A Simple Transformer-style Network for Lightweight Image Super-resolution

Garas Gendy[1], Nabil Sabor[2], Jingchao Hou[1], Guanghui He[1,*]
[1]Micro-Nano Electronics Department, Shanghai Jiao Tong University, Shanghai 200240, China.
[2]Electrical Engineering Department, Faculty of Engineering, Assiut University, Assiut 71516, Egypt
garasgaras@yahoo.com, nabil_sabor@aun.edu.eg, {jingchaohou, guanghui.he}@sjtu.edu.cn

Abstract

The task of single image super resolution (SISR) has taken much attention in the last few years due to the wide range of real-world applications. However, most of the recently developed methods are computationally expensive and need much more memory. To solve this issue, we propose a simple Transformer-style network (STSN) for the image super resolution (SR) task. The idea of this method is based on using convolutional modulation (Conv2Former), which is a very simple block with a linearly compared to quadratically as in Transformers. This Conv2Former is simplified the self-attention mechanism based on utilizing only convolutions and Hadamard product. Also, the original Conv2Former is further improved to be able to extract local features, which is helpful for SR task. Based on this Conv2Former and multi-layer perceptron (MLP), we propose a convolutional modulation block (Conv2FormerB) which is similar to the Transformers block. Based on this Conv2FormerB, 3 × 3 convolution and enhanced spatial attention (ESA) block, an STSN is designed for the SISR task. This STSN achieved good results in multiple SR benchmarks. Finally, our STSN model attained 5.6 × faster run time compared to LWSwinIR.

1. Introduction

The SISR is a fundamental task of the computer vision domain. This SISR task focuses on generating an output high-resolution (HR) image corresponding to the low-resolution (LR) input one [3, 13, 41]. There are different criteria to classify this SISR task based on degradation, model parameters, the use of deep learning, etc. For example, the models are classified based on parameters into the classical SR model [3, 41], lightweight SR models [13], and mobile scale models [15]. In addition, the models can also classify conventional methods [45] and deep learning-based methods [3, 13, 41].

Recently, the model based on deep learning is taken much attention to solve the SISR task. Dong et al. [11] developed the first deep learning model for the SISR task. However, this model is a very shallow and is not able for extracting more discriminative features. Then, the authors in [29] introduced residual learning to the task of image super-resolution, which helps to increase the model depth to hundreds of layers. In addition, methods-based attention mechanism are widely used due to the ability of attention mechanism for extracting non-local features [10, 49]. Finally, the transformer-based model is successfully used to solve the SISR task [9, 25, 28, 44].

For the transformer-based model, the initial method to use Swin Transformer is made by Liang [28] (SwinIR) to solve the SR task. This SwinIR model is based on using residual Swin Transformer blocks (RSTB) to extract deep feature, which takes benefits to form non-local features and residual learning. After that, an efficient transform is suggested [34] based on designing an efficient Transformer model. Despite the success of these transformer-based models in this task, these models have large computational cost issues for applications that require low latency due to the process of computing self-attention.

In this paper, we tried to solve the Transformer-based model's problem for solving the SISR task. So, we propose a simple Transformer-style network for image super-resolution (STSN). This STSN model is based on using the original convolutional modulation, but the model is improved to extract local features. This is done by designing the convolutional modulation block (Conv2Former) layer by introducing 3 × 3 instead of 1 × 1 for local feature extraction. Then, a convolutional modulation block (Conv2FormerB) is built based on using Conv2Former and multi-layer perceptron (MLP). Afterward, a convolutional modulation group (Conv2FormerGroup) is designed based on Conv2FormerB, 3 × 3 convolution, and enhanced spatial attention (ESA) block. Finally, the STSN is built based on using Conv2FormerGroup for deep feature extraction.

The paper contribution can be summarized as the following:

- We propose Conv2FormerB, which works as a main block for the SR model, in which its computation rises linearly rather than quadratically as in Transformer.

- Based on using the Conv2FormerB as the main block, an STSN model is built for the image SR task.
- The proposed method attained the state-of-the-art on the SR benchmark in run time with a good performance. Also, an ablation study is performed to indicate the impact of each model component.

2. Related Work

In this related work section, we will discuss the work related, including two types: classical SR models and lightweight SR models.

