NTIRE 2023 Challenge on Image Super-Resolution (×4): Methods and Results

Yulun Zhang†, Kai Zhang†,†, Zheng Chen†, Yawei Li†, Radu Timofte†, Junpei Zhang, Kexin Zhang, Rui Peng, Yanbiao Ma, Licheng Jia, Huaibo Huang, Xiaoqiang Zhou, Yang Ai, Ran He, Yajun Qiu, Qiang Zhu, Pengfei Li, Qianhui Li, Shuyuan Zhu, Dafeng Zhang, Jia Li, Fan Wang, Chunmiao Li, TaeHyung Kim, Jungkeong Kil, Eon Kim, Yeonseung Yu, Beomyeol Lee, Subin Lee, Seokjae Lim, Soha Chae, Heungjun Choi, ZhiKai Huang, YiChung Chen, YuanChun Chiang, HaoHsiang Yang, WeiTing Chen, HuaEn Chang, I-Hsiang Chen, ChiaHsuan Hsieh, SyYen Kuo, Ui-Jin Choi, Marcos V. Conde, Sunder Ali Khawaja, Jiseok Yoon, Ik Hyun Lee, Garas Gendy, Nabil Sabor, Jingchao Hou, GuangHui He, Zhao Zhang, Baiang Li, Huan Zheng, Suiyi Zhao, Yangcheng Gao, Yanyan Wei, Jiahuan Ren, Jiayu Wei, Yanfeng Li, Jia Sun, Zhanyi Cheng, Zhiyuan Li, Xu Yao, Xinyi Wang, Danxu Li, Xuan Cui, Jun Cao, Cheng Li, Jianbin Zheng, Anjali Sarvaiya, Kalpesh Prajapati, Ratnadeep Patra, Pragnesh Barik, Chaitanya Rathod, Kishor Uplaa, Kiran Raja, Raghavendra Ramachandra, Christoph Busch

Abstract

This paper reviews the NTIRE 2023 challenge on image super-resolution (×4), focusing on the proposed solutions and results. The task of image super-resolution (SR) is to generate a high-resolution (HR) output from a corresponding low-resolution (LR) input by leveraging prior information from paired LR-HR images. The aim of the challenge is to obtain a network design/solution capable of producing high-quality results with the best performance (e.g., PSNR). We want to explore how high performance we can achieve regardless of computational cost (e.g., model size and FLOPs) and data. The track of the challenge was to measure the restored HR images with the ground truth HR images on DIV2K testing dataset. The ranking of the teams is determined directly by the PSNR value. The challenge has attracted 192 registered participants, where 15 teams made valid submissions. They achieve state-of-the-art performance in single image super-resolution.

1. Introduction

Single image super-resolution (SR) is a field of research that focuses on the recovery of high-resolution (HR) images from their low-resolution (LR) counterparts that have undergone a certain degradation process. The topic has recently garnered significant attention from both the vision and graphics communities, with a surge of interest in the field. There has been a constant growth of related papers, and substantial progress has been made with employing CNN and Transformer [5, 7, 13, 23, 30, 33, 37, 41, 59, 63, 65].

Advancements in image SR techniques have facilitated the utilization of images for various tasks. Consequently, the range of applications for these techniques has constantly expanded to various fields, including surveillance, remote sensing, automotive industry, medical image analysis, and electronics. Furthermore, the widespread adoption and usage of mobile and wearable devices provide an excellent opportunity for developing new applications and faster methods. In this challenge, we mainly focus on higher performance with larger models usually.

For image SR, the LR image is obtained by applying a specific degradation process to its HR counterpart. Variations in this process can introduce different types of noise, blurring, or other artifacts, ultimately resulting in the loss of high-frequency information. The primary objective of image SR methods is to recover as much high-frequency information as possible. There are various standard prob-
lems for image SR that depend on the degradation process. The most commonly used degradation model is bicubic down-sampling, which involves different downscaling factors. This classical degradation model facilitates the comparison of different image SR methods directly. In practice, a model performing well under Bicubic degradation can also obtain consistent performance in other degradations or even other related applications. So, Bicubic degradation can serve as a testing ground for validating the superiority of newly proposed image SR methods.

In recent times, methods based on neural networks (e.g., CNN and Transformer) have exhibited impressive performance for image restoration, especially image super-resolution (SR). SRCNN [12] first utilizes a three-layer convolutional neural network (CNN), achieving noteworthy enhancements over conventional SR methods (e.g., sparse coding based methods). Due to the fast-paced evolution of hardware technologies, numerous larger and deeper neural networks are being trained for image super-resolution. VDSR [20] builds a 20-layer network based on residual learning. RCAN [62] proposes a residual-in-Residual structure to train a model over 400 layers. Moreover, numerous spatial and channel attention mechanisms [33,37,63,65] are proposed to improve the reconstruction quality. Apart from the development of large CNN models with high performance, Transformer proposed in the natural language processing (NLP) field is introduced to alternate CNN. The core component of the Transformer is the self-attention (SA) mechanism, which can directly model long-range dependencies for an accurate restoration. Several methods have successfully applied Transformer to image SR [5,7,30,59,61]. Those methods further achieve performance gains and show promising potential for future research.

Collaborating with the NTIRE workshop, we organized a challenge specifically focused on example-based single-image super-resolution (×4). This task requires restoring high-frequency information in a high-resolution image using a single low-resolution input image and a set of prior examples that include low and corresponding high-resolution images. The aim of the challenge is to obtain a network design/solution capable to produce high-quality results with the best performance (e.g., PSNR).

This challenge is one of the NTIRE 2023 Workshop series of challenges on: night photography rendering [43], HR depth from images of specular and transparent surfaces [57], image denoising [29], video colorization [19], shadow removal [47], quality assessment of video enhancement [35], stereo super-resolution [48], light field image super-resolution [50], image super-resolution (×4) [64], 360° omnidirectional image and video super-resolution [4], lens-to-lens bokeh effect transformation [10], real-time 4K super-resolution [11], HR nonhomogenous dehazing [3], efficient super-resolution [28].

2. NTIRE 2023 Image Super-Resolution (×4) Challenge

This challenge is part of the NTIRE 2023 associated challenges, which aims are: (1) provide an overview of the latest trends and advances in image SR; (2) offer a platform for academic and industrial attendees to interact and explore collaborations. This section will elaborate on the specifics of the challenge.

2.1. DIV2K Dataset [2,45]

The DIV2K dataset comprises 1,000 RGB images with a 2K resolution, exhibiting diverse content. It is split into three sets: 800 images for training, 100 images for validation, and 100 images for testing. In this challenge, the low-resolution (LR) version of the DIV2K dataset is created by down-sampling the high-resolution images with a bicubic interpolation using a downscaling factor of 4. The validation set has already been provided to the participants, while the high-resolution images in the testing set are kept hidden throughout the challenge.

