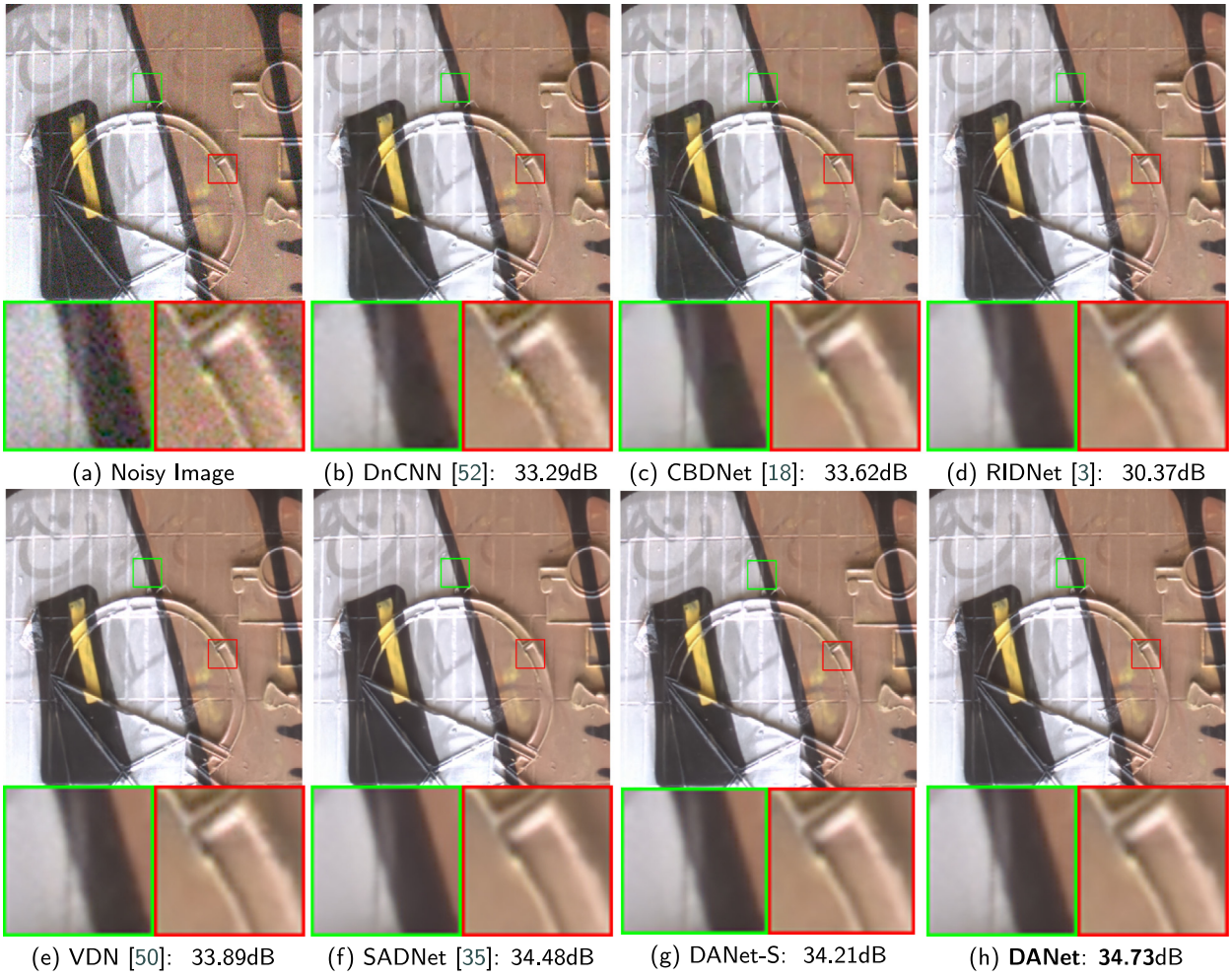


Fig. 9. Denoising results of a typical image ("*0002_19.png*" from *DND* [53] test set) using different denoising methods. The best result is highlighted in **bold**.

