
Uformer: A General U-Shaped Transformer for Image Restoration

Zhendong Wang¹, Xiaodong Cun^{2*}, Jianmin Bao, Jianzhuang Liu³

¹ University of Science and Technology of China, ² University of Macau,

³ University of Chinese Academy of Sciences

Abstract

In this paper, we present Uformer, an effective and efficient Transformer-based architecture, in which we build a hierarchical encoder-decoder network using the Transformer block for image restoration. Uformer has two core designs to make it suitable for this task. The first key element is a local-enhanced window Transformer block, where we use non-overlapping window-based self-attention to reduce the computational requirement and employ the depth-wise convolution in the feed-forward network to further improve its potential for capturing local context. The second key element is that we explore three skip-connection schemes to effectively deliver information from the encoder to the decoder. Powered by these two designs, Uformer enjoys a high capability for capturing useful dependencies for image restoration. Extensive experiments on several image restoration tasks demonstrate the superiority of Uformer, including image denoising, deraining, deblurring and demoiréing. We expect that our work will encourage further research to explore Transformer-based architectures for low-level vision tasks. The code and models will be available at <https://github.com/ZhendongWang6/Uformer>.

1 Introduction

Since the rapid development of consumer and industry cameras and smartphones, the requirements of removing undesired degradation (e.g., noise, blur, rain, and moire pattern) in images are constantly growing. Recovering clear images from their degraded versions, i.e., image restoration, is a classic problem in computer vision. Recent state-of-the-art methods [1, 2, 3, 4] are mostly CNN-based, which achieve impressive results but show a limitation in capturing long-range dependencies. To address this problem, several recent works [5, 6, 7] start to employ single or few self-attention layers in low resolution feature maps due to the self-attention computational complexity being quadratic to the feature map size.

In this paper, we aim to leverage the capability of self-attention in feature maps at multi-scale resolutions to recover more image details. To achieve this, we present Uformer, an effective and efficient Transformer-based structure for image restoration. Uformer is built upon an elegant architecture, the so-called "UNet" [8]. We modify the convolution layers to Transformer blocks while keeping the same overall hierarchical encoder-decoder structure and the skip-connections.

Uformer has two core designs to make it suitable for image restoration. The first key element is a local-enhanced window Transformer block. To reduce the large computational complexity of self-attention on high resolution feature maps, we use non-overlapping window-based self-attention instead of global self-attention for capturing long-range dependencies. Since we build hierarchical feature maps and keep the window size unchanged, the window-based self-attention at low resolution is able to capture more global dependencies. On the other hand, previous works [9, 10] suggest that

*Corresponding author

self-attention has limitation to obtain local dependencies. To overcome this problem, inspired by the recent vision Transformers [10, 11], we leverage a depth-wise convolutional layer between two fully-connected layers of the feed-forward network in the Transformer block for better capturing local context.

The second key element is that we explore how to achieve better information delivering in the Transformer-based encoder-decoder structure. First, similar to UNet [8], we concatenate the features from the l -th stage of the encoder and the $(l-1)$ -th stage of the decoder firstly, and use the concatenated features as the input to the Transformer block in the decoder. Besides, we formulate the problem of delivering information from the encoder to the decoder as a process of self-attention computing: the features in the decoder play the role of *queries* and seek to estimate their relationship to the features in the encoder which play the role of *keys* and *values*. To achieve this, we design another two schemes. In the first one, we add a self-attention module into the Transformer block in the decoder, and use the features from the encoder as the keys and values, and the features in the decoder as the queries. In the second scheme, we combine the keys and values in the encoder and decoder together, and only use the queries from the decoder to find related keys. These three connection schemes can achieve competitive results under constrained computational complexity.

We show the effectiveness and efficiency of Uformer on various image restoration tasks. For denoising, Uformer outperforms the previous state-of-the-art (NBNet [3]) by 0.02 dB and 0.07 dB on the SIDD [12] and DND [13] benchmarks, respectively, while reducing **half** of the computational complexity. On the SPAD dataset [14] for rain removal, it obtains 46.00 dB on PSNR, an improvement of **4.53 dB** over the previous state-of-the-art (RCDNet [15]). Uformer also shows the potential in dual-pixel defocus blur removal [16] and moire pattern removal [17], which outperforms the UNet baseline by **0.52 dB** and **0.81 dB** respectively. We expect our work will encourage further research to explore Transformer-based architectures for image restoration.

Overall, we summarize the contributions of this paper as follows:

- We propose Uformer, an effective and efficient U-shaped Transformer for image restoration.
- We introduce a local-enhanced window Transformer block to make Uformer achieve better performance for image restoration. Three types of skip-connections, including two novel skip-connection schemes, are also presented in the Uformer architecture.
- Extensive experiments for image restoration tasks on various datasets demonstrate the superiority of the proposed Uformer.

2 Related Work

Image restoration architectures. Image restoration aims to restore the clean image from its degraded version. Learning effective models using the U-shaped structures with skip-connection to capture multi-scale information hierarchically is a popular choice for various image restoration tasks, including image denoising [4, 1, 3], deblurring [18, 19, 16], and demoireing [17, 20]. Image restoration methods are also designed by borrowing the key insight from the rapid development of image classification [21, 22]. For example, ResNet-based structure has been widely used for general image restoration [23, 5] as well as for specific tasks in image restoration such as super-resolution [24, 25] and image denoising [26, 27]. Besides, exploring the attention mechanism can also boost the performance. For example, squeeze-and-excitation networks [28] and non-local neural networks [29] inspire a branch of methods for different image restoration tasks, such as super-resolution [30], deraining [31, 1], and denoising [2, 1]. More CNN-based image restoration architectures can be found in the recent surveys [32, 33, 34] and the NTIRE Challenges [35].