2.2. Flickr2K Dataset [31,45]

The Flickr2K dataset is a large-scale dataset of high-resolution images and is commonly used in image SR. The dataset contains 2,650 Flickr 2K images from the online photo-sharing platform Flickr. The images in the Flickr2K dataset are diverse in content, ranging from landscapes and nature scenes to portraits and still life photography. They are also diverse in quality, with some images being sharp and clear while others are blurry or contain noise.

2.3. LSDIR Dataset [27]

The LSDISR is a large-scale dataset where all high-resolution images are collected from Flickr. To ensure the pixel-level quality of the collected dataset, annotators were invited to manually inspect each of the collected images and remove the low-quality ones. The LSDIR contains 86,991 high-resolution images divided into a training set with 84,991 images, a validation set with 1,000 images, and a testing set with 1,000 images.

2.4. Track and Competition

The aim is to obtain a network design/solution capable to produce high-quality results with the best performance (e.g., PSNR).

Track: Restoration Track. The ranking of participating teams was determined based on the PSNR value of their restored high-resolution images compared to the ground truth high-resolution images on the DIV2K testing dataset.
**Challenge phases.**

1. **Development and validation phase:** The participants were given access to the training and validation image pairs of the DIV2K dataset. Additionally, participants were allowed to utilize supplementary data, such as the Flickr2K dataset and LSDIR Dataset, during training. To obtain prompt feedback, participants could submit their restored high-resolution images to the evaluation server, which would calculate the PSNR of the super-resolved image generated by their models.

2. **Testing phase:** Participants were given access to 100 LR testing images during the final testing phase, while the corresponding high-resolution ground-truth images were kept hidden from them. Subsequently, participants submitted their restored high-resolution images to the Codalab server and provided their factsheet and code to the organizers via e-mail. The organizers then verified and executed the submitted code to obtain the final results, which were later communicated to the participants at the end of the challenge.

**Evaluation protocol.** The evaluation process involves comparing the super-resolved images with their corresponding ground truth high-resolution images using standard metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [51] index, which are widely used in the literature. We calculate the PSNR and SSIM values on the Y channel of the YCbCr color space after discarding a 4-pixel boundary around each image. The average results over all the processed images of the DIV2K testing dataset are reported. We allow for a slight drop in accuracy to determine the final ranking. A code example for computing these metrics is provided at https://github.com/zhengchen1999/NTIRE2023_ImageSR_x4. The repository also includes the code and pre-trained weights of the submitted solutions.

3. **Challenge Results**

The final test results and rankings of the participating teams are presented in Tab. 1. The evaluated methods are briefly described in Sec.4, and the team members are listed in Appendix A. As can be observed from Tab. 1, the ZZPM team achieved the highest overall ranking in this Image Super-Resolution Challenge. Moreover, the average PSNR value of the top five teams is above 31 dB.

<table>
<thead>
<tr>
<th>Team</th>
<th>Rank</th>
<th>PSNR (primary)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZZPM</td>
<td>1</td>
<td>31.23</td>
<td>0.8750</td>
</tr>
<tr>
<td>Graphene</td>
<td>2</td>
<td>31.21</td>
<td>0.8665</td>
</tr>
<tr>
<td>IPLAB</td>
<td>3</td>
<td>31.18</td>
<td>0.8660</td>
</tr>
<tr>
<td>SRC-B</td>
<td>4</td>
<td>31.16</td>
<td>0.8656</td>
</tr>
<tr>
<td>LDCC</td>
<td>5</td>
<td>31.16</td>
<td>0.8655</td>
</tr>
<tr>
<td>NTU607-SR</td>
<td>6</td>
<td>30.97</td>
<td>0.8617</td>
</tr>
<tr>
<td>Swin2SR</td>
<td>7</td>
<td>30.86</td>
<td>0.8603</td>
</tr>
<tr>
<td>TUK-IKLAB</td>
<td>8</td>
<td>30.80</td>
<td>0.8595</td>
</tr>
<tr>
<td>GarasSjtu</td>
<td>9</td>
<td>30.78</td>
<td>0.8582</td>
</tr>
<tr>
<td>LVGroup_HFUT</td>
<td>10</td>
<td>30.68</td>
<td>0.8563</td>
</tr>
<tr>
<td>AhRightRightRight</td>
<td>11</td>
<td>30.65</td>
<td>0.8555</td>
</tr>
<tr>
<td>hlooooo</td>
<td>12</td>
<td>30.57</td>
<td>0.8551</td>
</tr>
<tr>
<td>chaobaer</td>
<td>13</td>
<td>30.23</td>
<td>0.8460</td>
</tr>
<tr>
<td>Alpha</td>
<td>14</td>
<td>30.02</td>
<td>0.8408</td>
</tr>
<tr>
<td>SVNIT_NYNU</td>
<td>15</td>
<td>29.92</td>
<td>0.8436</td>
</tr>
</tbody>
</table>

Table 1. Results of NTIRE 2023 Image Super-Resolution Challenge. PSNR/SSIM results are measured on the DIV2K testing dataset. The ranking of the teams is determined directly by the PSNR (primary) and SSIM (secondary).

1. **Architectures and main ideas**

Several techniques were proposed to improve the performance of image SR models during this challenge. Below, we list some of the typical ideas proposed by the participants.

- **Adopting and modifying the Transformer architecture** is the mainstream technology. Recently, some methods that adopt Transformer architecture, such as SwinIR [30], ART [59], and HAT [5], have achieved better performance than CNNs in image SR. Therefore, several teams participating in this challenge have investigated modifying existing architectures to enhance the model performance. For example, the Graphene team proposed a cross-scale attention and wavelet hallucination on the architecture of SwinIR.
- **Enhance model performance with global information.** Global information plays a crucial role in image SR as it can activate more pixels and significantly enhance the performance of the reconstruction process. Some teams try to utilize global information. For example, the SRC-B team proposed a spatial frequency block to extract the global information based on fast fourier convolution. The LVGroup_HFUT team introduced an additional global feature branch based on SwinIR.
- **Image augmentation methods could effectively improve performance.** The self-ensemble strategy [46] has been shown to be effective in improving the performance and is widely used. Some teams tried more image augment methods. The ZZPM team adopted six data augmentation methods to train five models, and proposed a new fusion method to generate the results.
- **More training data is an important factor.** In addition to DIV2K, the Flickr2K and LSDIR datasets are
allowed as training data. Models trained on large-scale datasets have better performance. Therefore, some teams try to use more extra data, e.g., OST and FFHQ. The LDCC team also selected the training data with the proposed latent discriminative cosine criterion.