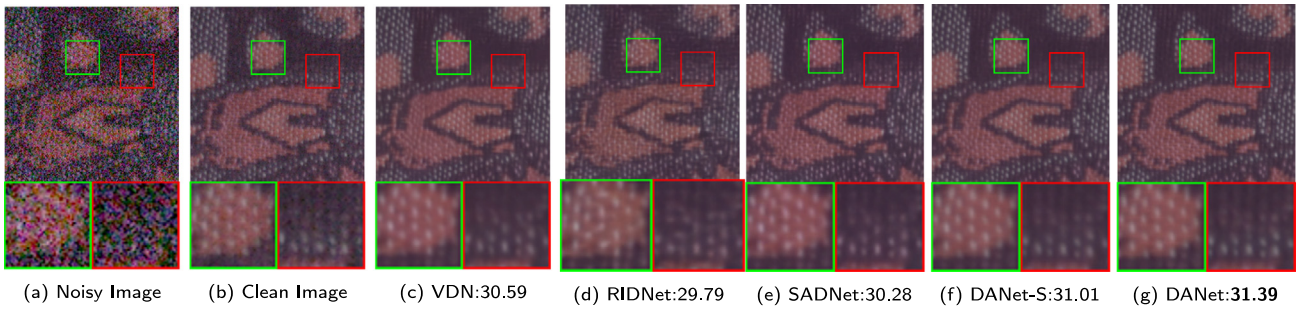
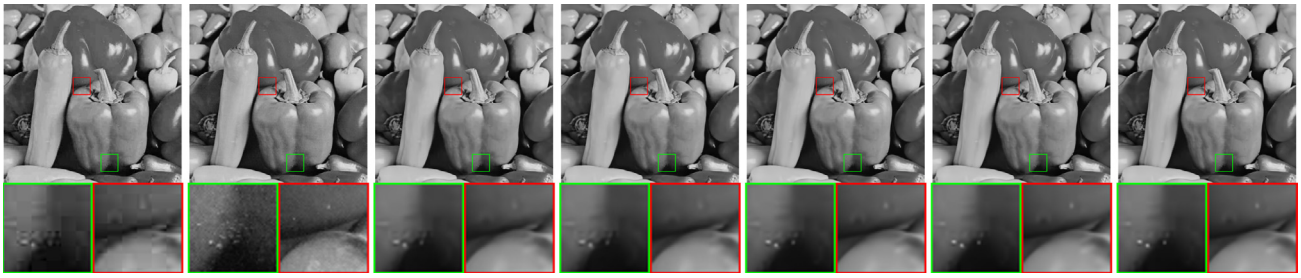


Fig. 10. Denoising results of a typical image ("*39_9.png*" from *SIDD* [18] test set) using different denoising methods. The best result (PSNR in dB) is highlighted in **bold**.



(a) JPEG (Q=20) (b) Clean Image (c) DnCNN: 34.10 (d) RNAN:34.30 (e) MWCNN:34.36 (f) DANet-S:34.43 (g) DANet: **34.44**

Fig. 11. JPEG artifacts removal (Q=20) results of a typical image ("*peppers*" from *Classic5* test set) using different JPEG artifacts removal methods. The best result (PSNR in dB) is highlighted in **bold**.

Table 5
Average PSNR (dB) and SSIM [48] results of different methods for JPEG artifacts removal, tested on *Classic5*, *LIVE1* [57] datasets with quality factors = {10, 20, 30, 40}. The best and the second best results are highlighted in red and blue, respectively. "NA" means "Not Available" due to unavailable code or model.