Vision Transformers. Transformer [36] shows a significant performance in natural language processing (NLP). Different from the design of CNNs, Transformer-based network structures are naturally good at capturing long-range dependencies in the data by the global self-attention. The success of Transformer in the NLP domain also inspires the computer vision researchers. The pioneering work of ViT [37] directly trains the network using the Transformer-based architecture on the medium-size (16×16) flattened patches. With large-scale data pre-training (i.e., JFT-300M), ViT gets excellent results compared to state-of-the-art CNNs on image classification. Since then, several methods [38, 39] have been proposed for image classification using Transformer. For example, to train Transformer from scratch on ImageNet, T2T [40] progressively structurizes the image to tokens

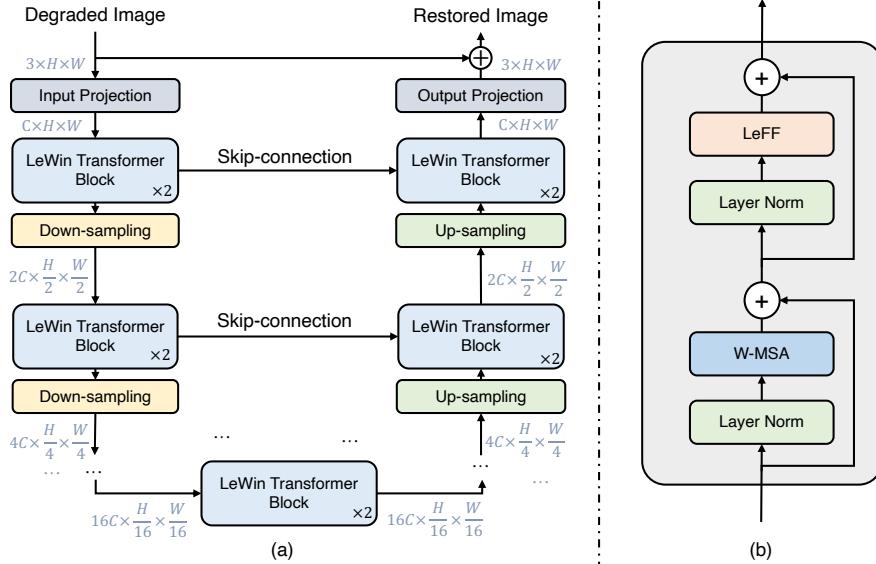


Figure 1: (a) Overview of the Uformer structure. (b) Structure of the LeWin Transformer block.

by recursively aggregating neighboring tokens into one token. Inspired by T2T, several works also try to involve neighborhood information (locality) into Transformer by adding convolutional layers into the multi-head self-attention (CVT [9]) and the feed-forward network (CeIT [10], LocalViT [11]). Recently, [41] designs the Swin Transformer, which is a hierarchical Transformer architecture with the shifted window-based multi-head attentions.

Besides image classification, Transformer-based architectures also show the potential in dense prediction tasks, such as semantic segmentation (SETR [42]) and depth estimation (DPT [43]). Their encoders are based on the Transformer pre-trained on ImageNet and their whole structures still majorly rely on the convolutional layers as the decoders.

For image restoration, IPT [44] jointly trains standard Transformer blocks with multi-tails and multi-heads on multiple low-level vision tasks. However, their framework still needs pre-training on a large-scale synthesized dataset and requires multi-task learning for good performance. Differently, we design a general U-shaped Transformer-based structure for image restoration under the same settings (training datasets and computational resources) as the CNN-based works.

3 Method

In this section, we first describe the overall pipeline and the hierarchical structure of Uformer for image restoration. Then, we provide the details of the *LeWin Transformer block* which is the basic component of Uformer. After that, we introduce three variants of skip-connection for bridging the information flow between the encoder and the decoder.

3.1 Overall Pipeline

As shown in Figure 1(a), the overall structure of the proposed Uformer is a U-shaped hierarchical network with skip-connections between the encoder and the decoder. To be specific, given a degraded image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$, Uformer firstly applies a 3×3 convolutional layer with LeakyReLU to extract low-level features $\mathbf{X}_0 \in \mathbb{R}^{C \times H \times W}$. Next, following the design of the U-shaped structures [8, 45], the feature maps \mathbf{X}_0 are passed through K encoder stages. Each stage contains a stack of the proposed LeWin Transformer blocks and one down-sampling layer. The LeWin Transformer block takes advantage of the self-attention mechanism for capturing long-range dependencies, and also cuts the computational cost due to the usage of self-attention through non-overlapping windows on the feature maps. In the down-sampling layer, we first reshape the flattened features into 2D spatial feature maps, and then down-sample the maps and double the channels using 4×4 convolution with stride 2. For

example, given the input feature maps $\mathbf{X}_0 \in \mathbb{R}^{C \times H \times W}$, the l -th stage of the encoder produces the feature maps $\mathbf{X}_l \in \mathbb{R}^{2^l C \times \frac{H}{2^l} \times \frac{W}{2^l}}$.

Then, a bottleneck stage with a stack of LeWin Transformer blocks is added at the end of the encoder. In this stage, thanks to the hierarchical structure, the Transformer blocks capture longer (even global when the window size equals the feature map size) dependencies.

For feature reconstruction, the proposed decoder also contains K stages. Each consists of an up-sampling layer and a stack of LeWin Transformer blocks similar to the encoder. We use 2×2 transposed convolution with stride 2 for the up-sampling. This layer reduces half of the feature channels and doubles the size of the feature maps. After that, the features inputted to the LeWin Transformer blocks are the up-sampled features and the corresponding features from the encoder through skip-connection. Next, the LeWin Transformer blocks are utilized to learn to restore the image. After the K decoder stages, we reshape the flattened features to 2D feature maps and apply a 3×3 convolution layer to obtain a residual image $\mathbf{R} \in \mathbb{R}^{3 \times H \times W}$. Finally, the restored image is obtained by $\mathbf{I}' = \mathbf{I} + \mathbf{R}$. In our experiments, we empirically set $K = 4$ and each stage contains two LeWin Transformer blocks. We train Uformer using the Charbonnier loss [46, 2]:

$$\ell(\mathbf{I}', \hat{\mathbf{I}}) = \sqrt{\|\mathbf{I}' - \hat{\mathbf{I}}\|^2 + \epsilon^2}, \quad (1)$$

where $\hat{\mathbf{I}}$ is the ground-truth image, and $\epsilon = 10^{-3}$ is a constant in all the experiments.

3.2 LeWin Transformer Block

There are two main issues to apply Transformer for image restoration. First, the standard Transformer architecture [36, 37] computes self-attention globally between all tokens, which contributes to the quadratic computation cost with respect to the number of tokens. It is unsuitable to apply global self-attention on the high-resolution feature maps. Second, the local context information is essential for image restoration tasks since the neighborhood of a degraded pixel can be leveraged to restore its clean version, but previous works [11, 9] suggest that Transformer shows a limitation in capturing local dependencies.

To address the above mentioned two issues, we propose a locally-enhanced window (LeWin) Transformer block, as shown in Figure 1(b), which benefits from the self-attention in Transformer to capture long-range dependencies, and also involves the convolution operator into Transformer to capture useful local context. Specifically, given the features at the $(l-1)$ -th block \mathbf{X}_{l-1} , we build the block with two core designs: (1) non-overlapping Window-based Multi-head Self-Attention (W-MSA) and (2) Locally-enhanced Feed-Forward Network (LeFF). The computation of a LeWin Transformer block is represented as:

$$\begin{aligned} \mathbf{X}'_l &= \text{W-MSA}(\text{LN}(\mathbf{X}_{l-1})) + \mathbf{X}_{l-1}, \\ \mathbf{X}_l &= \text{LeFF}(\text{LN}(\mathbf{X}'_l)) + \mathbf{X}'_l, \end{aligned} \quad (2)$$

where \mathbf{X}'_l and \mathbf{X}_l are the outputs of the W-MSA module and LeFF module respectively. LN represents the layer normalization [47]. Next, we introduce the details of W-MSA and LeFF.