2.1. Classical SR

For the classical SR models, these models considered as traditional models include the enhanced deep super-resolution network (EDSR) [29] that is based on using residual learning to improve the SR performance. After that, this EDSR is a further improved residual dense network [50] based on using the dense connection. The deep back-projection networks (DBPN) [16] is introduced based on exploiting iterative up- and down-sampling layers. Then, the ODE-inspired network design model [17] is developed based on using the ordinary differential equation (ODE). Also, based on a graph neural network (GNN), a cross-scale internal graph neural network (IGNN) [54] is developed. However, the recent models that solve the classical SR task are based on Transformer and attention mechanism.

The attention-based models show strong performance in solving the SISR task based on finding non-local features. One of the starting models to use this attention is the residual channel attention network (RCAN) [49] based on using the channel attention mechanism. After that, the second order attention (SAN) [10] is developed using the second-order channel attention (SOCA), which can adaptively rescale the channel-wise features. For the Transformer-based image SR, the initial model is developed based on Swin Transformer [33] to solve the image restoration task in SwinIR [28]. Afterward, the RCAN [49] is further improved [30] based on finding a proper training strategy and minimal changing in the architecture.

Moreover, a hybrid attention transformer (HAT) [7] is developed based on using overlapping cross-attention modules for improving the interaction between neighboring window features. An efficient long-range attention network (ELAN) [48] is developed by Zhang et al. based on calculating self-attention (SA) on non-overlapped feature groups. In [44], a hybrid SR network of CNNs and transformer is introduced based on using CNNS for captioning local features and Transformers to capture long-range multi-scale dependencies. A recursively defined residual network [36] is developed based on the effective use of the attention blocks. Also, a cross aggregation Transformer (CAT) [8] is introduced based on using rectangle-window self-attention (Rwin-SA) that uses parallel horizontal and vertical rectangle window attention in different heads for expanding the attention area and aggregating the features cross different windows. In [47], an attention retractable Transformer (ART) model is developed based on using both dense and sparse attention modules which permit the interaction of tokens from sparse areas for providing a wider receptive field.

2.2. Lightweight SR

For the lightweight SR model, there is strong progress in using CNNs for solving the SR task based on the low computational cost of the convolution operation. For example, the information distillation network (IDN) [21] is developed based on using the distillation of the feature maps. Then, this IDN is further improved [20, 31] based on improving the distillation task. This model is based on extracting feature channels based on the degree of channel redundancy. A hybrid pixel-unshuffled network (HPUN) [39] is suggested by Sun et al. based on using pixel-unshuffled operation for downsampling the input features and utilizing grouped convolution for decreasing the channels. Also, attention in an attention network (A^2N) [6] is developed based on the idea that not all feature maps are helpful to the model.

In addition, Yang et al. [43] developed a feature similarity ranking algorithm image SR task. An efficient non-local contrastive attention (ENLCA) [42] is introduced based on finding long-range dependencies and leveraging more relevant non-local features. In addition, the pixel attention module is further improved [52] based on reducing the model parameters and producing better performance. Then, a blueprint separable residual network (BSRN) [27] is introduced based on designing two blocks one takes the place of the redundant convolution operation. Also, Gendy et al. developed a balanced spatial feature distillation and pyramid attention (BSPAN) [14] for lightweight SR task.

Moreover, the pixel attention module is further improved [53] based on reducing the model parameters and producing better performance. After that, a residual local feature network (RLFN) [23] is developed by Kong et al. based on using three convolutional layers to learn residual for simplifying the feature aggregation. Afterward, many methods are based on using the Transformers are developed [9, 25, 38]. Then, a cross-receptive focused inference network (CFIN) [25] is developed based on using a hybrid model of CNNs and a Transformer.

In addition, N-Gram context is developed for the image super-resolution task based on using Transformer in N-Gram in the Swin Transformers network [9]. In [38], a wavelet-based Transformer for image super-resolution (WTSR) is introduced, which is able to implicitly mine