Dataset	Q	Method metric	JPEG	SA-DCT [5]	ARCNN [32]	TNRD [54]	DnCNN [6]	MemNet [22]	RNAN [30]	MWCNN [20]	DANet	DANet-S
Classic5	*10*	PSNR↑	27.82	28.88	29.03	29.28	29.04	29.69	29.96	30.01	**30.27**	**30.18**
		SSIM↑	0.7595	0.8071	0.7929	0.7992	0.8026	0.8107	0.8178	0.8195	**0.8333**	**0.8311**
	20	PSNR↑	30.12	30.92	31.15	31.47	31.63	31.90	32.11	32.16	**32.45**	**32.36**
		SSIM↑	0.8344	0.8663	0.8517	0.8576	0.8610	0.8658	0.8693	0.8701	**0.8821**	**0.8810**
	30	PSNR↑	31.48	32.14	32.51	32.78	32.91	NA	33.38	33.43	**33.66**	**33.61**
		SSIM↑	0.8744	0.8914	0.8806	0.8837	0.8861	NA	0.8924	0.8930	**0.9033**	**0.9027**
	40	PSNR↑	32.43	33.00	33.34	NA	33.96	NA	34.27	34.27	**34.45**	**34.43**
		SSIM↑	0.8911	0.9055	0.8953	NA	0.9247	NA	0.9061	0.9061	**0.9150**	**0.9148**
LIVE1	*10*	PSNR↑	27.77	28.65	28.96	29.15	29.19	29.45	29.63	29.69	**30.32**	**30.27**
		SSIM↑	0.7595	0.8093	0.8076	0.8111	0.8123	0.8193	0.8239	0.8254	**0.8343**	**0.8334**
	20	PSNR↑	30.07	30.81	31.29	31.46	31.59	31.83	32.03	32.04	**32.66**	**32.62**
		SSIM↑	0.8512	0.8781	0.8733	0.8769	0.8802	0.8846	0.8877	0.8885	**0.8969**	**0.8963**
	30	PSNR↑	31.41	32.08	32.67	32.84	32.98	NA	33.45	33.45	**34.06**	**33.97**
		SSIM↑	0.9000	0.9078	0.9043	0.9059	0.9090	NA	0.9149	0.9153	**0.9222**	**0.9203**
	40	PSNR↑	32.35	32.99	33.63	NA	33.96	NA	34.47	34.45	**35.05**	**34.95**
		SSIM↑	0.9173	0.9240	0.9198	NA	0.9247	NA	0.9061	0.9301	**0.9360**	**0.9327**

Table 6
Average PSNR (dB) and SSIM [48] results of different methods for real-world super resolution, tested on *RealSR* [35] dataset with scale factors *scale* = {2, 3, 4}. The best and the second best results are highlighted in red and blue, respectively. "NA" means "Not Available" due to unavailable code or model.

Dataset	*Scale*	Method metric	Bicubic	VDSR [7]	SRResNet [34]	RCAN [13]	LP-KPN [35]	DDet [58]	CDC [59]	DANet	DANet-S
RealSR	×2	PSNR↑	32.61	33.64	33.69	33.87	33.90	33.22	33.96	**34.25**	**34.13**
		SSIM↑	0.907	0.917	0.919	0.922	**0.927**	NA	0.925	**0.930**	0.924
	×3	PSNR↑	29.34	30.14	30.18	30.40	30.42	30.62	**30.99**	**30.99**	**30.95**
		SSIM↑	0.841	0.856	0.859	0.862	0.868	NA	**0.869**	**0.872**	0.850
	×4	PSNR↑	27.99	28.63	28.67	28.88	28.92	28.94	**29.24**	**29.30**	29.09
		SSIM↑	0.806	0.821	0.824	0.826	**0.834**	NA	0.827	**0.841**	0.832

Table 7
Parameters, FLOPs and running time for synthetic noise removal task on image of size 480 × 320 (from BSD68 test set).

	DnCNN	MWCNN	SADNet	RIDNet	DANet	DANet-S
Params.	558k	24930k	4234k	1499k	23481k	15940k
FLOPs	86.1G	159.3G	50.1G	230.0G	1040.9G	571.1G
times(ms)	21.3	85.6	26.7	84.4	225.3	169.0
PSNR(dB)	29.23	29.41	29.46	29.34	29.52	29.51

Table 8
Ablation experiments on the modules in DANet for synthetic noise removal task.

DeConv	ADEB	AROB	WT	PSNR(dB)/SSIM
×	×	×	×	30.15/0.8556
✓	×	×	×	30.81/0.8690
×	✓	×	×	30.91/0.8754
×	✓	✓	×	30.96/0.8767
×	✓	✓	✓	31.01/0.8772

PSNR improvement is around 0.4 to 0.5 dB comparing with LP-KPN and RCAN. The visual comparison in Fig. 12 further proves the advantage of DANet on restoring structural details. Our DANet and DANet-S can preserve clearer edges in the restored results, while RCAN and LP-KPN restore blurry edges and lack texture details.

