# NTIRE 2023 Image Super-Resolution ($\times$4) Challenge Factsheet
# Attention Retractable Frequency Transformer for Image Super-Resolution

Yajun Qiu, Qiang Zhu, Pengfei Li, Qianhui Li, Shuyuan Zhu
School of Information and Communication Engineering,
University of Electronic Science and Technology of China
qyjyun@gmail.com, zhuqiang@std.uestc.edu.cn, lipengfei202018@outlook.com,
2020010902028@std.uestc.edu.cn, eezsy@uestc.edu.cn

## 1. Team details

- Team name

  IPLAB

- Team leader name

  Yajun Qiu

- Team leader address, phone number, and email

  Address: School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China.

  Phone number: +86 15687169151

  Email: qyjyun@gmail.com

- Rest of the team members

  Qiang Zhu zhuqiang@std.uestc.edu.cn
  Pengfei Li lipengfei202018@outlook.com
  Qianhui Li 2020010902028@std.uestc.edu.cn
  Shuyuan Zhu eezsy@uestc.edu.cn

- Affiliation

  School of Information and Communication Engineering, University of Electronic Science and Technology of China.

- Affiliation of the team and/or team members with NTIRE 2023 sponsors (check the workshop website)

  N/A.

- User names and entries on the NTIRE 2023 Codalab competitions (development/validation and testing phases)

  User names: UESTC_IPLAB, QianhuiLi, Henryli

  Development/validation phase entries: 1

  Testing phase entries: 10

- Best scoring entries of the team during development/validation phase

| Username | PSNR | SSIM | Extra Data | Entries |
|---|---|---|---|---|
| QianhuiLi | 31.13 | 0.85 | 1.00 | 1 |

We also list our best scoring during testing phase,

| Username | PSNR | SSIM | Extra Data | Entries |
|---|---|---|---|---|
| Henryli | 31.18 | 0.86 | 1.00 | 3 |

- Link to the codes/executables of the solution(s)

  Codes/executables address:

  https://github.com/UESTCIPLAB/ARFT_for_NTIRE2023_Image_Super_Resolution.git

  Pre-trained model:

  https://drive.google.com/file/d/1fUDMXzHOornW4nDIxwAZbz3MynVBIKZP/view?usp=share_link

  Results:

  https://drive.google.com/file/d/1bW3FZLISpv10XxxAi1iAvx2ffWP8sPQG/view?usp=share_link

## 2. Method details

### 2.1. Attention Retractable Frequency Transformer

The overall architecture of our ARFT is shown in Fig. 1(a). Following ART [1], ARFT employs residual in residual structure to construct a deep feature extraction module. Given a low resolution image $I_{LR} \in \mathbb{R}^{H\times D\times C_{in}}$ ($H$, $D$, and $C_{in}$ are the height, width, and input channels of the input), ARFT firstly applies a 3$\times$3 convolution layer to obtain shallow feature $F_0 \in \mathbb{R}^{H\times D\times C}$, where $C$ is the dimension size of the new feature embedding. Next, the shallow feature is normalized and fed into the residual groups, which