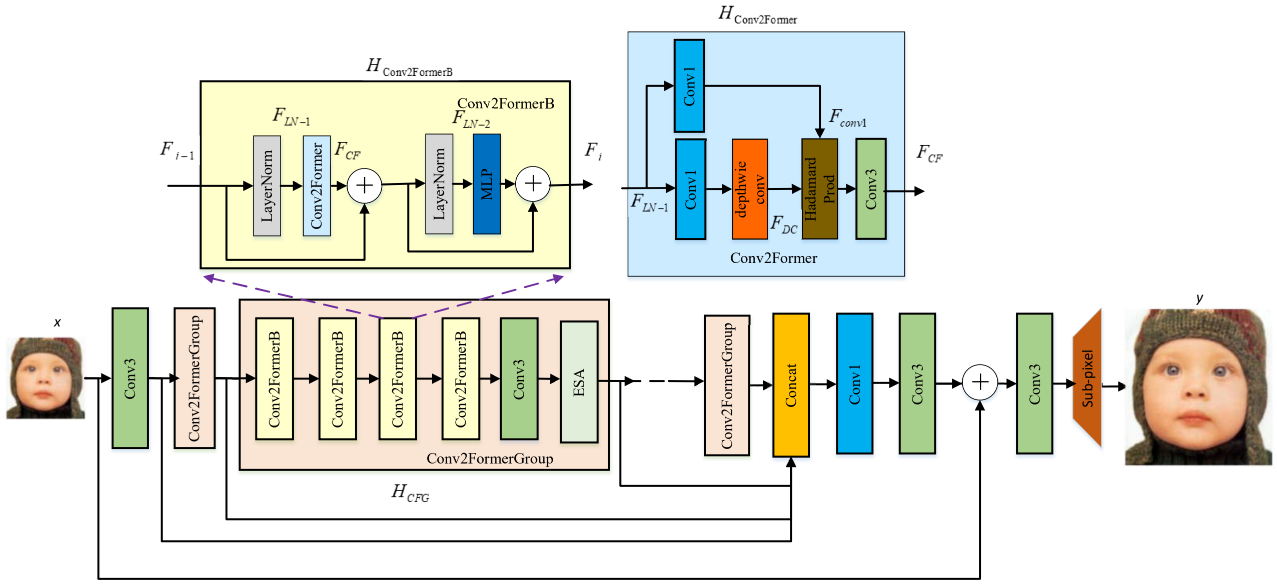


Figure 1. The architecture of the proposed STSN

the self-similarity of image patches on the wavelet domain. Even though these Transformer-based models attained good performance, these models need a long time for inference, which limit their use in some applications. Based on this limitation of these models, we tried to design a model with linear computational complexity for a fast-processing.

3. Proposed Model.

Our model is built based on using three stages of shadow and deep feature extraction modules and the image reconstruction module. A traditional convolution layer is used for shallow feature extraction. After that, we designed the deep feature extraction based on using the convolutional modulation group (Conv2FormerGroup). Next, we will discuss the details of each block.

3.1. Convolutional Modulation Group (Conv2FormerGroup)

The Conv2FormerGroup is built based on stack n Conv2FormerB blocks with 3 × 3 convlution and ESA [32]. Assuming the input feature F_{i-1}, the functions of Conv2FormerGroup, then function of this block can be represented as:

$$F_i = H_{Conv2FormerB_i}(F_{i-1}), \quad i = 1, 2, .., n \tag{1}$$

$$F_{con3} = H_{Conv3}(F_n), \tag{2}$$

$$F_{ESA} = H_{ESA}(F_{con3}), \tag{3}$$

where $H_{Conv2FormerB_i}$ is the function of i^{th} Conv2FormerB, H_{Conv3} is the function of 3 × 3 convolution, and H_{ESA} is the function of the ESA layer. This 3 × 3 convolution is utilized for extracting the local features. Also, F_{ESA} is the output feature map of the Conv2FormerGroup. Therefore, the operation of the g^{th} Conv2FormerGroup can be represented as:

$$F_g = H_{CFG}(F_{g-1}), \tag{4}$$

where H_{CFG} is the function of the Conv2FormerGroup and F_g the output of g^{th} layer of Conv2FormerGroup.

3.2. Convolutional Modulation Block (Conv2FormerB)

The Conv2FormerB is designed similarly to the Transformer block of LayerNorm and convolutional modulation (Conv2Former) layer and multi-layer perceptron (MLP). The function of i^{th} Conv2FormerB block ($H_{Conv2Former_i}$) is defined as follows:

$$F_{LN-1} = H_{LN}(F_{i-1}), \tag{5}$$

$$F_{CF} = H_{Conv2Former}(F_{LN-1}) + F_{i-1}, \tag{6}$$

$$F_{LN-2} = H_{LN}(F_{CF}), \tag{7}$$

$$F_i = F_{MLP}(F_{LN-2}) + F_{CF}, \tag{8}$$

where F_{LN-1}, F_{CF}, F_i are outputs of of Layernorm layer, Conv2Former layer, and the output feature, respectively.