5. Ablation studies

This section studies the impact of our architectural components on the final model performance. All the ablation experiments are performed for the synthetic noise removal task on

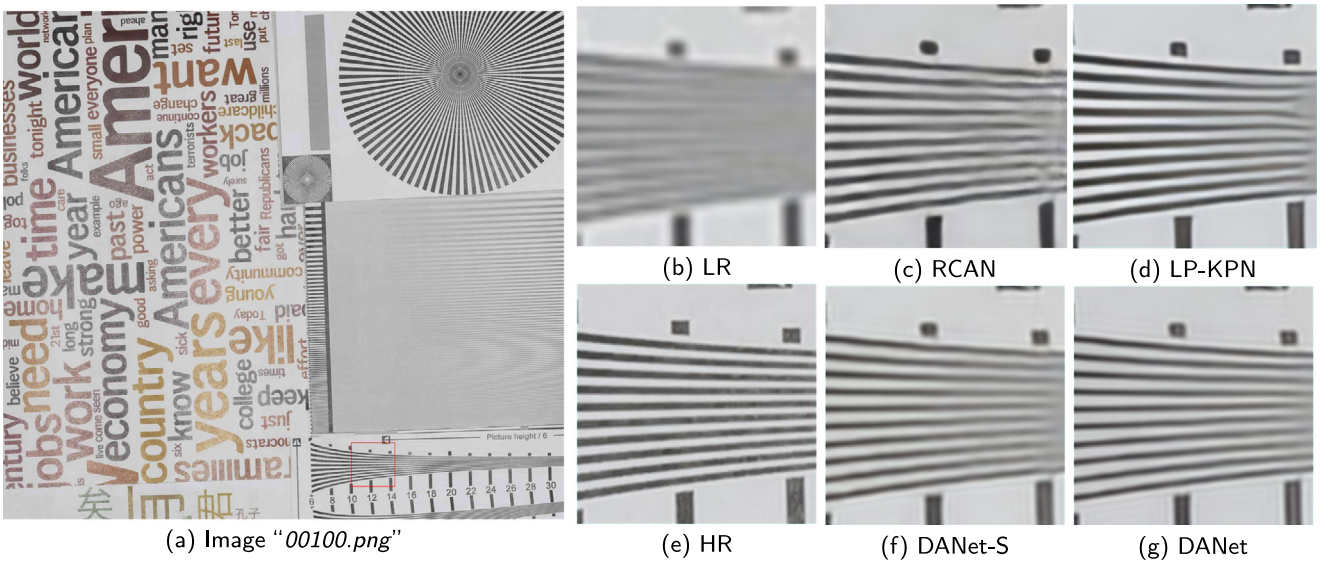


Fig. 12. Super resolution results of a typical image ("*00100.png*" from *RealSR* [35] test set) using different super resolution methods.

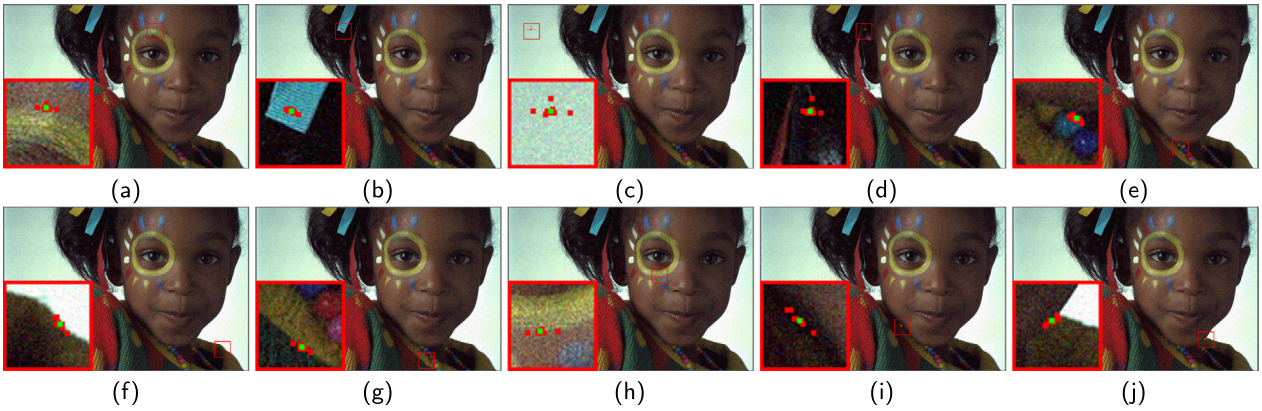


Fig. 13. Visualization of the sampling locations in the learnt adaptive receptive fields.