5. **Advanced training strategy also boosts performance.** It is observed that some teams employed well-designed training strategies in their methods. For instance, the progressive training strategy is used, gradually increasing the patch size during the training phase.

6. **Various loss functions are considered.** The loss function is also a crucial element in neural networks. Apart from the commonly used $L_1$ and $L_2$ loss, several teams used more loss functions, like the fast fourier transform (FFT) loss and MSE loss, to achieve PSNR gain. For instance, the chaobaer team uses different losses at different stages of training.

7. **Some other techniques are also attempted.** Some teams also proposed solutions based on the U-Net architecture and Hadamard product.

### 3.2. Participants

The challenge attracted 33 registered participants, out of which 15 teams submitted valid entries. These entries set the state-of-the-art in image SR ($\times 4$).

### 3.3. Fairness

To ensure the fairness of the challenge, several rules were established regarding fair and unfair practices. Firstly, training with the DIV2K test LR images is prohibited. Secondly, training with extra datasets, such as the Flickr2K and LSDIR datasets, is allowed. Thirdly, using advanced data augmentation strategies during training is considered a fair practice.

### 3.4. Conclusions

The following conclusions can be drawn based on the above analysis of the image SR challenge results.

1. The proposed methods submitted during this challenge have contributed to the advancement of research and implementation in the field of image SR.

2. The methods based on the Transformer architecture demonstrate impressive performance and set new records regarding restoration quality.

3. High-quality, large-scale datasets play a critical role in image SR, especially for large neural networks.

### 4. Challenge Methods and Teams

#### 4.1. ZZPM

**Network Architecture.** The overall workflow of the solution proposed by the ZZPM team is illustrated in Fig. 1. The method first performs data augmentation locally on the training set on the DIV2K and LSDIR datasets. Multiple models are then used to train on the augmented data. The final result consists of the fusion of multiple models. A total of six data augmentation methods are used, including CutBlur, Blend, RGB permute, Mixup, CutMix, and CutMixup. These methods greatly increase the diversity of the data and improve the robustness of the model.

In the model selection phase, the team selects a total of five models, including Liif-EDSR, Liif-RDN, RDN-LTE, SwinIR-LTE, and SwinIR, which are widely used and have shown excellent performance in super-resolution tasks.

I. The Liif-EDSR is a deep learning model used for image super-resolution. It combines the Liif (Learned Intrinsic Image Filtering) model with the EDSR (Enhanced Deep Residual Networks) architecture to produce high-quality, high-resolution images. The Liif component is responsible for image filtering and refinement, and the EDER part is for image super-resolution. EDSR consists of multiple residual blocks, each containing convolutional layers with densely connected feature maps.

II. The Liif-RDN is similar to Liif-EDSR. The RDN component is used for super-resolution, and it consists of multiple dense blocks, each containing convolutional layers with densely connected feature maps.

III. The RDN-LTE is a variant of the Residual Dense Networks (RDN) architecture that has been optimized for low-latency applications. It uses densely connected convolutional layers to enable the model to learn and refine image features at multiple scales.

IV. The SwinIR architecture consists of three main components: the Swin Transformer encoder, the Swin Transformer decoder, and the image restoration module. It combines the Swin Transformer architecture with an image restoration module for super-resolution.

V. The SwinIR-LTE is a variant of the SwinIR model optimized for low-latency applications. It uses a smaller model architecture and a more lightweight image restoration module to reduce the computational cost and latency of the model while maintaining high-quality super-resolution results.

The main purpose of the post-processing part is to fuse the outputs of the models. Here a new fusion method is proposed. In order to fuse the outputs of the five models, first calculate the mean of the five images to be fused. And then, the MSE is calculated between each image and the mean. The smaller the MSE value, the greater the weight
assigned to the model during fusion. Finally, the final output result is obtained by performing a weighted fusion of the outputs of the five models using the calculated weights. The significance of this method is that it can avoid the final result being skewed by special bad points.

Training strategy. During the training phase, an $L_1$ loss is used for SR, and an MSE loss is used for enhancement. The training of the aforementioned models is conducted on 8 Nvidia V100 GPUs, using the Adam optimizer and multi-step learning rate decay method. The initial learning rate is set to $1 \times 10^{-4}$. In the testing phase, five models are used for the ensemble, and a series of test-time augmentation approaches, including RGB channel shuffling and horizontal and vertical flipping.

4.2. Graphene

Network Architecture. The Graphene team designed an Image Super-Resolution Transformer with Cross-Scale Attention (CSA) and Wavelet Hallucination (WH) to address the image super-resolution task. The framework is shown in Fig. 2.

In cross-scale attention, the method adds a $3 \times 3$ depth-wise convolution in the query transformation and adds multi-scale depth-wise convolutions in the key and value transformation. Such a depth-wise convolution can extract the local features and model the relationship between features across multiple scales. The cross-scale attention is used to improve the vanilla and shifted window attention in SwinIR [30].

For the wavelet hallucination, the method first hallucinates four frequency sub-bands using a linear layer, then perform depth-wise convolution followed by a linear projection separately for each sub-band. A Haar wavelet reconstruction function reconstructs the higher-resolution feature maps. Finally, a $3 \times 3$ convolution with stride 2 is applied to reduce the size of feature maps into the original one. A higher-resolution hidden feature is predicted by hallucinating features in different frequency sub-bands. Such hidden high-resolution features help to extract finer details. The wavelet hallucination is used to replace the $3 \times 3$ convolution at the end of each Transformer layer in SwinIR.

Training strategy. The model is trained on the combination of the trainsets of DIV2K, Flickr2K, and LSDIR. The training process is divided into two stages. In the first stage, the model is trained on $64 \times 64$ randomly cropped images with the Adam optimizer. The batch size, initial learning rate, and total iterations are $32, 2 \times 10^4$ and $800K$, respectively. The learning rate is halved at $300K, 500K, 650K, 700K$, and $750K$ iterations. In the second stage, the model is fine-tuned on $128 \times 128$ images with $200K$ iterations.

4.3. IPLAB

Network Architecture. The NEESR team proposed Attention Retractable Frequency Transformer (ARFT) for image super-resolution [66]. The overall architecture of the ARFT is shown in Fig. 3. Following ART [59], 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 consist of core Transformer blocks. The deep feature is extracted and then passes through another $3 \times 3$ convolution layer to get further feature embeddings $F_1$. Then the element-wise sum is used to obtain the final feature map $F_R = F_0 + F_1$. Finally, ARFT employs the pixel shuffle layer to generate the high-resolution image $I_{SR}$ from the feature $F_R$.