Also, H_{LN}, $H_{Conv2Former}$, and F_{MLP} are the functions of the Layernorm layer, Conv2Former layer, and the MLP layer, respectively. The operation of the $HConv2Former$ will be explained in the next section.

3.3. Convolutional Modulation (Conv2Former) Layer.

The convolutional modulation (Conv2Former) layer [18] is first developed for improving the traditional self-attention. However, this Conv2Former layer has some limitations, such as it cannot extract local features, which makes it not helpful for image SR task. To solve this problem, we further replaced the 1 × 1 with a traditional 3 × 3 convolution, so the model can extract local features. Given an input feature map $F_{LN-1} \in R^{H \times W \times C}$, the Conv2Former is designed using two branches. The first one is built using only one pointwise convolution, and the second is built using another pointwise convolution followed by depth-wise convolution with a kernel size of $k \times k$. Then, these two branches are multiplied using the Hadamard product. We can express the function of this operation as follows:

$$F_{DC} = H_{DConv_{k \times k}}(H_{Conv1}(F_{LN-1})), \tag{9}$$

$$F_{conv1} = H_{Conv1}(F_{LN-1}), \tag{10}$$

$$F_{CF} = H_{Conv1}(F_{DC} \odot F_{conv1}), \tag{11}$$

where $\odot$ represents the Hadamard product. Also, H_{Conv1} defines the pointwise convolution layer. $H_{DConv_{k \times k}}$ represents a depthwise convolution with kernel size of $k \times k$. F_{DC}, F_{conv1}, F_{CF} are the outputs of the depthwise convolution, pointwise convolution, and the output of the Conv2Former layer. Therefore, the above operations of the Conv2Former can be defined as follows:

$$F_{CF} = H_{Conv2Former}(F_{LN-1}), \tag{12}$$

where $H_{Conv2Former}$ is the function of Conv2Former.

As proved in ref. [18], the computational complexity of convolution modulation is proportional linearly, the complexity of our model rises linearly because it depends on convolution modulation instead of the similarity score matrix in self-attention in the Transformer.

3.4. The simple Transformer-style network (STSN) Framework

Our STSN framework is built based on three modules of shallow feature extraction, deep feature extraction, and image reconstruction, as shown in Fig. 1. The shallow feature extraction is designed using 3 × 3 convolution (H_{conv3}) to extract coarse features (F_0) from the LR input image. So, we can define this module as:

$$F_0 = H_{conv3}(x) \tag{13}$$

Then, the deep feature extraction is made using m layers of Conv2FormerGroup (H_{CFG}).

$$F_g = H_{CFG_g}(F_{g-1}), \quad g = 1, 2, .., m \tag{14}$$

where F_g defines the output of the g layer of the Conv2FormerGroup. Following that, both the coarse feature map (F_0) and the deep feature Conv2FormerGroup ($F_g; g = 1, 2, \dots, m$) are concatenated. Then, we included both 3 × 3 and 1 × 1 to smooth the aggregated the features as fellows:

$$F_{Comb} = H_{Conv}(H_{ConCat}(F_0, F_1, F_2, \dots, F_m), \tag{15}$$

where the H_{ConCat} is mean to concatenate the the channel dimension. Also, H_{Conv} defines 1 × 1 convolution next by a 3 × 3 convolution, F_{comb} represents the overall feature map form combing both coarse and deep features. Finally, the output SR image is generated using reconstruction modules as follows:

$$y = H_{recont}(F_{comb} + F_0), \tag{16}$$

where H_{recont} defines the reconstruction function, which contains both 3 × 3 convolution and Sup-pixel upsampling. Finally, y represents the output of the model.

4. Experiment

4.1. Benchmarks

For the training section, the DIV2K [1] dataset is utilized for training our method, and an down-sampling the HR image using the bicubic down-sampling to generate the LR image. We tested the model using a benchmark of 5 datasets, including Set5 (5 images) [5], Set14 (14 images) [46], B100 (100 images) [4], Urban100 (100 images) [19], and Manga109 (109 images) [35]. Finally, the PSNR and the structural similarity index (SSIM) [40] are used for model evaluation based on using the Y channel.