AWGN noise level $\sigma = 25$, using Nvidia cuDNN-v7.0 deep learning library under Ubuntu 16.04 system.

5.1. Model size, complexity and test time

Table 7 lists the model parameter size, the number of FLoating-point OPerations(FLOPs), and the GPU run time of the competing methods on denoising tasks for comparison. FLOPs can measure the number of floating point operations an algorithm required to solve a problem and compare the relative speed of methods, which are widely used in the CNN-based image restoration methods comparison. From Table 8, we can observe that despite the superior performance of DANet, the parameter size, FLOPs, and run-time evaluation demonstrate that it is also a ponderous model. The proposed DANet has a larger model size, and it is more computational complex than the comparison methods due to the multiple uses of dual-attention modules and deformable convolution modules. To further optimize this, we proposed a knowledge distillation scheme to refine DANet and train a light weighted version DANet, DANet-S. After applying the proposed knowledge distillation method, the DANet-S can retain superior performance with an acceptable parameter size, while the FLOPs and run time still need to be optimized.

5.2. Ablation on the proposed modules

Table 8 shows the detailed comparison results. In particular, we first adopt a baseline U-Net architecture (mostly based on stacked convolutional layers) without using any proposed components. In the second row in the table, we add DeConv as SADNet did, and the performance is relatively poor. Subsequently, we gradually add the proposed components to see how the performance changes. Based on the results from the third to last rows, we have the following observations: (1) Adding ADEB instead of directly applying DeConv has the most considerable effect on the overall performance, with the model even outperforming some recent CNN based approaches (such as RIDNet in Table 2), which aligns well with our motivations and indicates the importance of learning adaptive receptive fields. Comparing with directly adding DeConv in the structure like SADNet, our ADEB further improve the performance. (2) The proposed AROB, to cooperate with ADEB, further improve the overall performance. The results show that building attentive recurrent offset features for ADEB is beneficial to image denoising. (3) After incorporating all the proposed components, we introduced DWT/IWT modules in the model to further improve the results. Furthermore, the final model acquired a performance increase over the baseline U-Net structure and U-net structure with DeConv (*e.g.*, 0.86 dB and 0.2 dB in PSNR, respectively). Once again, it verifies the

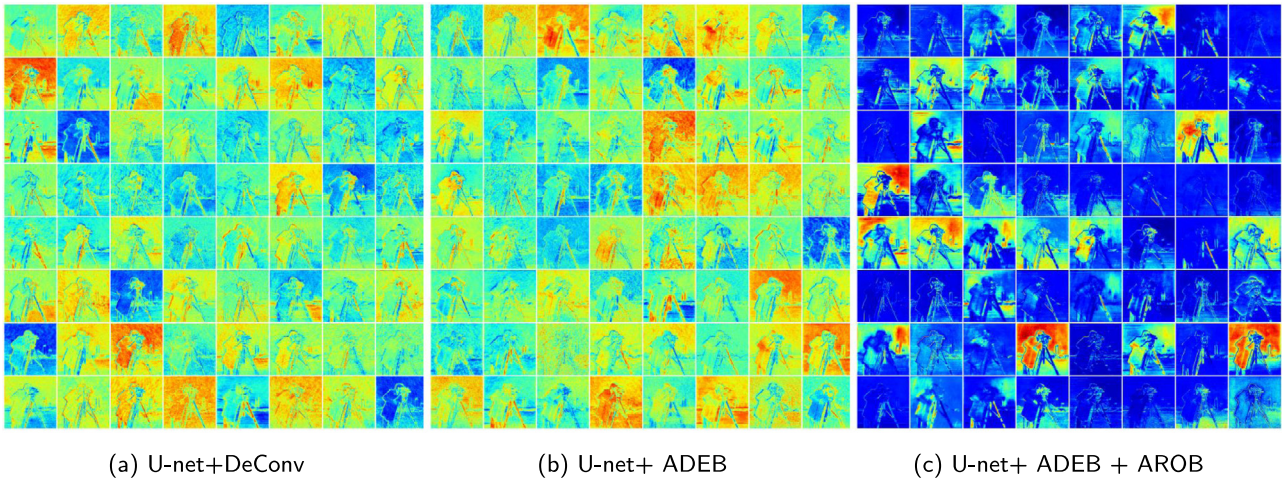


Fig. 14. Visualization of offset transferred feature generated in the last block by (a) U-net+ DeConv, (b) U-net+ ADEB and (c) U-net+ADEB+AROB on the gray scale image "*cameraman.png*" (from Set12) for synthetic noise removal.

importance of all the components and the superiority of the overall architecture of DANet.