As shown in Fig. 3(b), two attention strategies, i.e., D-MSA and S-MSA, are applied to design two types of self-attention blocks named 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. 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 from sparse positions with interval size $I$. After that, the updates of all tokens are split into several groups, each with tokens.

As shown in Fig. 3(c), the FEB network architecture is composed of two primary components: a frequency branch on the up and a spatial branch on the down. Two distinct domains process the input feature to generate frequency feature $X_{frequency}$ and spatial feature $X_{spatial}$. The outputs of two branches are concatenated and operated by a convolution layer to obtain the final result. Specifically, $X_{frequency}$ is intended to capture the long-range context in the frequency domain, and $X_{spatial}$ is utilized in the spatial domain.

Progressive Model Training Strategy. The team proposed a novel progressive model training strategy to improve SR performance. Specifically, the progressive model
Figure 2. Graphene Team: The pipeline of the proposed method. A Transformer-based architecture is adopted. The cross-scale attention and wavelet hallucination are proposed to enhance the feature extraction.

Figure 3. IPLAB Team: (a) The architecture of the proposed ARFT for image SR. (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.
The network architecture of the proposed SwinFIR for image super-resolution.

**Loss Function.** Recently, the Fast Fourier Transform loss (FFTLoss) [16] is proposed to constrain the frequency information to get better performance in SR tasks. The team uses $L_1$ loss, $L_2$ loss, and the FFTLoss [16] to optimize their network for generating promising SR results.

In each training stage, the basic loss function composed of $L_1$ loss and the FFTLoss are used to obtain the basic SR performance,

$$Loss_1 = \|I_{HR} - I_{SR}\|_1 + \alpha FFTLoss(I_{HR}, I_{SR}),$$  \hspace{1cm} (1)

where $I_{HR}$ is the corresponding HR image and $\alpha$ is the penalty factor with a value of 0.1. Then, the $L_2$ loss is applied to continuously train the model to improve the SR performance,

$$Loss_2 = \|I_{HR} - I_{SR}\|_2.$$  \hspace{1cm} (2)

**Implementation details.** The ARFT is trained on a large combination training dataset, which is composed of DIV2K, Flicker2K, and LSDIR. Data augmentation is performed on the training data through the horizontal flip and random rotation of $90^\circ$, $180^\circ$, and $270^\circ$. Besides, the original images are cropped into $64 \times 64$ patches as the basic training inputs for image SR. Due to using the progressive model fusion strategy, different batch sizes and patch sizes are used in each stage. Specifically, in three stages, the training batch and patch size are (32, 48), (16, 64), and (8, 84), respectively. The ARFT is optimized by the Adam optimizer 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, the total iterations are 500K, and the learning rate is halved when training iterations reach 250K, 400K, 450K, and 475K. The ARFT is implemented on PyTorch with 4 NVIDIA RTX 3090 GPUs.

**4.4. SRC-B**

**Network Architecture.** Inspired by SwinIR [30] and HAT [5], the SRC-B team proposed SwinFIR [58] using Swin Transformer [36] and fast fourier convolution [8], as shown in Fig. 4. HAT proposed the Residual Hybrid Attention Group (RHAG) to activate more pixels in the image super-resolution transformer to improve performance. RHAG contains N hybrid attention blocks (HAB), an overlapping cross-attention block (OCAB), and a $3 \times 3$ convolutional layer. SwinFIR replaces the convolution ($3 \times 3$) with Fast Fourier Convolution and a residual module to fuse global and local features, Spatial-frequency Block (SFB), to
4.5. LDCC

Network Architecture. The LDCC team proposed the latent discriminative cosine criterion (LDCC) to select train data, which helps improve performance by computing the cosine similarity with test data. For this, the latent features are extracted by using the pre-trained ViT [14] model as an encoder and measure the cosine similarity [44] between the train image and the test image as follows:

$$\cos(\Theta) = \frac{1}{H \times W} \sum_{i \in (H,W)} \frac{I_{\text{train}}(i) \cdot I_{\text{test}}(i)}{||I_{\text{train}}(i)||^2 ||I_{\text{test}}(i)||^2},$$  (3)

where $I_{\text{train}}$ and $I_{\text{test}}$ denote low-resolution images in train and test datasets, respectively. $H$ and $W$ are the height and width of the low-resolution image. $i$ means the pixel position of $I_{\text{train}}$. () indicates the inner product. After that, the top 200 samples are chosen as training data according to the similarity values for each test image. By doing so, about 18K image samples are obtained from DIV2K [45], Flickr2K [31], LSDIR [27], OST [49], and AI-HUB [1] datasets.

Training strategy. The training process contains two stages according to competition phases, i.e., development and testing. At the development phase, the SR model is pre-trained on DIV2K, Flickr2K, and LSDIR datasets with initial model parameters, which are publicly available from HAT [5]. After that, the latent discriminative cosine criterion (LDCC) is applied to DIV2K, Flickr2K, LSDIR, OST, and the additional dataset obtained from AI-HUB in the testing phase. Finally, the SR model is fine-tuned on selected samples.

To train the SR model, the random rotation, horizontal flip, and random crop (i.e., $64 \times 64$ patches) are used as data augmentation. Inspired by previous works [46, 52], the self-ensemble and multi-model ensemble is applied to improve the performance.

4.6. NTU607-SR

Training strategy. The approach of the NTU607-SR is based on ART [59] and involves a 3-stage training process for fine-tuning the model.

The model is trained on DIV2K and LSDIR datasets. In the entire training process, the batch size is set to 8, and the Adam optimizer with $\beta_1=0.9$, $\beta_2=0.99$, weight decay=0 is used. The train iterations are 250K in each stage. All training images are cropped to a size of $256 \times 256$. Different loss functions are used for the three stages of the training, including $L_1$ loss, MSE loss, and PSNR loss, respectively.

The PSNR loss is calculated only on the Y channel, and for the learning rate, a lower rate is used when optimizing the model later. Regarding data augmentation, more augmentation tricks are applied in the early stages of the training,
which allows the model to converge quickly and steadily. As the training progresses, the usage of augmentation is reduced.

Specifically, in the first stage, the $L_1$ loss is used with the learning rate of $1 \times 10^{-4}$. Data augmentation includes the horizontal flip, the vertical flip, and the rotation with 90 degrees. In the second stage, the MSE loss with a learning rate of $5 \times 10^{-5}$ is applied. Finally, the PSNR loss on the Y channel is adopted, and the learning rate is set to $1 \times 10^{-5}$.

