NTIRE 2023 Image Denoising ($\sigma = 50$) Challenge Factsheet DDT: Dual-branch Deformable Transformer for image denoising

Kangliang Liu
Fudan University
Shanghai, China
klliu21@m.fudan.edu.cn

1. Team detail

- Team ID: 18
- Team name: IMCgo
- Team leader name: Kangliang Liu
- Team leader address, phone number, and email: Fudan University, Shanghai, China.
  (+86)15651706080
  klliu21@m.fudan.edu.cn
- Rest of the team members: //
- Affiliation: Fudan University
- Affiliation of the team and/or team members with NTIRE 2023 sponsors (check the workshop website): //
- User names and entries on the NTIRE 2023 Codalab competitions (development/validation and testing phases) Merenguelkl
- Best scoring entries of the team during development/validation phase: ♯6 30.50dB
- Link to the codes/executables of the solution(s): GitHub link

2. Method detail

The main body of the method is from our another paper which has been already accepted by ICME 2023. Our main purpose is to develop an efficient Transformer model that can process high-resolution images the image denoising tasks. In order to enhance feature representations, we use a dual-branch structure to parallelly capture local and global features. In addition, we use a deformable strategy to focus attention on more important regions and further reduce computational costs. The overall pipeline of the Dual-branch Deformable Transformer (DDT) is described as Fig. 1.

2.1. Overall Pipeline

Given a noisy image $I_{in} \in \mathbb{R}^{H \times W \times 3}$, the input is first projected into the feature $F_0 \in \mathbb{R}^{H \times W \times C}$ by a 3×3 convolution layer. Then the feature F_0 goes through a 4-stage Unet-like encoder-decoder architecture. Each stage includes multiple DDTBs and a sample layer (*i.e.* downsample or upsample). The output feature $F_3 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 8C}$ goes through the Bottleneck and is fed into the decoder.The decoder has a symmetrical structure, where an upsample layer is used at the beginning of each stage. The skip-connection [5] is used to boost performance, and we use the concatenation operation and a 1×1 convolution layer for feature fusion. Inspired by [7], we introduce a refinement stage after the decoder, which aims to enhance feature representation for more details of images. Finally, the feature F_r from the refinement stage through a 3×3 convolution filter to obtain the residual image $I_r \in \mathbb{R}^{H \times W \times 3}$, and the restored output I_{out} is denoted as:

$$I_{out} = I_{in} + I_r \tag{1}$$

2.2. Dual-branch Deformable Transformer Block

DDTB consists of two main components: Dual-branch Deformable Attention (DDA) and Depth Feed-Forward Network (DFFN). A layer normalization is first applied in each part.

Specifically, DDA splits the feature over the channel dimension and sends them into a dual-branch structure, where local and global interactions are performed parallelly. In the local branch, we divide the feature into non-overlapping patches with pre-defined patch sizes and apply the spatial attention mechanism inside the patches. In the global brunch, we use a pre-defined number of patches to do the patch partitioning and perform calculations among the corresponding positions of each patch. The multi-scale features output by the dual-branch are then fused by concatenation operation and a linear layer. We utilize a deformable attention mechanism as the spatial operation in both branches, which will be introduced in Section 2.3.