5.3. Visualization of adaptive receptive fields

To better understand the adaptive receptive fields learned by the proposed DANet, we show the obtained sampling locations around specific regions on a typical image in Fig. 13. In simple convolution, the sampling location around one point should be in a square. With the proposed module, DANet can acquire attentive, adaptive receptive fields. We can observe from the visualization of the receptive fields in DANet that the learned sampling locations tend to gather around the fine spatial details, *e.g.*, the yellow edge in Fig. 13(a) and the textural band in Fig. 13(b). For less critical regions on the image, *e.g.*, the background area in Fig. 13(c) and the less textual hair in Fig. 13(d), the receptive fields radiate out to capture more contextual information and structural content surrounding the target pixel. All the above observations demonstrate the superior capability of DANet, *i.e.*, attending to fine spatial details and capturing rich contextual features. Fig. 13(e) to (j) show that the sampling location can be adaptive to the image content and texture, which can prove the effectiveness of the proposed model. In this case, the proposed model can preserve more attentive texture information during IR.

5.4. Visualization of feature transfer

To explore the benefits of employing the original DeConv and the proposed ADEB and AROB in an encoder–decoder structure, we visualized the feature maps generated by the DeConv, ADEB, and ADEB with AROB in the IR model. In Fig. 14, we chose three structures: (1) U-net with DeConv in the decoder (the structure proposed in SADNet), (2) U-net with the proposed ADEB modules (applied in the encoder and decoder), and (3) U-net with ADEB and AROB modules in the encoder and decoder. The visualization of the offset branch feature maps in the last block of the model shows the effectiveness of AROB in propagating multi-scale features through the IR model. The proposed attentive and recurrent offset branch is transferred from local and former features to generate more sparse and adaptive receptive fields. From Fig. 14(a) to (c), we can see that with adding the attention modules and recurrent connection, the feature maps show progressive enhanced contextual edges of the image. Especially from Fig. 14(b) to (c), a more sparse and attentive offset feature is generated.

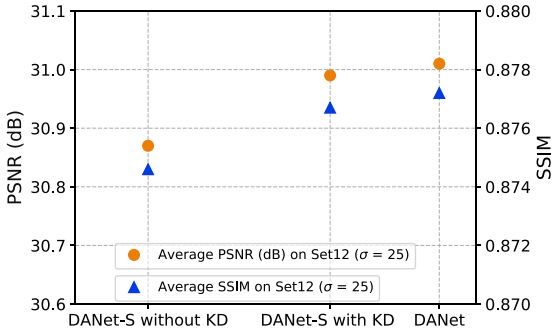


Fig. 15. Average PSNR (dB) and SSIM results of Set12 on synthetic image denoising corrupted by AWGN noise ($\sigma = 25$), which compare the performance of DANet-S training with and without the proposed knowledge distillation scheme. .

5.5. Benefits of the proposed knowledge distillation scheme

In this section, we discover the performance gain brought by the proposed knowledge distillation scheme for the DANet. Fig. 15 shows a comparison of training a single DANet-S with and without the knowledge distillation scheme. After applied the KD, the performance is improved by 0.11 dB and achieves comparable performance with DANet, which has deeper models and more extensive parameters.

6. Conclusion

This paper proposes a novel neural network named DANet for image restoration tasks. We equip the conventional encoder–decoder structure with the additional capabilities of preserving fine spatial details by learning adaptive and attentive features. Motivated by the success of deformable convolution, we propose two novel blocks (*i.e.*, the ADEB and AROB) to learn informative contextual features and capture important local details (such as edges and textures). With the additional help of wavelet transform, our overall framework consistently achieves SOTA performance in synthetic and real noise removal, JPEG artifacts removal, and real image super resolution tasks on several popular test sets. To further reduce the parameter size of the DANet, we propose a KD scheme and a light-weighted DANet (DANet-S), which achieves a comparable performance with DANet. Since the test time still needs optimization, we will improve this in future work.