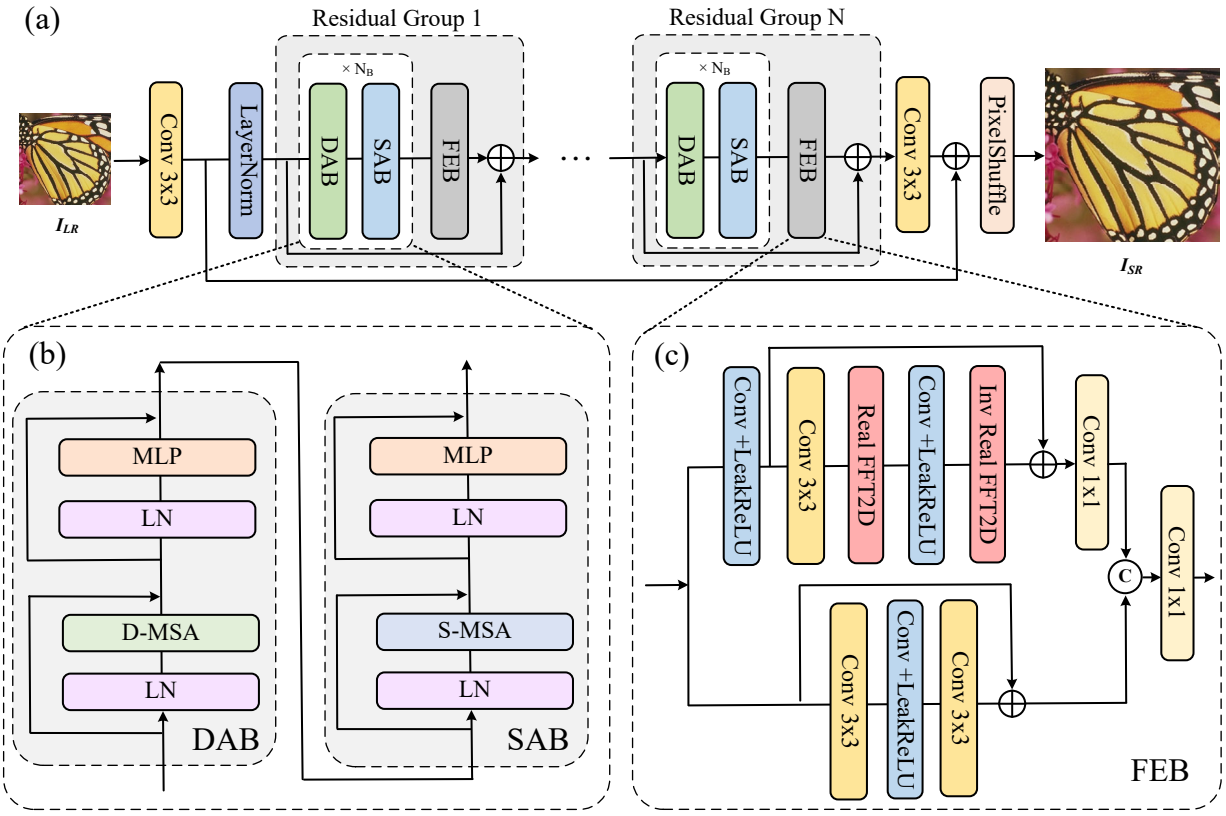


Figure 1. (a) The architecture of our proposed ARFT for image super resolution. (b) The structure of two successive attention blocks DAB and SAB with two attention modules D-MSA and S-MSA. (c) The structure of frequency enhancement block.

consist of core Transformer blocks. The deep feature is extracted and then passes through another 3×3 convolution layer to get further feature embeddings $F_1$. Then we use element-wise sum to obtain the final feature map $F_R = F_0 + F_1$. Finally, we employ the pixelshuffle layer to generate the high-resolution image $I_{\mathrm{SR}}$ from the feature $F_R$.

As shown in Fig. 1(b), we apply these two attention strategies, i.e., D-MSA and S-MSA, to design two types of self-attention blocks named as dense attention block (DAB) and sparse attention block (SAB). In the dense attention block (DAB), the dense multi-head self-attention module ( D-MSA) allows each token to interact with a smaller number of tokens from the neighborhood position of a non-overlapping $W \times W$ window. All tokens are split into several groups and each group has $W \times W$ tokens. We apply these groups to compute self-attention for $W$ times. Meanwhile, in sparse attention block (SAB), the dense multi-head self-attention module (S-MSA) is proposed to allow each token to interact with a smaller number of tokens, which are from sparse positions with interval size $I$. After that, the updates of all tokens are also split into several groups and each group has tokens.

Attention Retractable Transformer (ART) [1] demonstrates that the application of these two blocks enables SR model to capture local and global receptive field simultaneously. We also use the successive attention blocks to provide interactions for both local dense tokens and global sparse tokens.

As shown in Fig. 1(c), the FEB network architecture is composed of two primary components: a frequency branch on the up and a spatial branch on the down. We send $X$ into two distinct domains to generate $X_{\text{frequency}}$ and $X_{\text{spatial}}$ . We concatenate the outputs of up and down branch, and perform a convolution operation to obtain the final result. Specifically, $X_{\text{frequency}}$ is intended to capture the long-range context in the frequency domain and $X_{\text{spatial}}$ is utilized in the spatial domain. In frequency branch, the output feature of frequency branch is denoted as,

$$X_{\text{frequency}} = H_{\text{frequency}}(X) \tag{1}$$

where the $X$ is the input feature, $H_{\text{frequency}}$ is the frequency branch network. Specifically, we transform the conventional spatial features into the frequency domain to extract the global information by using the 2-D Fast Fourier Trans-

form (FFT). We then perform inverse 2-D FFT operation to obtain spatial domain features. The input feature $X$ is firstly refined using a convolution layer to obtain the $X_{\text{finit}}$ ,

$$X_{\text{finit}} = C_L(X) \tag{2}$$

where $C_L$ denotes a 3×3 convolution layer with a LeakyReLU activate function. Then the $X_{\text{finit}}$ is fed into the frequency domain to generate high quality frequency feature $X_{\text{frequency}}$ ,

$$X_{\text{frequency}} = C_1(\hat{\mathcal{F}}_T(C_L(\mathcal{F}_T(C(X_{\text{finit}})))) + X_{\text{finit}})), \tag{3}$$

where $C_1$ denotes a 1×1 convolution layer, $\mathcal{F}_T$ denotes a Fast Fourier Transform layer, $\hat{\mathcal{F}}_T$ denotes a inverse Fast Fourier Transform layer.