4.2. Implementation Details

We set the patch size to 96, 144, 192 for scales the × 2, × 3, and × 4, respectively. Also, batch sizes of 32 are used for training. In addition, 90, 180, and 270 degrees of random rotation and horizontal flipping are used as augmentation methods for the input images. Then, the number of Conv2FormerGroup blocks (m) is set to 4 for STSN models. Additionally, the number of the Conv2FormerB (n) in the Conv2FormerGroup is empirically set to 4 for STSN models to balance the performance and computation. Moreover, the number of features is set to 50 for the STSN. The ADAM optimizer [22] is utilized with β_1 = 0.9, β_2 = 0.99 and ϵ= $1e^{-8}$. Also, the learning rate begins with 5 × 10^{-4}

Table 1. Benchmark Datasets Results for Quantitative evaluation. Best is shown in **Bold** and Second Best is shown in Underline. The Time in (ms) Averaged on DIV2K validation dataset.

Method	Scale	#Params	#Mult-Adds	Time	Set5		Set14		B100		Urban100		Manga109	
					$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$
SRCNN [11]	2	8K	52.7G	23	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9663
FSRCNN [12]	2	12k	6.0G	15	37.00	0.9558	32.63	0.9088	31.53	0.8920	29.88	0.9020	36.67	0.9710
CARN [2]	2	1,592K	222.8G	207	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
LapSRN [24]	2	251K	29.9G	360	37.63	0.9588	33.04	0.9118	31.85	0.8942	30.75	0.9133	37.55	0.9732
IDN [21]	2	553K	174.1G	250	37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196	38.01	0.9749
IMDN [20]	2	694K	158.8G	165	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
RFDN [31]	2	626K	120.4G	160	38.08	0.9606	33.67	0.9190	32.18	0.8996	32.24	0.9290	38.95	0.9773
A^2N [6]	2	1036K	247.5G	310	38.06	0.9608	33.75	0.9194	32.22	0.9002	32.43	0.9311	38.87	0.9769
LWSwinIR [28]	2	878K	195.6G	3590	38.14	**0.9611**	33.86	0.9206	**32.31**	0.9012	**32.76**	**0.9340**	39.12	**0.9783**
ELAN-light [48]	2	582K	168.4G	940	38.17	**0.9611**	**33.94**	**0.9207**	32.30	0.9012	**32.76**	**0.9340**	39.11	0.9782
STSN (Our)	2	881.9k	197.7G	640	**38.19**	**0.9611**	33.78	0.9199	32.30	**0.9013**	32.68	0.9336	**39.13**	0.9778
SRCNN [11]	3	8K	52.7G	14	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
FSRCNN [12]	3	12 k	5.0G	9	33.18	0.9140	29.37	0.8240	28.53	0.7910	26.43	0.8080	31.10	0.9210
CARN [2]	3	1,592K	118.8G	117	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
IDN [21]	3	553K	105.6G	181	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
IMDN [20]	3	703K	71.5G	82	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
RFDN [31]	3	626K	54.1G	81	34.47	0.9280	30.35	0.8421	29.11	0.8053	28.32	0.8547	33.78	0.9458
A^2N [6]	3	1036K	117.5G	158	34.47	0.9279	30.44	0.8437	29.14	0.8059	28.41	0.8570	33.78	0.9458
LWSwinIR [28]	3	886K	87.2G	1687	**34.62**	0.9289	30.54	0.8463	29.20	0.8082	28.66	**0.8624**	33.98	0.9478
ELAN-light [48]	3	590K	75.7G	405	34.61	0.9288	**30.55**	0.8463	29.21	0.8081	**28.69**	**0.8624**	**34.00**	0.9478
STSN (Our)	3	888.7K	99.9G	298	**34.62**	**0.9292**	30.54	**0.8466**	**29.22**	**0.8090**	28.59	0.8621	**34.11**	**0.9480**
SRCNN [11]	4	8K	52.7G	10	30.48	0.8626	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [12]	4	12 k	4.6G	8	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280	27.90	0.8610
CARN [2]	4	1,592K	90.9G	93	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
LapSRN [24]	4	502K	149.4G	113	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
IDN [21]	4	553K	81.87G	150	31.82	0.8903	28.25	0.7730	27.41	0.7297	25.41	0.7632	29.41	0.8942
IMDN [20]	4	715K	40.9G	58	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
RFDN [31]	4	643K	31.0G	55	32.28	0.8957	28.61	0.7818	27.58	0.7363	26.20	0.7883	30.61	0.9096
A^2N [6]	4	1047K	72.4G	110	32.30	0.8966	28.71	0.7842	27.61	0.7374	26.27	0.7920	30.67	0.9110
LWSwinIR [28]	4	897K	49.6G	945	32.44	0.8976	28.77	0.7858	**27.69**	**0.7406**	26.47	0.7980	30.92	**0.9151**
ELAN-light [48]	4	601K	43.2G	230	32.43	0.8975	**28.78**	0.7858	**27.69**	**0.7406**	**26.54**	**0.7982**	30.92	0.9150
STSN (Our)	4	898.2K	50.3G	168	**32.46**	**0.8982**	28.76	**0.7860**	27.68	0.7405	26.39	0.7971	**30.93**	0.9142