CRediT authorship contribution statement

Yuan Huang: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing. **Xingsong Hou:** Conceptualization, Writing - Validation, Supervision, Writing – original draft, Writing – review & editing. **Yujie Dun:** Writing – original draft, Writing – review & editing. **Jie Qin:** Writing – original draft, Formal analysis. **Li Liu:** Writing – review & editing. **Xueming Qian:** Writing – review & editing. **Ling Shao:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.knosys.2021.107384. In this supplementary document, we provide: (a) more qualitative results of different methods for grayscale image denoising on the *BSD68* and *Urban100* test sets, (b) more qualitative results of different methods for RGB image denoising on the *CBSD68* and *Urban100* test sets, (c) more qualitative results of different methods for real image denoising on the *DND* and *SIDD* test sets, (d) more qualitative results of different methods for JPEG artifacts removal on Classic5 test sets, (e) add reproduced results for comparison to replace the 'NA' in Table 5 of the main paper. These reproduced results, consistent with the original papers' results, are marked with $*$.

References

[1] H. Jiang, G. Zhai, H. Cai, J. Yang, Scalable motion analysis based surveillance video de-noising, in: 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2018, pp. 1–6.

[2] L. Gondara, Medical image denoising using convolutional denoising autoencoders, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), IEEE, 2016, pp. 241–246.

[3] Z. Huang, Y. Zhang, Q. Li, T. Zhang, N. Sang, H. Hong, Progressive dual-domain filter for enhancing and denoising optical remote-sensing images, IEEE Geosci. Remote Sens. Lett. 15 (5) (2018) 759–763.

[4] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[5] Y. Li, F. Guo, R. Tan, M. Brown, A contrast enhancement framework with JPEG artifacts suppression, in: European Conference on Computer Vision, Springer, 2014, pp. 174–188.

[6] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, IEEE Trans. Image Process. 26 (7) (2017) 3142–3155.

[7] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.

[8] G. Li, Y. Yang, X. Qu, D. Cao, K. Li, A deep learning based image enhancement approach for autonomous driving at night, Knowl.-Based Syst. (2020) 106617.

[9] C. Tian, R. Zhuge, Z. Wu, Y. Xu, W. Zuo, C. Chen, C. Lin, Lightweight image super-resolution with enhanced CNN, Knowl.-Based Syst. 205 (2020) 106235.

[10] Y. Dun, Z. Da, S. Yang, Y. Xue, X. Qian, Kernel-attended residual network for single image super-resolution, Knowl.-Based Syst. (2020) 106663.

[11] C. Dong, C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer, 2014, pp. 184–199.

[12] J. Kim, J. Kwon Lee, K. Mu Lee, Deeply-recursive convolutional network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1637–1645.

[13] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 286–301.

[14] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image restoration, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2020) 1.

[15] Z. Yue, H. Yong, Q. Zhao, D. Meng, L. Zhang, Variational denoising network: Toward blind noise modeling and removal, in: Advances in Neural Information Processing Systems, 2019, pp. 1688–1699.

[16] S. Anwar, N. Barnes, Real image denoising with feature attention, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3155–3164.

[17] C. Meng, L. Qi, F. Huajun, X. Zhihai, Spatial-adaptive network for single image denoising, 2020, arXiv:2001.10291.

[18] A. Abdelhamed, S. Lin, M. Brown, A high-quality denoising dataset for smartphone cameras, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[19] X. Mao, C. Shen, Y. Yang, Image restoration using convolutional auto-encoders with symmetric skip connections, in: Advances in Neural Information Processing Systems, 2016.

[20] P. Liu, H. Zhang, K. Zhang, L. Lin, W. Zuo, Multi-level wavelet-CNN for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 773–782.

[21] P. Zhuang, X. Ding, Divide-and-conquer framework for image restoration and enhancement, Eng. Appl. Artif. Intell. 85 (Oct.) (2019) 830–844.

[22] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4539–4547.