**Testing description.** The test time augmented images are used to compute the predictions and ensemble the results to obtain the final prediction. Specifically, the average of the predictions made on the original image and its three flips (horizontal, vertical, and both) are taken to generate the final prediction. This ensemble approach [46] helped reduce the impact of the noise in individual predictions and improved the overall accuracy of the model.

### 4.7. Swin2SR

**Network Architecture.** The solution of Swin2SR team studies Swin2SR [9] and HAT [5]. The Swin2SR is illustrated in Fig. 6. Note that Swin2SR is a previously published work by the team. They aim to test these methods on the novel LSDIR dataset.

In Swin2SR, the novel Swin Transformer V2 is explored as a possible update and improvement of SwinIR [30] for image super-resolution. Through this method, the major issues in training transformer vision models, such as training instability, resolution gaps between pre-training and fine-tuning, and hunger for data, can be tackled. Swin2SR is also tested on: JPEG compression artifacts removal, real-world image super-resolution, and compressed image super-resolution [9]. Experimental results also demonstrate that Swin2SR can improve the training convergence and performance of SwinIR [30].

Compare HAT [5] and Swin2SR [9]: (1) HAT has 40.84M parameters, 1998 GFLOPs, while Swin2SR has 12.23M parameters, 515 GFLOPs, and max memory allocation of 2677M. Swin2SR, therefore, is more efficient than HAT ($3 \times$ fewer params and GFLOPs) and achieves similar results.

**Training strategy.** Training and testing description can be consulted in Swin2SR [9]. The model is trained with the Adam optimizer, $L_1$ loss, flips, and rotation augmentations. The datasets: DIV2K, Flickr2K, OST, WED, FFHQ, Manga109, and LSDIR are used. The training iterations are 500K, and the progressive lr decay is the same as SwinIR [30].

### 4.8. TUK-IKLAB

**Network Architecture.** The proposed method of team TUK-IKLAB is named Dense Residual Swin Transformers (DRSTNet) for image super-resolution. The proposed method comprises four modules, i.e., Hierarchical Feature Extraction, Dense Residual Feature Enhancement, Fusion, and HR Reconstruction modules, as shown in Fig. 7.

Existing studies have revealed that using a hierarchical feature extraction module allows the network to extract meaningful representations from images at different scales in a divide-and-conquer manner [6, 39]. Furthermore, it helps the network deal with complex and severe degradation in an efficient manner. The term hierarchical is used for this
module as it extracts the representation from low-resolution (LR) images with a three-step architecture that applies convolution operation with varying strides and kernels using three different scales. For the implementation of hierarchical feature extraction modules, DRSTNet refers to the work [39]. The first step comprises padding, stride, number of channels, and kernel size, which are set to be 3, 1, 60, and 7, respectively. For the second step in this hierarchy, DRSTNet follows the same convention for feature extraction but with the values 2, 2, 60, and 5, followed by the third step that takes the values 1, 2, 60, and 3, respectively. Existing works for image restoration and super-resolution consider convolutional neural networks as feature enhancers [26, 39, 60].

Recently, some of the studies considered Swin Transformer block to enhance the features as well as model long-range dependencies [25, 30]. Shifted Windows (Swin) transformers have proven to be effective for such degradation tasks, including image restoration and super-resolution, while yielding less number of parameters. Different from the study [25, 30], the team proposes a dense residual feature enhancement (DRFE) block. As shown in Fig. 7, the DRFE block combines the Swin Transformer layers with dense residual convolutional blocks. Four layers are used for DRFE. The residual convolutional and Swin Transformer blocks are then connected in a dense scheme. This dense connection scheme was inspired by DenseNet, which helps to deal with such a complex degradation task of image super-resolution. The Swin Transformer block is further divided into Swin Transformer layers, multi-head self-attention, and layer normalization, accordingly.

The feature fusion module undertakes the enhanced features from DRFE and performs the feature level fusion with the ascending hierarchical step module. Such a fusion strategy leverages contextual information while fusing features extracted and enhanced from middle and lower branches. The features are upscaled and concatenated within the fusion module with the middle branch features. Similarly, the features from the middle branch are extracted and enhanced using DRFE, followed by the upsampling and concatenation with the feature maps from the first branch, respectively. The upsampling operation is performed using convolutional layers and pixel shuffle layer [42].

Finally, the last module is the HR reconstruction module that undertakes the enhanced features and outputs 4× higher resolution RGB image. As shown in Fig. 7, the module has a two-pixel shuffle, two convolutional layers, and two sub-pixel convolutional layers. Lastly, a convolutional layer complies with the existing works to generate the high-resolution image.
resolution image [25, 30].

Training strategy. The model is trained on Flickr2K, and the training dataset available from the competition website. For the testing stage, the DIV2K validation images and testing images from the competition website are used. The number of channels, stride, padding, and kernel size for each hierarchical branch is mentioned in the previous section. For the DRFE module, the number of channels is set to 60 for each hierarchical branch, and the number of STL layers is 6. The window size is set to 8, and the number of STBs is 2, 6, and 8 for the third, second, and first steps of the hierarchy. The number of connections in DRFE is set to 6. The network is trained using 2 NVIDIA 3060Ti GPUs with a batch size of 8. The pre-trained network is used and fine-tuned with the learning rate of $5 \times 10^{-5}$ and trained the network for 15K iterations.

4.9. GarasSjtu

Network Architecture. The GarasSjtu team designed a simple Transformer-style network for image super-resolution (STSN) [17], as shown in Fig. 8. The low-level vision model is developed based on Conv2former [18] that proposed a simple Transformer-style ConvNet model for high-level vision. In order to improve the model further, the local features can be extracted based on $3 \times 3$ convolutions in the main block of the model. In addition, the ESA [33] block is used to further improve the model.

The model is built based on three stages of shallow feature extraction, deep feature extraction, and high-resolution image reconstruction, as shown in Fig. 8. Shallow feature extraction is performed using $3 \times 3$ convolutions, which change the domain from image to feature. After that, there are five blocks of Conv2FormerGroup included in the deep feature extraction modules. In each Conv2FormerGroup contains 4 Conv2FormerB with $3 \times 3$ convolution and ESA [33]. The Conv2FormerB block is designed based on the Conv2Former block and multi-layer perception (MLP) using Layernorm before each one and using residual learning, as shown in Fig. 8. The MLP is composed of two linear layer intermediate with one point-wise convolution. The Conv2Former block is further improved by changing $1 \times 1$ convolution to $3 \times 3$ convolution for extracting local features. The Conv2Former includes two branches, the first is only $1 \times 1$ convolution, and the second is $1 \times 1$ convolution followed by depth-wise convolution with a kernel size of $k \times k$.