Window-based Multi-head Self-Attention (W-MSA). Instead of using global self-attention like the vanilla Transformer, we perform the self-attention within non-overlapping local windows, which reduces the computational cost significantly. Given the 2D feature maps $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ with H and W being the height and width of the maps, we split \mathbf{X} into non-overlapping windows with the window size of $M \times M$, and then get the flattened and transposed features $\mathbf{X}^i \in \mathbb{R}^{M^2 \times C}$ from each window i . Next, we perform self-attention on the flattened features in each window. Suppose the head number is k and the head dimension is $d_k = C/k$. Then computing the k -th head self-attention in the non-overlapping windows can be defined as:

$$\begin{aligned} \mathbf{X} &= \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N\}, \quad N = HW/M^2, \\ \mathbf{Y}_k^i &= \text{Attention}(\mathbf{X}^i \mathbf{W}_k^Q, \mathbf{X}^i \mathbf{W}_k^K, \mathbf{X}^i \mathbf{W}_k^V), \quad i = 1, \dots, N, \\ \hat{\mathbf{X}}_k &= \{\mathbf{Y}_k^1, \mathbf{Y}_k^2, \dots, \mathbf{Y}_k^M\}, \end{aligned} \quad (3)$$

where $\mathbf{W}_k^Q, \mathbf{W}_k^K, \mathbf{W}_k^V \in \mathbb{R}^{C \times d_k}$ represent the projection matrices of the queries, keys, and values for the k -th head, respectively. $\hat{\mathbf{X}}_k$ is the output of the k -th head. Then the outputs for all heads

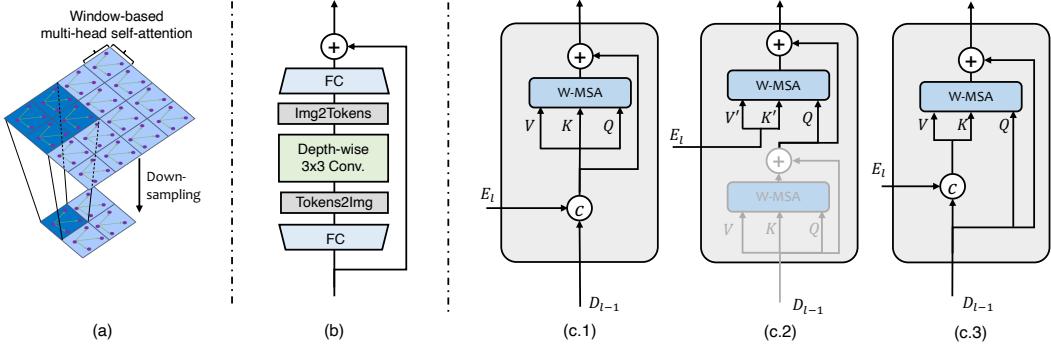


Figure 2: (a) The non-overlapping windows of two neighboring stages where the receptive fields at the low resolution are larger. (b) Structure of LeFF. (c) Three skip-connection schemes: (c.1) Concat-Skip, (c.2) Cross-Skip and (c.3) ConcatCross-Skip.

$\{1, 2, \dots, k\}$ are concatenated and then linearly projected to get the final result. Inspired by previous works [48, 41], we also apply the relative position encoding into the attention module, so the attention calculation can be formulated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{B}\right)\mathbf{V}, \quad (4)$$

where \mathbf{B} is the relative position bias, whose values are taken from $\hat{\mathbf{B}} \in \mathbb{R}^{(2M-1) \times (2M-1)}$ with learnable parameters [48, 41].

Window-based self-attention can significantly reduce the computational cost compared with global self-attention. Given the feature maps $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, the computational complexity drops from $O(H^2W^2C)$ to $O(\frac{HW}{M^2}M^4C) = O(M^2HWC)$. As shown in Figure 2(a), since we design Uformer as a hierarchical architecture, our window-based self-attention at low resolution feature maps works on larger receptive fields and is sufficient to learn long-range dependencies. We also try the shifted-window strategy [41] in the even LeWin Transformer block of each stage in our framework, which gives only slightly better results.

Locally-enhanced Feed-Forward Network (LeFF). As pointed out by previous works [9, 10], the Feed-Forward Network (FFN) in the standard Transformer presents limited capability to leverage local context. However, neighboring pixels are crucial references for image restoration [49, 50]. To overcome this issue, we add a depth-wise convolutional block to the FFN in our Transformer-based structure following the recent works [51, 10, 11]. As shown in Figure 2(b), we first apply a linear projection layer to each token to increase its feature dimension. Next, we reshape the tokens to 2D feature maps, and use a 3×3 depth-wise convolution to capture local information. Then we flatten the features to tokens and shrink the channels via another linear layer to match the dimension of the input channels. We use GELU [52] as the activation function after each linear/convolution layer.

3.3 Variants of Skip-Connection

To investigate how to deliver the learned low-level features from the encoder to the decoder, considering the self-attention computing in Transformer, we present three different skip-connection schemes, including concatenation-based skip-connection, cross-attention as skip-connection, and concatenation-based cross-attention as skip-connection.

Concatenation-based Skip-connection (Concat-Skip). Concatenation-based skip-connection is based on the widely-used skip-connection in UNet [8, 4, 3]. To build our network, firstly, we concatenate the l -th stage flattened features \mathbf{E}_l and each encoder stage with the features \mathbf{D}_{l-1} from the $(l-1)$ -th decoder stage channel-wisely. Then, we feed the concatenated features to the W-MSA component of the first LeWin Transformer block in the decoder stage, as shown in Figure 2(c.1).

Cross-attention as Skip-connection (Cross-Skip) Instead of directly concatenating features from the encoder and the decoder, we design Cross-Skip inspired by the decoder structure in the language Transformer [36]. As shown in Figure 2(c.2), we first add an additional attention module into the first

	GMACs	# Param	PSNR		GMACs	# Param	PSNR		Uformer ₃₂	PSNR
UNet ₃₂	15.53G	9.5M	39.62	ViT ₂₅₆	8.83G	14.86M	38.51	w/o window shift		39.76
UNet ₄₈	34.82G	21.38M	39.65	Uformer ₁₆	10.05G	5.23M	39.66	w window shift		39.77
Uformer ₁₆	10.05G	5.23M	39.66							
Uformer ₃₂	40.86G	20.63M	39.77							

(a) Transformer vs. convolution.	(b) Hierarchical structure vs. single scale.	(c) Effect of window shift.