[23] O. Kupyn, T. Martyniuk, J. Wu, Z. Wang, Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8878–8887.

[24] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.

[25] X. Wang, L. Zhu, Y. Wu, Y. Yang, Symbiotic attention for egocentric action recognition with object-centric alignment, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[26] H. Fan, L. Zhu, Y. Yang, F. Wu, Recurrent attention network with reinforced generator for visual dialog, ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 16 (3) (2020) 1–16.

[27] D. Yu, J. Fu, X. Tian, T. Mei, Multi-source multi-level attention networks for visual question answering, ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 15 (2s) (2019) 1–20.

[28] S. Yu, B. Park, J. Jeong, Deep iterative down-up CNN for image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[29] K. Zhang, W. Zuo, L. Zhang, Ffdnet: Toward a fast and flexible solution for CNN-based image denoising, IEEE Trans. Image Process. 27 (9) (2018) 4608–4622.

[30] Y. Zhang, K. Li, K. Li, B. Zhong, Y. Fu, Residual non-local attention networks for image restoration, in: International Conference on Learning Representations, 2019.

[31] S. Guo, Z. Yan, K. Zhang, W. Zuo, L. Zhang, Toward convolutional blind denoising of real photographs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1712–1722.

[32] K. Yu, C. Dong, C. Loy, X. Tang, Deep convolution networks for compression artifacts reduction, 2016, arXiv preprint arXiv:1608.02778.

[33] X. Fu, Z.J. Zha, F. Wu, X. Ding, J. Paisley, JPEG artifacts reduction via deep convolutional sparse coding, in: IEEE International Conference on Computer Vision, 2019.

[34] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.

[35] J. Cai, H. Zeng, H. Yong, Z. Cao, L. Zhang, Toward real-world single image super-resolution: A new benchmark and a new model, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3086–3095.

[36] C. Buciluǎ, R. Caruana, A. Niculescu-Mizil, Model compression, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 535–541.

[37] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, Comput. Sci. 14 (7) (2015) 38–39.

[38] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1116–1124.

[39] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, Int. Conf. Learn. Represent. (2017).

[40] Shiming, Ge, Shengwei, Zhao, Chenyu, Li, Jia, Li, Low-resolution face recognition in the wild via selective knowledge distillation, IEEE Trans. Image Process. 28 (4) (2019) 2051–2062.

[41] Q. Gao, Y. Zhao, G. Li, T. Tong, Image super-resolution using knowledge distillation, in: Asian Conference on Computer Vision, Springer, 2018, pp. 527–541.
[42] W. Lee, J. Lee, D. Kim, B. Ham, Learning with privileged information for efficient image super-resolution, in: European Conference on Computer Vision, Springer, 2020, pp. 465–482.
[43] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Springer, 2016, pp. 391–407.
[44] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.
[45] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2016, pp. 4898–4906.
[46] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[47] S. Woo, J. Park, J. Lee, S. K., Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.
[48] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[49] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2, 2001, pp. 416–423.
[50] J. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.
[51] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: NeurIPS Workshops, 2017.
[52] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space, in: IEEE International Conference on Image Processing, IEEE, 2007, pp. 313–316.
[53] T. Plotz, S. Roth, Benchmarking denoising algorithms with real photographs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2750–2759.
[54] Y. Chen, W. Yu, T. Pock, On learning optimized reaction diffusion processes for effective image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5261–5269.
[55] H.C. Burger, C.J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with BM3D? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2392–2399.
[56] E. Agustsson, R. Timofte, NTIRE 2017 challenge on single image super-resolution: Dataset and study, in: The IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.
[57] H.R. Sheikh, M.F. Sabir, A.C. Bovik, A statistical evaluation of recent full reference image quality assessment algorithms, IEEE Trans. Image Process. 15 (11) (2006) 3440–3451.
[58] Y. Shi, H. Zhong, Z. Yang, X. Yang, L. Lin, DDet: Dual-path dynamic enhancement network for real-world image super-resolution, IEEE Signal Process. Lett. PP (2020) 1.
[59] P. Wei, Z. Xie, H. Lu, Z. Zhan, Q. Ye, W. Zuo, L. Lin, Component divide-and-conquer for real-world image super-resolution, in: Proceedings of the European Conference on Computer Vision, 2020, pp. 101–117.