The Hadamard product is used to multiply the outputs of the first and second branches. Then, this Conv2Former block ends by using $3 \times 3$ convolution instead $1 \times 1$ in the original block [18]. At the end of the deep feature extraction stage, $1 \times 1$ convolution and $3 \times 3$ convolution are used. Then, residual learning is used between the input and the output of this STSN network. The final stage is the image reconstruction made by utilizing one $3 \times 3$ convolution. Then, the pixel shuffle layer is utilized for mapping features to HR image space.

Training strategy. The STSN model contains five Conv2FormerGroup blocks containing 4 Conv2FormerB, in which the number of feature maps is set to 150. Also, the channel number of the ESA is set to 32, similar to previous work [18]. DIV2K and LSDIR are used to train the model.

At the starting stage, the model is trained from scratch using the DIV2K and LSDIR datasets, with a patch size of $192 \times 192$. The batch size is 16 for 70 epochs. Then, the pre-trained weights are used to train it again for 450 epochs with the same setting based on using Warm-Start Strategy [22]. The $L_1$ loss function is used with the Adam optimizer. After the previous stage, the model is trained starting from the pre-trained weights using the DIV2K and Flickr2K datasets with an initial learning rate of $5 \times 10^{-5}$ for 200 epochs using $L_1$ loss.

4.10. LVGroup_HFUT

Network Architecture. The LVGroup_HFUT team proposed a GlobalSwinIR based on the SwinIR model since existing representative methods pay less attention to the global features of images and mainly focus on the local features of images. The diagram of the GlobalSwinIR is shown in Fig. 9. Specifically, the GlobalSwinIR follows the basic architecture of SwinIR [30]. While increasing the depth of SwinIR to obtain better detail recovery quality, another global feature branch [53] is introduced to better capture the global features of the image, thereby guiding image recovery.

Training strategy. The GlobalSwinIR is trained a total of 500K iterations. The patch size(random crop) is 192 for a high-resolution image. The Adam optimizer is used. The learning rate is initially set to $2 \times 10^{-4}$ and is halved every 100K iterations. Meanwhile, in the first 400K iterations, $L_1$ loss is used as our loss function, and in the last 100,000 iterations, $L_2$ loss is adopted.

4.11. AhRightRightRight

Network Architecture. Inspired by HNCT [15], NeWCRFs [56], the AhRightRightRight team proposed the Hybrid Attention Single Image Super Resolution Network with Conditional Random Field (HANCRF+). The HANCRF+ combines spatial attention, channel attention, and self-attention. While retaining the advantage of performing local feature extraction quickly, it can explore feature correlations along spatial and channel dimensions and activate more input pixels. Feature fusion using NeW FC-CRFs [56] uses pixel-to-pixel relations to constrain features and fuse the different layers of features.

As shown in Fig. 11, HANCRF+ consists of four parts: shallow feature extraction, deep feature extraction, feature
fusion, and up-sampling module. Specifically, a $3 \times 3$ convolution is used for extracting shallow features $f_0$ from the input LR image $I_{LR}$:

$$f_0 = H_s(I_{LR}),$$

where $H_s$ represents a $3 \times 3$ convolutional layer. Subsequently, the shallow feature is used for the deep feature extraction by a stack of Enhanced Hybrid Blocks of CNN and Transformer (EHBCT+). This process can be formulated as:

$$f_k = H^{k}_{EHBCT+}(f_{k-1}), \quad k = 1, ..., 6,$$

where $H^{k}_{EHBCT+}$ denotes the $n$-th EHBCT+. The outputs of all EHBCTs will concatenate together, and the number of channels is adjusted by a $1 \times 1$ convolution to obtain the residual features $f_c$.

Subsequently, $f_c$ and $f_k$ will be input to Neural Window Fully-connected CRFs (NeW FC-CRFs) to aggregate
the observed information from each module of the feature extraction section, and the output of each layer can be denoted as $f_m$. The outputs of the NeW FC-CRFs are concatenated together, and the number of channels is adjusted by a $1 \times 1$ convolution to obtain the features $f_{\text{c}}$ followed by a final adjustment by a $3 \times 3$ convolution to obtain $f_{\text{fin}}$.

$$f_k' = H_{\text{NeWFC-CRF}}(f_k, f_{k+1}), \quad k = 1, \ldots, 6, \quad (6)$$

where $H_{\text{NeWFC-CRF}}$ denotes the $k$-th NeW FC-CRFs. Finally, the image is reconstructed by using $3 \times 3$ convolutional layers and pixel shuffle. The results of the reconstructed super-resolution image are as follows:

$$I_{\text{SR}} = H_{\text{up}}(H_{\text{rec}}(f_{\text{fin}} + f_0)). \quad (7)$$

The $L_1$ pixel loss is minimized to optimize the parameters of HANCRF+.

$$L_1 = \| I_{\text{SR}} - I_{\text{HR}} \|_1. \quad (8)$$

**Enhanced hybrid block of C’N’N and transformer**

As shown in Fig. 11, the EHBCT+ be composed of two Swin Transformer Blocks(STB) [30], two Enhanced Spatial Attention(ESA) modules [34], Multi-Spectral Channel Attention Module [40] and one convolutional layer.

**Neural Window Fully-connected CRFs**

CRFs mainly consist of unary potentials on individual pixels or image patches and pairwise potentials on neighboring pixels or patches. The energy function of the fully connected CRFs is usually defined as

$$E(x) = \sum_i \psi_u(x_i) + \sum_{ij} \psi_p(x_i, x_j), \quad (9)$$

where $x_i$ is the value of a point in the image, $j$ is the other points in the image, $\psi_u$ represents the unary potential of the pixel, and $\psi_p$ represents the pairwise potential calculated by the pixel and its surrounding pixels.