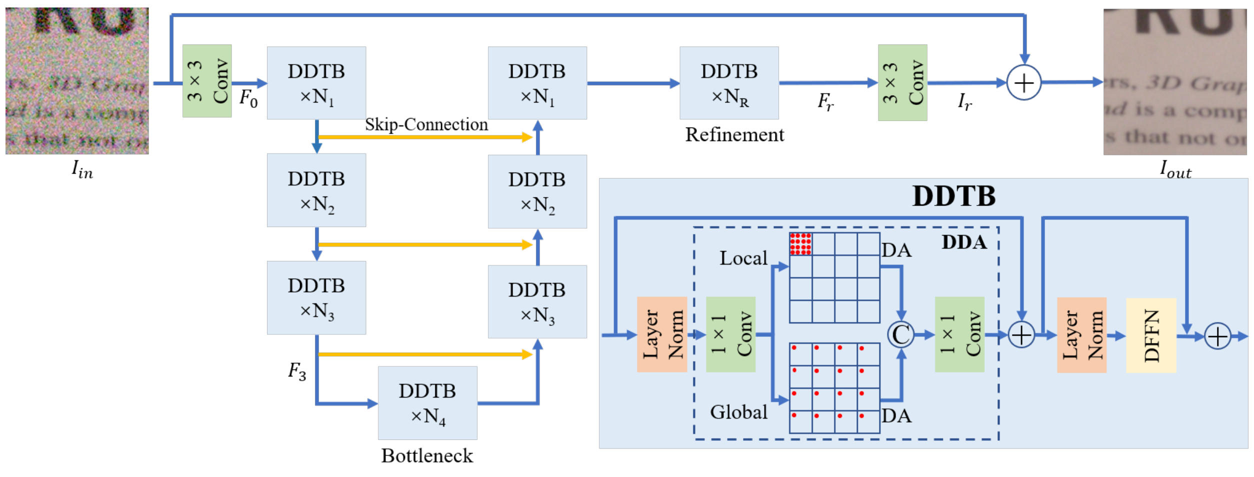

Figure 1. The overall architecture of the Dual-branch Deformable Transformer (DDT). The Unet-like hierarchical architecture with an extra refinement stage is used for image denoising, and each stage contains a different number of Dual-branch Deformable Transformer Blocks (DDTBs). The DDTB captures the local and global features parallelly using Dual-branch Deformable Attention (DDA) and Depth-wise Feed-Forward Network (DFFN).

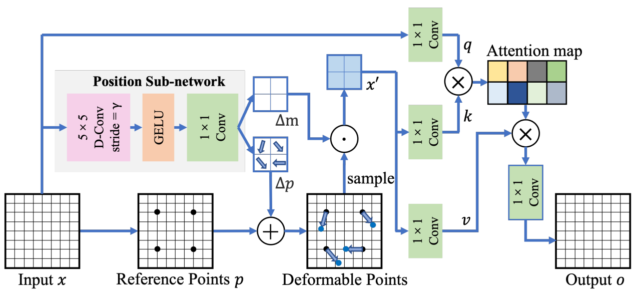

Figure 2. The structure of deformable attention. A group of offsets Δp are learned from the input x and added on the pre-defined reference points p to get deformable points. Features x' sampled from the input at the positions of deformable points are then projected as keys(k) and values(v), while queries(q) are still from the input x.

Inspired by [3, 7], we add depth-wise convolution into the standard Feed-Forward Network [1] as DFFN to further enhance features. Formally, The l-th DDTB is formulated as:

$$\hat{X}_l = \texttt{DDA}(\texttt{LN}(X_{l-1})) + X_{l-1} \tag{2}$$

$$X_l = \texttt{DFFN}(\texttt{LN}(\hat{X}_l)) + \hat{X}_l \tag{3}$$

where X_{l-1} represents the output of $(l-1)th$ DDTB, and LN means the layer normalization. The residual connection [2] is used in both components. With the dual-branch structure, we can process inputs with linear complexity, making it possible to process high-resolution images using Transformer architecture, preserving global receptive fields while capturing locality.

2.3. Deformable Attention

To further save computational costs, the deformable attention (Fig. 2) is applied for efficient spatial operation. Inspired by [6], we first define a set of grid-shape reference points $p \in \mathbb{R}^{H_G \times W_G \times 2}$ based on the input $x \in \mathbb{R}^{H \times W \times C}$. The number of points is determined by a hyper-parameter γ, $H_G = H/\gamma$ and $W_G = W/\gamma$. Meanwhile, we feed x into a Position Sub-network containing a 5×5 depth-wise convolutional layer with stride γ. The output of the sub-network has three channels, and the first two channels are deformable offsets $\Delta p \in \mathbb{R}^{H_G \times W_G \times 2}$ for each reference point. We add the offsets Δp on each point to obtain the deformable points and sample the feature $x' \in \mathbb{R}^{H_G \times W_G \times C}$ from x. The remaining channel is used as a modulation scalar $\Delta m \in \mathbb{R}^{H_G \times W_G \times 1}$ to multiply x', aiming to enhance the spatial representations. The sampled feature x' can be expressed as:

$$x' = \psi(x, p + \Delta p) \odot \Delta m \tag{4}$$

where $\odot$ denotes element-wise multiplication, and $\psi(\cdot,\cdot)$ represents the sampling process. We leverage bilinear interpolation to make this process differentiable.