	GMACs	# Param	PSNR		W-MSA	FFN	GMACs	# Param	PSNR
Uformer ₃₂ -Concat	40.86G	20.63M	39.77		-	-	39.91G	20.47M	39.74
Uformer ₄₄ -Cross	40.64G	27.95M	39.75	Uformer ₃₂	DConv.	-	41.72G	20.75M	39.72
Uformer ₄₄ -ConcatCross	38.61G	27.28M	39.73		-	DConv.	40.86G	20.63M	39.77

(a) Transformer vs. convolution.
(b) Hierarchical structure vs.
single scale.

(c) Effect of window shift.

(d) Different skip-connections.

(e) Enhancing locality in different modules.

Table 1: Ablation studies on SIDD [12] for image denoising. The results on PSNR are reported.

LeWin Transformer block in each decoder stage. The first self-attention module in this block (the shaded one) is used to seek the self-similarity pixel-wisely from the decoder features \mathbf{D}_{l-1} , and the second attention module in this block takes the features \mathbf{E}_l from the encoder as the *keys* and *values*, and uses the features from the first module as the *queries*.

Concatenation-based Cross-attention as Skip-connection (ConcatCross-Skip). Combining above two variants, we also design another skip-connection. As illustrated in Figure 2(c.3), we concatenate the features \mathbf{E}_l from the encoder and \mathbf{D}_{l-1} from the decoder as the *keys* and *values*, while the *queries* are only from the decoder.

4 Experiments

In this section, we first perform comprehensive ablation studies to validate each component of our proposed Uformer. Then we verify the effectiveness and efficiency of Uformer on various image restoration tasks (denoising, deraining, deblurring, and demoireing).

4.1 Experimental Setup

Settings. Following the common training strategy of Transformer [36], we train our framework using the AdamW optimizer [53] with the momentum terms of (0.9, 0.999) and the weight decay of 0.02. We randomly augment the training samples using the horizontal flipping and rotate the images by 90°, 180°, or 270°. We use the cosine decay strategy to decrease the learning rate to 1e-6 with the initial learning rate 2e-4. We set the window size to 8×8 in all LeWin Transformer blocks. More dataset-specific experimental settings can be found in the supplementary materials.

Evaluation metrics. We apply the commonly-used PSNR and SSIM [54] metrics to evaluate the performance. These metrics are calculated in the RGB color space except for deraining where we evaluate the PSNR and SSIM on the Y channel in the YCbCr color space, following the previous method [15].

4.2 Ablation Studies

We analyze each component of Uformer on the SIDD dataset [12] for image denoising. Following the common training strategy in existing works [3, 2], we train Uformer for 250 epochs with batch size 32 and input image size 128 × 128 on 2 NVIDIA V100 GPUs. Table 1 presents the ablation results, discussed as follows:

Transformer vs. convolution. We replace all the LeWin Transformer blocks in Uformer with the convolution-based ResBlocks [3], resulting in the so-called "UNet", while keeping all others unchanged. We adopt the feature concatenation (Concat-Skip) as the default skip-connection method. Table 1(a) reports the results. We observe that Uformer₁₆ achieves 39.66 dB and outperforms UNet₃₂ by 0.04 dB with fewer parameters and less computation. Uformer₃₂ achieves 39.77 dB and outperforms UNet₄₈ by 0.12 dB with fewer parameters and a slightly higher computation cost. Here, the subscript number devotes the channel dimension C of the output from the first convolution layer

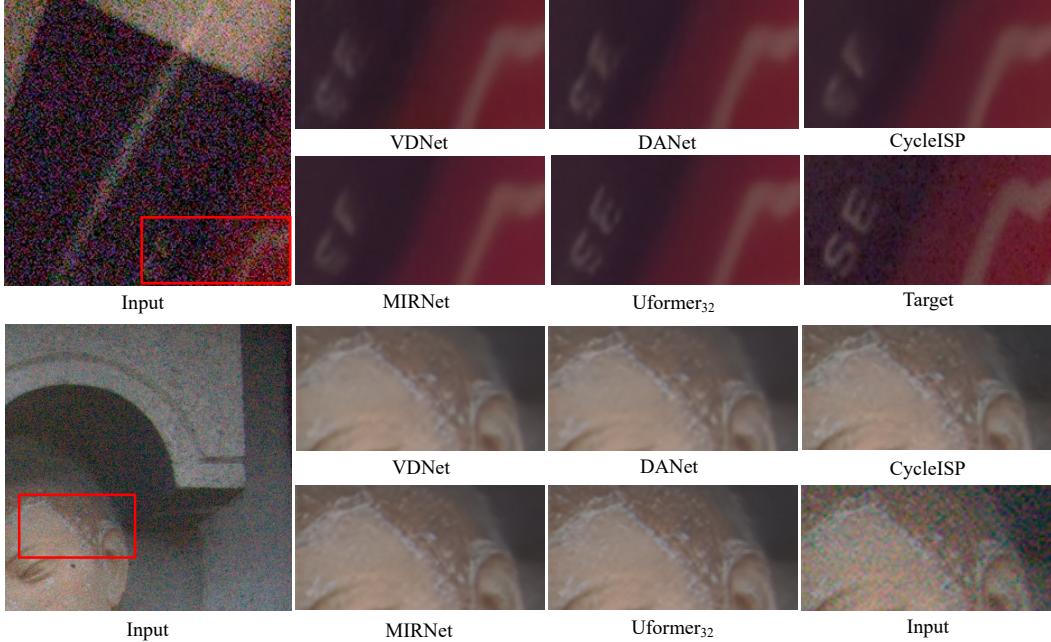


Figure 3: Comparison with state-of-the-art methods on real noise removal. The top sample comes from SIDD while the bottom one is from DND.

in each model (see Figure 1(a)). This study indicates the effectiveness of the LeWin Transformer block.

Hierarchical structure vs. single scale. We further build a ViT-based architecture which only contains a single scale of the feature maps for image denoising. This architecture employs a head of two convolution layers for extracting features from the input image and also a tail of two convolution layers for the output. 12 standard Transformer blocks are used between the head and the tail. We train the ViT with the hidden dimension of 256 on patch size 16×16 . The results are presented in Table 1(b). We observe that our Uformer significantly outperforms the ViT-based architecture, which demonstrates the importance of hierarchical feature maps for image restoration.

Which skip-connection is better? Table 1(d) compares the results of using different skip-connections in our Uformer: concatenating features (*Concat*), cross-attention (*Cross*), and concatenating *keys* and *values* for cross-attention (*ConcatCross*). These three skip-connections achieve similar results, and concatenating features gets slightly better performance. We adopt the feature concatenation as the default setting in Uformer.