In the spatial branch, a residual module is utilized to obtain the better spatial feature. A residual connection and convolution layer is inserted to increase the expressiveness of the feature. The $X_{\text{spatial}}$ is represented as,

$$X_{\text{spatial}} = C(C_L(C(X))) + X \tag{4}$$

Finally, the output of the SFB is denoted as,

$$X_{FEB} = C_1([X_{\text{frequency}} , X_{\text{spatial}} ]) \tag{5}$$

where $[\cdot]$ denotes a concatenation operation.

### 2.2. Progressive Model Training Strategy

From the beginning of training the model to convergence, there will be many intermediate models. In general, the model with the highest performance on the validation set will be selected as the final one, and other models will be deleted. We propose a novel progressive model training strategy is used to improve SR performance. Specifically, progressive model training strategy combines the inference results from various models. We train our model with different patch sizes of training datasets in multi progressive stages. Specifically, the parameter of previous stage is utilized to initial the current model. We use three training stages for improving SR performance, our model is gradually trained using the patch with 48, 64, 84, respectively to get our final SR model.

### 2.3. Loss Function

In addition to the structure of our network, the loss function also determines whether the model can achieve good results. In low level visual tasks, such as denoising and deblurring, the $L_1$, $L_2$, and perceptual adversarial loss functions are often used to optimize neural networks. Recently, the Fast Fourier Transform loss (FFTLoss) [2] is proposed to constraint the frequency information to get better performance in super resolution task. In our experiment, we use $L_1$ loss, $L_2$ loss, the FFTLoss [2] optimize our network for generating the promising SR results.

In each training stage, we firstly use the basic loss function composed of $L_1$ loss and the FFTLoss to obtain the basic SR performance,

$$Loss_1 = \|I_{\text{HR}} - I_{\text{SR}}\|_1 + \alpha FFTLoss(I_{\text{HR}}, I_{\text{SR}}), \tag{6}$$

where $I^{\text{HR}}$ is the corresponding HR image and $\alpha$ is the penalty factor with a value of 0.1. Then, we use the $L_2$ loss to continuously train our model to improve the SR performance,

$$Loss_2 = \|I_{\text{HR}} - I_{\text{SR}}\|_2. \tag{7}$$

With three progressively training stages, our model achieves the state-of-the-art performance.

### 2.4. Datasets

We train the ARFT on a large combination training dataset, which composed of DIV2K [3], Flicker2K [5] and LSDIR [4]. Additionally, we use Bicubic to obtain the necessary degradation ×4 inputs by downsampling this training dataset. DIV2K includes 800 training images and Flicker2K includes 2650 training images. Besides, LSDIR is a new large scale dataset contains 84991 high-quality training images, 1000 validation images, and 1000 test images to fully exploited information of datasets.

### 2.5. Implementation details

Data augmentation is performed on the training data through horizontal flip and random rotation of $90°$, $180°$, and $270°$. Besides, we crop the original images into 64×64 patches as the basic training inputs for image SR. Due to using the progressively model fusion strategy, in each stage, we use the different batch size and patch size. Specifically, in three stages, we use the training batch and patch size to (32, 48),(16, 64),(8, 84), respectively. We choose ADAM to optimize our ART model with $\beta_1$ = 0.9, $\beta_2$ = 0.999, and zero weight decay. The initial learning rate is set as $2 \times 10^{-4}$ and is reduced by half as the training iteration reaches a certain number. Taking image SR as an example, we train ARFT for total 500k iterations and adjust learning rate to half when training iterations reach 250k, 400k, 450k, and 475k, where 1k means one thousand. Our ARFT is implemented on PyTorch with 4 NVIDIA RTX 3090 GPUs.

In test phase, we have not employed any kind of ensemble techniques to boost the evaluation metrics. The evaluate experimental results with PSNR and SSIM values on Y channel of images transformed to YCbCr space.

## 3. Other details

- Planned submission of a solution(s) description paper at NTIRE 2023 workshop.

  Yes, we will do it.

- General comments and impressions of the NTIRE 2023 challenge.

  Very Good!

- What do you expect from a new challenge in image restoration, enhancement and manipulation?

  You can also open a compressed video enhancement challenge.

- Other comments: encountered difficulties, fairness of the challenge, proposed subcategories, proposed evaluation method(s), etc.

  No comments.

## References


[1] Z. Jiale, Z. Yulun, Gu. Jinjin, Z. Yongbing, K. Linghe, Y. Xin, "Accurate Image Restoration with Attention Retractable Transformer," In ICLR, 2023. 1, 2

[2] D. Fuoli, L. Van Gool and R. Timofte, "Fourier Space Losses for Efficient Perceptual Image Super-Resolution," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 2340-2349. 3

[3] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. "Enhanced deep residual networks for single image super-resolution," In CVPR workshops, pages 136–144, 2017. 3

[4] https://data.vision.ee.ethz.ch/yawli/index.html. 3

[5] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming Hsuan Yang, and Lei Zhang. "Ntire 2017 challenge on single image super-resolution: Methods and results", In CVPR workshops, pages 114–125, 2017. 3