for the STSN, and half every 200 epochs. The L_1 loss function is used to train the model for 1000 epochs. In addition, the warm-start strategy [23] is used for the STSN but not used for the ablation study. Finally, we built the model using the PyTorch [37] framework and trained based on using Nvidia 2080 Ti GPUs.

4.3. Comparison with State-of-the-art SR models

This section compares our methods with 8 state-of-the-art lightweight images SR methods such as SRCNN [11], FSRCNN [12], CARN [2], LapSRN [24], IDN [21], IMDN [20], RFDN [31], A^2N [6], ELAN-light [48], and LWSwinIR [28]. We compared our model with these methods in three factors quantitative, qualitative, and model size analyses.

4.3.1 Quantitative Evaluations

To show the quantitative result of our model, five test datasets are used to compare our model with other state-of-the-art models, as illustrated in Table 1. In this case, the STSN model is used for different scale factors. It is clear from the table that our model achieved better performance compared to A^2N [6], ELAN-light [48] and LWSwinIR [28]. However, our model achieved a much faster run time due to the simple model design. For instance, the STSN model improved from 38.06 dB and 32.22 dB to 38.19 dB and 32.30 dB compared to A^2N at the scale of $\times$ 2. Also, the STSN model improved from 34.00 dB to 34.11 dB compared to ELAN-light at the scale of $\times$ 3 in the Manga109 dataset. In addition, the SSIM for our model at the scale of $\times$ 4 is improved from 0.8976 and 0.7858 to 0.8982 and 0.7860 compared to LWSwinIR for Set5 and Set14, respectively.

4.3.2 Qualitative Evaluations

To show the efficient performance of our model, we made a comparison with a state-of-the-art model in the qualitative result. As indicated in Fig. 2, our model shows good results compared to the other models. For example, for img 070 in the Urban100 dataset, the details are much clearer compared to the LWSwinIR, which represents the state-of-the-art result. In addition, for img 074 , the details of the lines are much more clearer compared to the other methods. It is clear from the result that our model can achieve good visual quality.

4.3.3 Model Size Analysis

We tried to make a comparison with other state-of-the-art models in the case of parameters, Multi-Adds, and runtime,