Where to enhance locality? Table 1(e) compares the results of no locality enhancement and enhancing locality in the self-attention calculation [9] or the feed-forward network. We observe that introducing locality into the feed-forward network yields +0.03 dB over the baseline (no locality), while introducing locality into the self-attention yields -0.02 dB, indicating that introducing locality into the feed-forward network is more suitable for image denoising.

Is shifted window important? Table 1(c) reports the results of whether to use the shifted window design [41] in Uformer. We observe that window shift brings an improvement of 0.01 dB for image denoising. So we use the window shift as the default setting in our experiments.

4.3 Real Noise Removal

Table 2 reports the results of real noise removal on the SIDD [12] and DND [13] datasets. We compare Uformer with 8 state-of-the-art denoising methods, including the feature-based BM3D [55] and 7 learning-based methods: RIDNet [56], VDN [57], CycleISP [58], NBNNet [3], DANet [4], MIRNet [2], and MPRNet [1]. Our Uformer₃₂ achieves 39.77 dB on PSNR, surpassing all the other methods by at least 0.02 dB. As for the DND dataset, we follow the common evaluation strategy and test our model

Method	SIDD		DND	
	PSNR	SSIM	PSNR	SSIM
BM3D [55]	25.65	0.685	34.51	0.851
RIDNet [56]	38.71	0.914	39.26	0.953
VDN [57]	39.28	0.909	39.38	0.952
DANet [4]	39.47	0.918	39.59	0.955
CycleISP [58]	39.52	0.957	39.56	0.956
MIRNet [2]	39.72	0.959	39.88	0.956
MPRNet [1]	39.71	0.958	39.80	0.954
NBNet [3]	39.75	0.973	39.89	0.955
Uformer₃₂	39.77	0.970	39.96	0.956

Table 2: Comparison on SIDD and DND for real noise removal.

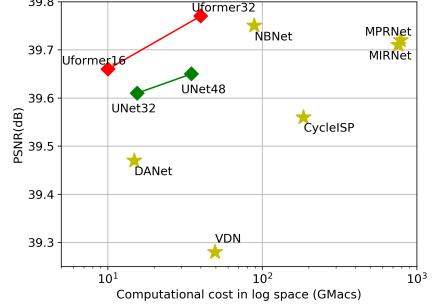


Figure 4: PSNR vs. computational cost on the SIDD dataset.

trained on SIDD via the online server testing. Uformer surpasses the previous state-of-the-art method NBNet [3] by 0.07 dB. To verify whether the gains benefit from more computation cost, we present the results of PSNR vs. computation in Figure 4. We notice that our Uformer₁₆ can achieve a better performance than most models but with the least computation cost. This demonstrates the efficiency and effectiveness of Uformer. We also show the qualitative results on SIDD and DND in Figure 3, in which Uformer can not only successfully remove the noise but also keep the texture details.

4.4 Real Rain Removal

We conduct the deraining experiments on SPAD [14] and compare with 7 deraining methods: GMM [59], DDN [60], RESCAN [31], PReNet [61], JORDER-E [62], SPANet [14], and RCDNet [15]. As shown in Table 3, Uformer presents a significantly better performance, achieving almost 5 dB improvement over existing works. This indicates the strong capability of Uformer for deraining on this real rain dataset. We also provide the qualitative results in Figure 5 where Uformer can remove the rain successfully while introducing fewer artifacts.

	input	GMM [59]	DDN [60]	RESCAN [31]	PReNet [61]	SPANet [14]	JORDER-E [62]	RCDNet [15]	Uformer₁₆
PSNR↑	34.15	34.30	36.16	38.11	40.16	40.24	40.78	41.47	46.00
SSIM↑	0.9269	0.9428	0.9463	0.9707	0.9816	0.9811	0.9811	0.9834	0.9912

Table 3: Comparison on the SPAD dataset [14] for real rain removal.



Figure 5: Deraining results on the SPAD dataset [14]. Our Uformer shows a significant better performance than other methods.

	Defocus Blur Removal				Moire Pattern Removal			
	DMENet [64]	JNB [63]	DPDNet [16]	Uformer₁₆	UNet [8]	CFNet [20]	MS-Net [17]	Uformer₁₆
PSNR↑	23.41	23.84	25.13	25.65	26.49	26.09	26.99	27.30
SSIM↑	0.714	0.715	0.786	0.795	0.864	0.863	0.871	0.865

Table 4: Defocus blur removal on DPD [16] and moire pattern removal on TIP18 [17].



Figure 6: Visual comparisons for defocus blur removal (first row) and moire pattern removal (second row). Uformer shows a strong ability to reconstruct the clean images from the blur and moire images.

4.5 Defocus Blur Removal

We perform defocus blur removal on the dual-pixel dataset DPD [16]. Table 4 and Figure 6 report the quantitative and qualitative results respectively. Uformer achieves a better performance (+2.24 dB, +1.81 dB, and +1.55 dB) over previous state-of-the-art methods DPDNet [16], JNB [63], and DMENet [64], respectively. From the visualization results, we observe that Uformer reduces more out-of-focus blur.

4.6 Moire Pattern Removal

We also conduct an experiment of moire pattern removal on the TIP18 dataset [17]. As shown in Table 4, Uformer outperforms previous methods CFNet [20] and MS-Net [17] by 1.21 dB and 0.31 dB respectively. It is worth noting that these two methods are carefully designed for demoireing. This experiment further demonstrates the superiority of Uformer. The visual results are presented in Figure 6. Uformer achieves better visual results with fewer artifacts.

5 Conclusions

In this paper, we have presented an alternative architecture for image restoration tasks by introducing the Transformer block. In contrast to existing CNN-based structures, our Uformer builds upon the main component LeWin Transformer block, which can not only handle local context but also capture long-range dependencies efficiently. To explore how to achieve better information delivering in the encoder-decoder structure, we further study three different skip-connection schemes in Uformer, which achieve competitive results. Extensive experiments demonstrate that Uformer achieves the state-of-the art performance on several tasks, including denoising, deraining, deblurring, and demoireing, surpassing the UNet family by a large margin with less computation cost and fewer model parameters.

Limitation and Broader Impacts. Thanks to the proposed structure, Uformer achieves the state-of-the-art performance on a variety of image restoration tasks (image denoising, deraining, deblurring, and demoireing). We have not tested Uformer for more vision tasks such as image-to-image translation, image super-resolution, and so on. We look forward to investigating Uformer for more applications. Meanwhile we notice that there are several negative impacts caused by abusing image restoration techniques. For example, it may cause human privacy issue with the restored images in surveillance. Meanwhile, the techniques may destroy the original patterns for camera identification and multi-media copyright [65], which hurts the authenticity for image forensics.