Using neural networks instead of hand-designed potential functions can discover more potential information while performing end-to-end training. NeW FC-CRFs [56] unary potential is obtained directly from the network based on image features.

$$\psi_u(x_i) = \theta_u(I, x_i), \quad (10)$$

where $\theta$ is a parameter of the unary network, $I$ is the input color image. NeW FC-CRF divides the image into block-based windows and uses the Swin Transformer strategy [30] to calculate the pairwise potential energy within the window.

$$\psi_p = \text{SoftMax}(qK^T + b)X, \quad (11)$$

In HANCRF+, the New FC-CRFs calculate the unary potential and pairwise potential through the feature map $F$ output by EHBCT+ and the feature map $X$ output by the previous layer network, as shown in Fig. 12. A learnable weight $\alpha$ is added in front of the unary potential, allowing the network to learn adaptively from the pixels themselves and from pairs of pixels.
Training strategy. The LSDIR dataset is adopted to perform pre-training, and the DIV2K dataset is used for fine-tuning. The patch size is 64×64 for input images, and the batch size is set to 16. Data augmentation methods, including random rotation and horizontal and vertical flipping, are applied during our training. All models are trained using the Adam with $L_1$ loss. For pre-training, the learning rate is initialized to $5 \times 10^{-4}$, halved per 50 epochs. The total number of epochs is 300. For fine-tuning, the learning rate is initialized to $6.25 \times 10^{-5}$, halved per 100 epochs. The total number of epochs is 300. The model is implemented by Pytorch 1.11.0 and trained with NVIDIA GeForce RTX 3090 GPUs.

4.12. helloooo

Network Architecture. The helloooo team adopted SwinIR [30] as the base network model and combined the advantages of CNN and Transformer to further improve the model effect. As shown in Fig. 14, the network mainly comprises three modules: shallow feature extraction, deep feature extraction, and high-quality image reconstruction module.

The shallow feature extraction module adopts 3×3 convolutional layer to extract shallow features. The network adopts a long-distance connection to directly transmit low-frequency information to the reconstruction module, which can help the deep feature extraction module focus on high-frequency information and stabilize training.

The extraction of deep features is different from that of shallow features. Deep features focus on recovering lost high-frequency information. The extraction module is mainly composed of residual Swin Transformer blocks(RSTB), and each Transformer block utilizes multiple Swin Transformers for local attention and cross-window interaction. In addition, a convolution layer is added at the end of the block for feature enhancement and uses a residual join to provide a shortcut for feature aggregation.

The reconstruction module integrates shallow layer and depth features for high-quality image reconstruction. In the image reconstruction module, pixel shuffle replaces the convolution operation, and the features are up-sampled to realize reconstruction and save calculation.

Training strategy. The model is trained on DIV2K, and data augmentation, including vertical flipping, is used. The input patch size is set to 48×48. The model is trained by Adam optimizer, $\beta_1=0.9, \beta_2=0.99$. The initial learning rate is set as 0.0002, and the number of iterations is 500K. The warmup is performed 500 times per iteration. Moreover,
training strategy is divided into two stages. In the first stage, the model is trained from scratch with 255K iterations. In the second stage, the pre-trained weights of the first stage are used to accelerate the convergence rate of the model. The model is implemented using Pytorch 1.13.1 and trained using two GeForce RTX 2080ti GPUs.

4.13. chaobaer

Network Architecture. The chaobaer team developed an improved version of NAFNet to super-resolve the single image. The network architecture of the proposed method is shown in Fig. 13. NAFBlk has been proven to be efficient and easy to train in image super-resolution, so it is used as the basic stem. Most current DL-based super-resolution methods aim to learn the residual between LR and HR images. Since the residuals are mainly concentrated in edge regions, the boundary maps generated by Sobel are introduced as prior information and combine the features of a larger receptive field as the input embedding. In addition, skip connections are used for residual in residual learning.

Training strategy. During the training phase, only the DV12K is used as the training dataset. The generated LR images are cropped into patches of size 128×128, and data is augmented with random flipping. CosineAnnealingLR is used to decay the learning rate. The $L_1$ loss to train for 200 epochs and then switched to $L_2$ loss for fine-tuning. The learning rate for fine-tuning is $5 \times 10^{-5}$.


Network Architecture. The Alpha team proposed a Lightweight Residual Feature Distillation Network (LRFDN) for efficient image super-resolution, as shown in Fig. 15. The LRFDN is mainly inspired by RFDN [32] and MAFFSRN [38]. Following the overall architecture of RFDN, LRFDN consists of four stages: shallow feature extraction, deep feature extraction, and reconstruction. To further reduce the parameters and computational complexity of the original RFDB module, the number of channels of layered distillation is effectively compressed. These distillation features are extracted by three shared 1×1 and one 3×3 convolutional filter. The design of the LRFDB block is shown in Fig. 16(a).

Inspired by the residual block proposed in the RFANet [33] and MAFFSRN [38], The team proposed an enhanced fast spatial attention module (EFSA). It aims to realize spatial attention weighting to make the features more concentrated in some desired regions to obtain more representative features. The design of the EFSA module is shown in Fig. 16(c). Using the blocks above, the proposed model can better extract and integrate compact contextual information with fewer parameters.

Furthermore, it has been found that channel-wise feature rescaling is effective for shallow SR models to boost reconstruction accuracy. Therefore, a channel weighting layer is involved in each LRFDB for modeling channel-wise relationships to utilize inter-dependencies among channels with slightly additional cost. Additionally, the GeLU activation function is adopted to replace LeakyReLU in RFDN.

Training strategy. The proposed LRFDN has four LRFDBs, in which the number of feature channels is 64. During training, DIV2K and Flickr2K datasets are used for the whole process. The LRFDN model is trained from scratch with only one stage. HR patches of size 640×640 are randomly cropped from HR images, random horizontal flip, vertical flip, and rotation are introduced as the data augmentation, and the mini-batch size is set to 64. The Adam optimizer with $\beta_1=0.9$, $\beta_2=0.999$ is used to train our LRFDN model by minimizing the $L_1$ loss function. The base learning rate is set to $5 \times 10^{-4}$ equipped with co-
sine learning rate decay and 3000 warm up steps. The total number of epochs is 1000. An exponential moving average (EMA) of LRFDN weights is also maintained over training with a decay of 0.9999, which improves the performance of the model.

4.15. SVNIT_NTNU

Network Architecture. The SVNIT_NTNU used the dense convolutional neural network approach in the proposed method. Figure 17(a) depicts the proposed architecture for single image SR for scaling factors of ×4. The LR image is applied as input to the network, and it is passed to extract the salient features from it. The low-frequency features are extracted with first layers that employ a convolutional layer, while high-frequency features are extracted with residual blocks. The architecture uses the Exponential Linear Unit (ELU) activation function to improve learning performance at each layer in an efficient manner.

A new and core element of the proposed architecture is the partially densely connected design of ResBlock that preserves the high-frequency details of the SR image by retaining salient features, which is displayed in Fig. 17(b). The kernel sizes (i.e., 3×3) adopted in ResBlock recover details distributed at local and global regions. The channel attention modules are further used in concat block, shown to perform adaptive re-scaling of features on a per-channel basis. The pixel shuffle is used to upscale the feature maps to the desired scaling factor (i.e., ×4) [44]. Additionally, low-frequency features are upscaled and added to the reconstructed output to retain more versatile information.