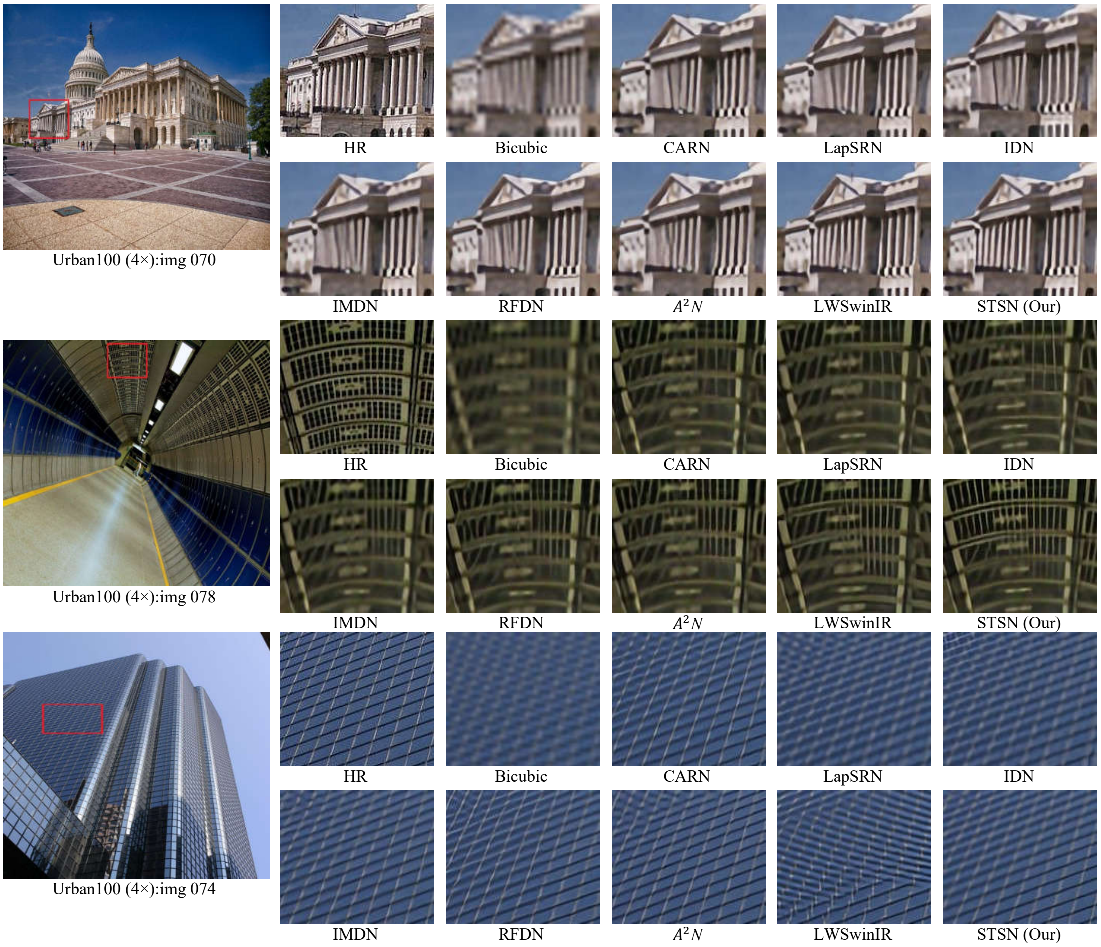


Figure 2. Urban100 Dataset Visual Comparison at $\times$ 4 SR.

as shown in Table 1. The table indicates that our model has many similarities in parameter and Multi-Adds compared to LWSwinIR [28]; however, our model has a much faster run time. In addition, our model has lower parameters, Multi-Adds, than A^2N [6] for all the SR scales. For example, our model has 17 % and 25 % fewer in the number of parameters, and Multi-Adds compared to the A^2N with much better performance. Moreover, for the run time, our model is 5.6 $\times$ faster runtime than to LWSwinIR, with mostly similar performance. This indicates that our model is efficient for the application that needs faster models.

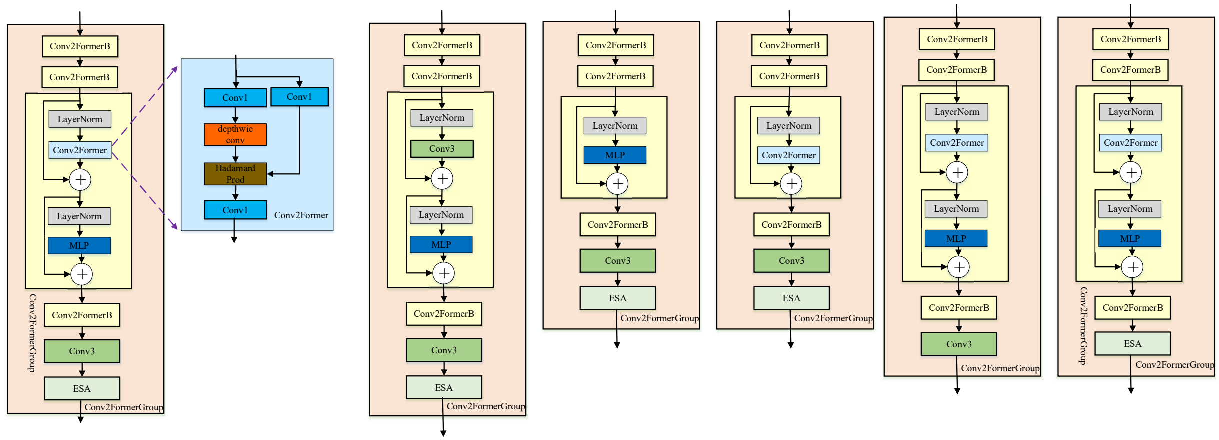
4.4. Ablation Study

In our ablation, the STSN model is used before using the warm-start strategy to save time. Also, we used the model at scale of $\times$ 2. The study aim is to study the impact of factors, the impact of some modules in the Conv2Former, the impact of modules in the Conv2FormerB, the impact of modules in the Conv2FormerGroup, and the impact of the warm-Start Strategy.