References

- [1] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.
- [2] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020.
- [3] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. Nbnnet: Noise basis learning for image denoising with subspace projection. In *CVPR*, 2021.
- [4] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020.
- [5] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019.
- [6] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *ACMMM*, 2018.
- [7] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018.
- [8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 2015.
- [9] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.
- [10] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.
- [11] Yawei Li, Kai Zhang, Jiezheng Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.
- [12] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.
- [13] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.
- [14] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, 2019.
- [15] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020.
- [16] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*. Springer, 2020.
- [17] Yujing Sun, Yizhou Yu, and Wenping Wang. Moiré photo restoration using multiresolution convolutional neural networks. *TIP*, 27(8):4160–4172, 2018.
- [18] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. *ArXiv e-prints*, 2017.
- [19] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019.
- [20] Bolin Liu, Xiao Shu, and Xiaolin Wu. Demoiréing of camera-captured screen images using deep convolutional neural network. *arXiv preprint arXiv:1804.03809*, 2018.
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [23] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *CVPR*, 2019.
- [24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshop*, 2017.
- [25] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018.
- [26] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 26(7):3142–3155, 2017.
- [27] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *ICCV*, 2019.
- [28] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

- [29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [30] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.
- [31] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018.
- [32] Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan. Deep learning for image denoising: a survey. In *International Conference on Genetic and Evolutionary Computing*, 2018.
- [33] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *TPAMI*, 2020.
- [34] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, 2019.
- [35] Abdelrahman Abdelhamed, Radu Timofte, and Michael S. Brown. Ntire 2019 challenge on real image denoising: Methods and results. In *CVPR Workshop*, 2019.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.
- [38] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.
- [39] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020.
- [40] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.
- [41] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
- [42] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.
- [43] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021.
- [44] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.
- [45] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- [46] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994.
- [47] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [48] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.
- [49] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021.
- [50] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.
- [51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.
- [52] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.

- [55] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 16(8):2080–2095, 2007.
- [56] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019.
- [57] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019.
- [58] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *CVPR*, 2020.
- [59] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. Rain streak removal using layer priors. In *CVPR*, 2016.
- [60] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017.
- [61] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019.
- [62] Wenhao Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *TPAMI*, 42(6):1377–1393, 2020.
- [63] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015.
- [64] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019.
- [65] Davide Cozzolino and Luisa Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *arXiv preprint arXiv:1808.08396*, 2018.

A Appendix

A.1 Additional Experiment Settings for Each Dataset

SIDD [12] and DND [13] for denoising. The training samples are randomly cropped from the original images in SIDD [12] with size 128×128 , which is also the common training strategy for image denoising in recent works [4, 3, 2]. Then, the trained model is evaluated on the 256×256 patches of SIDD and 512×512 patches of DND test images, following [2].

SPA [14] for deraining. This dataset contains over 64k 256×256 images for training and 1k 512×512 images for evaluation. We train the proposed Uformer₁₆ on one GPU, with mini-batches of size 8 on the 256×256 samples. Since this dataset is large enough and the training process converges fast, we just train Uformer₁₆ for 5 epochs in the experiment. Finally, we evaluate the performance on the test images with the default settings.

DPD [16] for dual-pixel deblurring. Following the official patch segmentation algorithm [16] of DPD, we crop the training and validation samples to 60% overlapping 512×512 patches to train the model. We also discard 30% of the patches that have the lowest sharpness energy (by applying Sobel filter to the patches) as [16]. For evaluation, the trained model is tested on the full-size test images.

TIP18 dataset [17] for demoireing. Since images in this dataset contain additional borders, following [17], we crop the central regions with the ratio of $[0.15, 0.85]$ in all training/validation/testing splits and resize them to 256×256 for training and evaluation. For this task is sensitive to the down-sampling operation, we choose the bilinear interpolation as the previous work [17].

A.2 More Visual Comparisons

As shown in Figures 7-10, we give more visual results of our method and others on the four tasks (denoising, deraining, deblurring, and demoireing) as the supplement of the visualization in the main paper.