Training strategy. The proposed network is trained using a weighted combination of $L_1$, SSIM functions with a learning rate of $1 \times 10^{-4}$, which is decayed by $1 \times 10^4$ iterations, and the same is optimized using Adam optimizer. The model has trained up to $1 \times 10^5$ iterations with a batch size of 8. The model is implemented using Pytorch.

5. Acknowledgements

This work was partially supported by the Humboldt Foundation. We thank the NTIRE 2023 sponsors: Sony Interactive Entertainment, Meta Reality Labs, ModelScope, ETH Zürich (Computer Vision Lab) and University of Würzburg (Computer Vision Lab).

A. Teams and Affiliations

NTIRE 2023 team

Title: NTIRE 2023 Image Super-Resolution (×4) Challenge

Members:
- Yulun Zhang¹ (yulun100@gmail.com),
- Kai Zhang¹ (cskaizhang@gmail.com),
- Zheng Chen² (463465810cz@gmail.com),
- Yawei Li¹ (yawei.li@vision.ee.ethz.ch),
- Radu Timofte³ (radu.timofte@uni-wuerzburg.de)

Affiliations:
1. Computer Vision Lab, ETH Zurich, Switzerland
2. Shanghai Jiao Tong University, China
3. University of Würzburg, Germany

ZZPM

Title: SwinIR-LTE Fusion Net

Members: Junpei Zhang
(22171214671@stu.xidian.edu.cn), Kexin Zhang, Rui Peng, Yanbiao Ma, Licheng Jiao

Affiliation:
Xidian University

Graphene

Title: Image Super-Resolution Transformer with Cross-Scale Attention and Wavelet Hallucination

Members: Huaibo Huang¹,²
(huaibo.huang@cripac.ia.ac.cn), Xiaqiang Zhou¹,³,
Yang Ai¹,4, Ran He¹,²,⁵

Affiliation:
IPLAB

**Title:** Attention Retractable Frequency Transformer for Image Super-Resolution  
**Members:** Yajun Qiu (qyjyun@gmail.com), Qiang Zhu, Pengfei Li, Qianhui Li, Shuyuan Zhu  
**Affiliation:** School of Information and Communication Engineering, University of Electronic Science and Technology of China

SRC-B

**Title:** SwinFIR: Revisiting the SwinIR with Fast Fourier Convolution and Improved Training for Image Super-Resolution  
**Members:** Dafeng Zhang (dfeng.zhang@samsung.com), Jia Li, Fan Wang, Chunmiao Li  
**Affiliation:** Samsung Research China - Beijing (SRC-B)

LDCC

**Title:** LDCC: Latent Discriminative Cosine Criterion  
**Members:** TaeHyung Kim (thkim07@lotte.net), Jungeekil, Eon Kim, Yeonsung Yu, Beomyeol Lee, Subin Lee, Seokjae Lim, Somi Chae, Heungjun Choi  
**Affiliation:** LOTTE DATA COMMUNICATION COMPANY, Seoul, Korea

NTU607-SR

**Title:** Transfer learning with ART for image super-resolution  
**Members:** ZhiKai Huang (brent5481@gmail.com), YiChung Chen, YuanChun Chiang, HaoHsiang Yang, WeiTing Chen, HuaEn Chang, I-Hsiang Chen, ChiaHsuan Hsieh, SyYen Kuo  
**Affiliation:**  
1. Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan  
2. Department of Electrical Engineering, National Taiwan University, Taiwan  
3. Graduate Institute of Communication Engineering, National Taiwan University, Taiwan  
4. ServiceNow, USA

Swin2SR

**Title:** Swin2SR  
**Members:** Dafeng Zhang (dfeng.zhang@samsung.com), Jia Li, Fan Wang, Chunmiao Li  
**Affiliation:** Samsung Research China - Beijing (SRC-B)

IKLAB-TUK

**Title:** Dense Residual Swin Transformers for Image Super Resolution  
**Members:** Sunder Ali Khowaja (sandar.ali@usindh.edu.pk), Jiseok Yoon, Ik Hyun Lee  
**Affiliation:**  
1. University of Sindh, Pakistan  
2. IKLAB Inc., Tech University of Korea, Siheung-Si, South Korea

GarasSjtu

**Title:** A Simple Transformer-style Network for Image Super-resolution  
**Members:** Garas Gendy (garasgaras@yahoo.com), Nabil Sabor, Jingchao Hou, Guanghui He  
**Affiliation:**  
1. Micro-Nano Electronics Department, Shanghai Jiao Tong University  
2. Electrical Engineering Department, Faculty of Engineering, Assiut University

LVGroup_HFUT

**Title:** Global Swin Transformer for Image Super-resolution  
**Members:** Zhao Zhang (cszzhang@gmail.com), Baiang Li, Huan Zheng, Suiyi Zhao, Yangcheng Gao, Yanyan Wei, Jiahuan Ren  
**Affiliation:** Hefei University of Technology

AhRightRightRight

**Title:** The Hybrid Attention Single Image Super Resolution Network with Conditional Random Field  
**Members:** Jiayu Wei (jywei@bjtu.edu.cn), Yanfeng Li, Jia Sun, Zhanyi Cheng, Zhiyuan Li, Xu Yao  
**Affiliation:**
Beijing Jiaotong University

**hellooo**

**Title:** Image super-resolution reconstruction algorithm SwinIR based on warm-up learning rate strategy.

**Members:** Xinyi Wang (493983152@qq.com), Danxu Li, Xuan Cui

**Affiliation:**

chaobaer

**Title:** NAF-Reload: Multilayer Residuals for Image Super Resolution

**Members:** Jun Cao (jonahcao0109@gmail.com), Cheng Li

**Affiliation:**

Alpha

**Title:** Lightweight Residual Feature Distillation Network for Efficient Image Super-Resolution

**Members:** Jianbin Zheng (jabir.zheng@outlook.com)

**Affiliation:** South China University of Technology, Guangzhou, Guangdong, China

SVNIT_NTNU

**Title:** Convolution Neural Network based Single Image Super-Resolution

**Members:** Anjali Sarvaiya¹
(anjali.sarvaiya.as@gmail.com), Kalpesh Prajapati¹, Ratnadeep Patra¹, Pragnesh Barik¹, Chaitanya Rathod¹, Kishor Upla¹, Kiran Raja², Raghavendra Ramachandra², Christoph Busch²

**Affiliation:**
1 Sardar Vallabhbhai National Institute of Technology
2 Norwegian University of Science and Technology, Norway

**References**


[58] 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. 2