Then we apply linear projections on x and x' to obtain queries(q), keys(k) and values(v) and perform multi-head attention with M heads to get the output o:

$$q = W_q x, k = W_k x', v = W_v x' \tag{5}$$

$$z^{(m)} = \texttt{softmax}(\frac{q^{(m)} k^{(m)\top}}{\sqrt{d}}) v^{(m)}, m = 1, 2, ..., M \tag{6}$$

$$o = W_o \texttt{concat}(z^{(1)}, z^{(2)}, ..., z^{(m)}) \tag{7}$$

where $W_{(\cdot)} \in \mathbb{R}^{C\times C}$ is the 1×1 convolution for linear projection, and $d = C/M$ is the number of channels for each head. $z^{(m)}, q^{(m)}, k^{(m)}, v^{(m)}$ represent the output, queries, keys and values in the m-th head. It is noteworthy that k and v are generated from the sampled feature x' with size $H_G W_G \times C$, which is smaller than q.

We consider that the sampled feature x' preserves most of the important information of x, reducing redundant contents compared with the origin feature x. It is a data-dependent way that can adapt to different inputs. In this way, we reduce the computation on redundant areas and let the model focus on more informative regions.

2.4. Computational costs analysis

Assuming the input feature size in DDTB is $H \times W \times C$, and the pre-defined patch size and the number of patches in branches are $p \times p$ equally. The hyper-parameter γ controls the number of sampled points in deformable attention. The computational costs of DDTB come mainly from DDA, which can be expressed as:

$$\Omega(\text{DDA}) = \frac{8\gamma^2+4}{\gamma^2}HWC^2 + \frac{4p^2+58}{\gamma^2}HWC + \frac{4}{\gamma^2}HW \quad (8)$$

where the computational costs are linear with the spatial size $H\times W$. It is possible to apply DDT on high-resolution images frequently appearing in image denoising tasks.

2.5. Training detail

DDT adopts a 4-stage encoder-decoder architecture. Following [7], we set the number of DDTBs from the 1st stage to Bottleneck as (4, 6, 6, 8) with the number of attention heads (1, 2, 4, 8) and 4 extra blocks for the Refinement stage. The channel number in the 1st stage is set as 32, which will increase gradually in deeper stages. We use AdamW optimizer($\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay $1e^{-4}$) and L_1 loss with 300K iterations for optimization. The learning rate is initialized as $3e^{-4}$ and reduce to $1e^{-6}$ with the cosine annealing scheduler [4]. Progressive learning strategy [7] from 128×128 to 256×256 is also used, and we utilize rotation and flips for data augmentations. All training processes are conducted on 4 NVIDIA TITAN Xp GPUs. We use the provided NTIRE traing data ($i.e.$, DIV2K and LSDIR training set) for training.

2.6. Experimental results

We use the provided noisy DIV2K validation set in development phase. We upload clean images output by our method to the validation server, and DDT obtains 30.50dB PSNR result, ranking $\sharp$11 in the result page. In the testing phase, we use the provided test data($i.e.$, 200 images from DIV2K testset and LSDIR dataset) and submit the result to the test server, and we get 29.20 PSNR result. Our DDT takes 18.4M parameters and 86GFLOPs costs with the input size $3 \times 256 \times 256$. We also evaluate DDT on the real-world denoising dataset SSID, and it get 38.83dB PSNR and 96.0 SSIM metrics which are competitive with previous SOTA methods with fewest computational costs.

References

[1] 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. *arXiv:2010.11929*, 2020. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[3] Hunsang Lee, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. Knn local attention for image restoration. In *CVPR*, pages 2139–2149, 2022. 2

[4] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 3

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 1

[6] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *CVPR*, pages 4794–4803, 2022. 2

[7] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 1, 2, 3