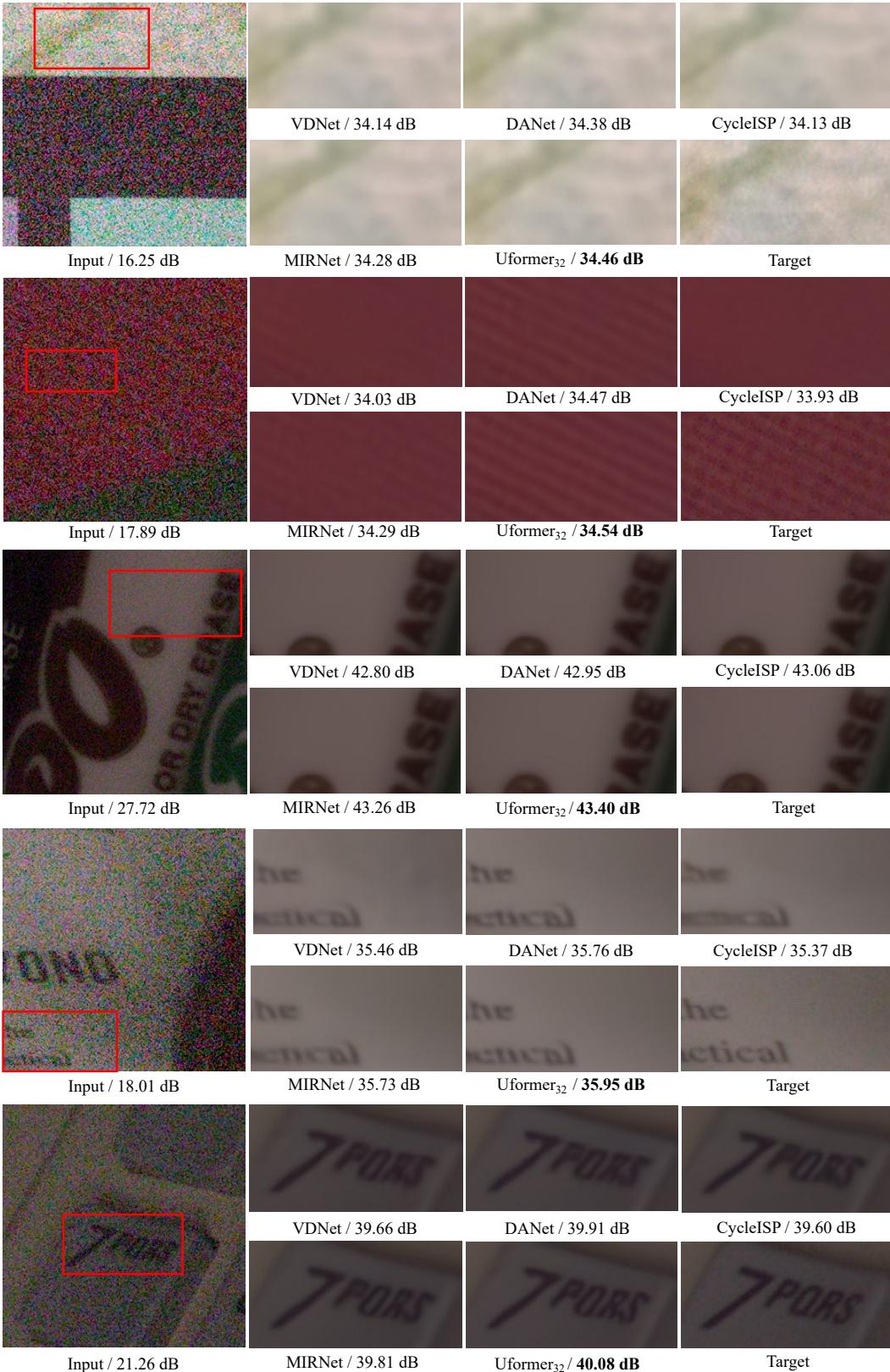


Figure 7: More visual results on the SIDD dataset [12] for image denoising. The PSNR value under each patch is computed on the corresponding whole image.

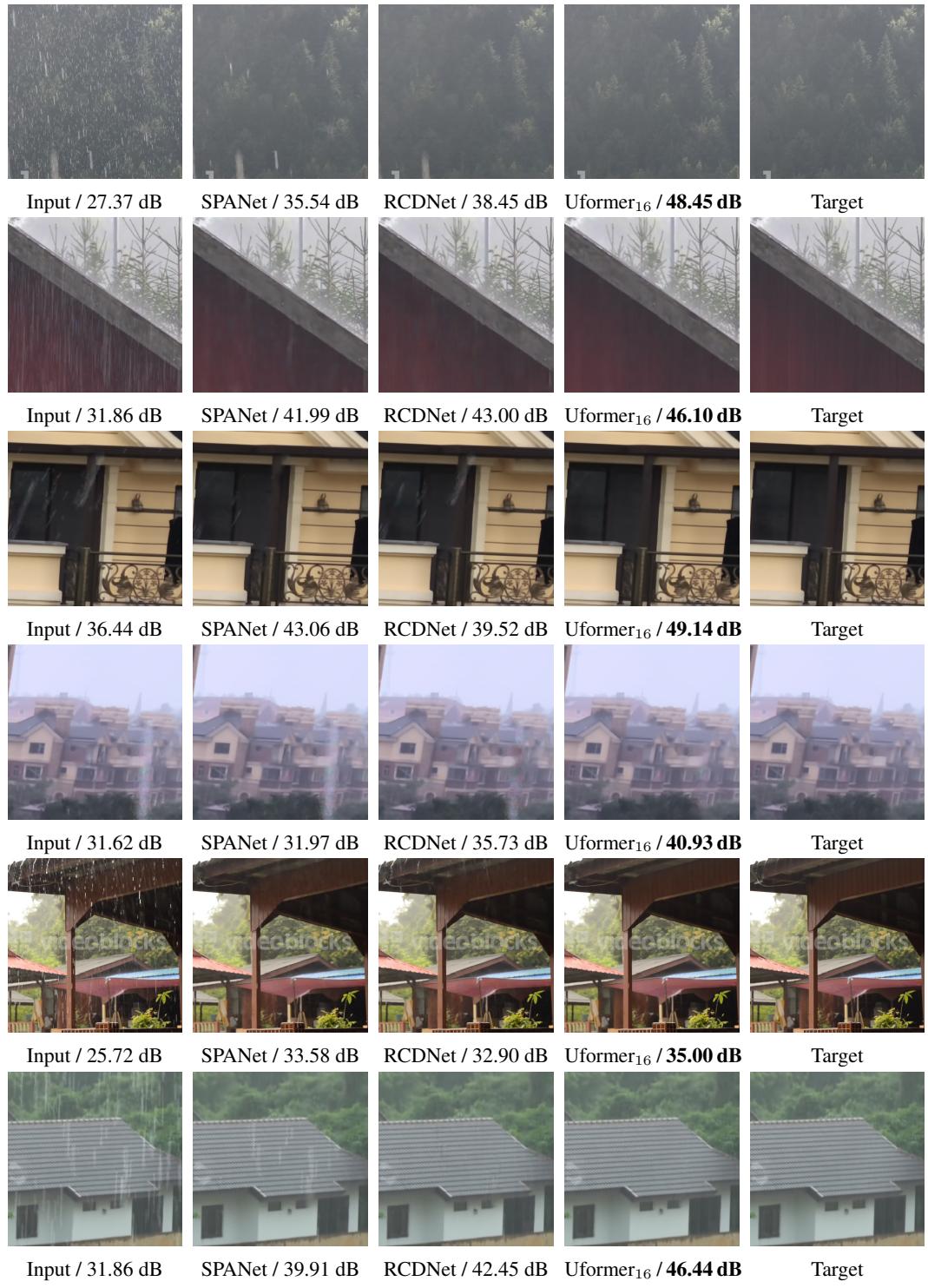


Figure 8: More results on SPA [14] for image deraining.

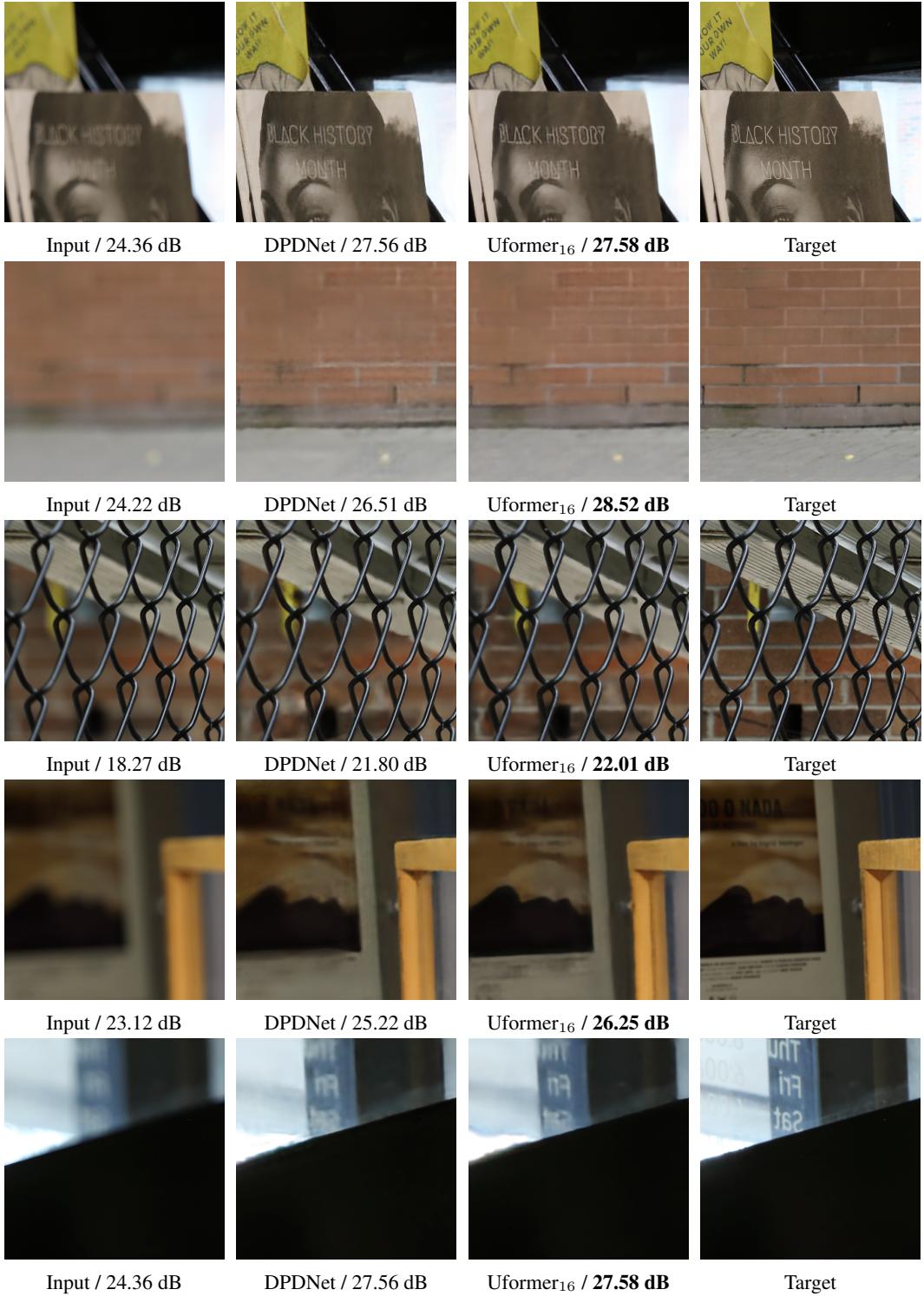


Figure 9: More results on DPD [16] for image deblurring. We report the performance of PSNR on the whole test image and show the zoomed region only for visual comparison.

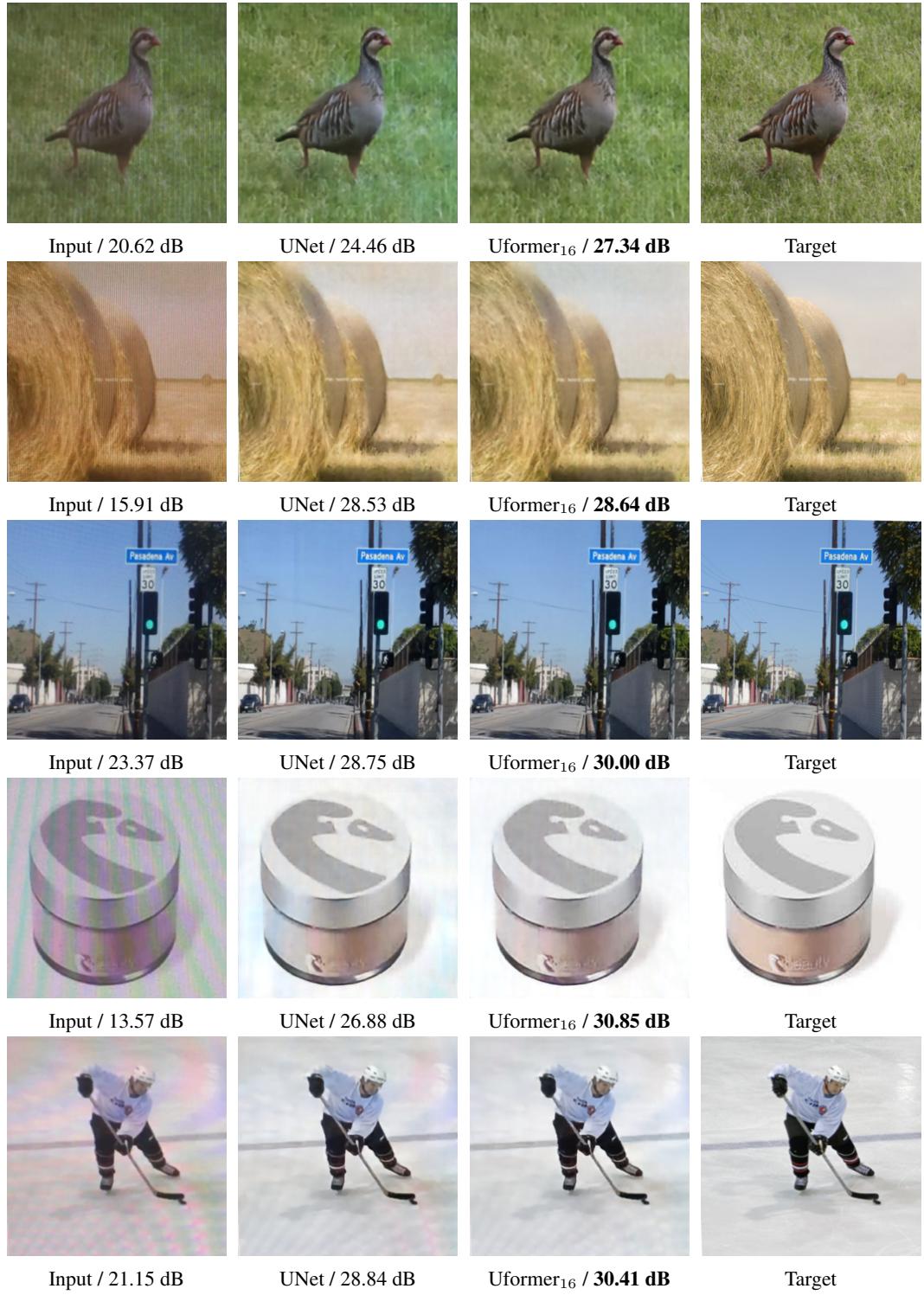


Figure 10: More results on TIP18 [14] for image demoiréing.