4.4.1 Ablation Study in the Conv2Former Block

The impact of using 3 $\times$ 3 instead of 1 $\times$ 1. To illustrate why the 1 $\times$ 1 conv of the original Conv2Former is replaced by 3 $\times$ 3 conv, the improved Conv2Former and the original one are used independently in the proposed model, as indicated in Fig. 3a (Model 1). The results are listed in Table 2, where 1^{st} row represents the results of using improved Conv2Former, and 2^{nd} row represents the results of using the original Conv2Former. The result indicated that the 3 $\times$ 3 greatly impacts all test datasets, especially for the Urban100 dataset. This is because the 3 $\times$ 3 can extract local features, leading to performance improvement.

The impact of using attention module. In this task, all the contents of the conv2Former block are removed, except of the 3 $\times$ 3 to indicate the impact of the attention module, as indicated in Fig. 3b (Model 2). The obtained results are indicated in Table 2, where 1^{st} row represents the results of using the Conv2Former, and 3^{rd} row represents the results without using the attention module. The results show that



(a) Model 1 (b) Model 2 (c) Model 3 (d) Model 4 (e) Model 5 (f) Model 6

Figure 3. The Models Used in our Ablation Study.

Table 2. The Ablation Study on the Conv2Former Block at the Scale × 2

Method	#Params	#Mult-Adds	Set5		Set14		B100		Urban100		Manga109	
			$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$
STSN	881.9k	197.7G	38.16	0.9611	33.74	0.9191	32.29	0.9012	32.58	0.9326	39.06	0.9775
STSN W conv1	561.9k	123.9G	38.13	0.9610	33.73	0.9194	32.26	0.9008	32.48	0.9317	39.01	0.9775
STSN W/O attention	702.7k	156.3G	38.08	0.9607	33.61	0.9176	32.21	0.9000	32.20	0.9291	38.85	0.9774

Table 3. The Ablation Study on the Conv2FormerB Block at Scale × 2

Method	#Params	#Mult-Adds	Set5		Set14		B100		Urban100		Manga109	
			$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$
STSN	881.9k	197.7G	38.16	0.9611	33.74	0.9191	32.29	0.9012	32.58	0.9326	39.06	0.9775
STSN W/O Conv2Former	339.5	73.2G	37.97	0.9605	33.58	0.9175	32.16	0.8994	32.06	0.9275	38.70	0.9770
STSN W/O MLP	701.1k	156.5G	38.11	0.9607	33.74	0.9195	32.23	0.9004	32.30	0.9306	38.86	0.9772

the attention module has a big impact on performance. For instance, the PSNR dropped from 33.77 dB to 33.61 dB on the Set14 dataset. So, these results show that the attention module can greatly impacts the performance.

4.4.2 Ablation Study in the Conv2FormerB Block

The impact of using the Conv2Former block. In this task, the conv2Former block is removed from the Conv2FormerB to indicate the impact of the conv2Former on the performance, as indicated in Fig. 3c (Model 3). The obtained results are listed in Table 3, where 1^{st} row represents the results of using Conv2Former, and 2^{nd} row represents the results of model without using the conv2Former layer in conv2FormerB block. The results indicate that this block has an impact on performance. For example, the PSNR dropped from 39.06 dB to 38.70 dB on the Manga109 dataset. So, the conv2Former block in the Conv2FormerB block can greatly impact the performance due to its ability to extract local and non-local features.

The impact of using the MLP block. In this task, the MLP block is not included from the Conv2FormerB to indicate the impact of the MLP on the performance, as indicated in Fig. 3d (Model 4). The obtained results are shown in Table 3, where 1^{st} row represents the results of using the full model, and 3^{rd} row represents the results without using MLP layer in conv2FormerB block. The results show that the MLP block has a significant impact on performance. For instance, the PSNR decreased from 32.58 dB to 32.30 dB on Urban100 dataset. So, the MLP block in the Conv2FormerB block could impact the performance due to its ability to make feature transformations.

4.4.3 Ablation Study in the Conv2FormerGroup Block

The impact of using the ESA block. In this task, the ESA is not included in the Conv2FormerGroup to show the impact of the ESA on the performance, as shown in Fig. 3e (Model 5). The obtained results are indicated in Table 4, where 1^{st} row represents the results of using the the full model, and 2^{nd} row represents the results without using ESA layer. The results illustrates that the ESA convolution impacts performance. For example, the PSNR decreased from 38.16 dB to 38.10 dB on the Set5 dataset. So, these

Table 4. The Ablation Study on the Conv2FormerGroup Block at Scale × 2

Method	#Params	#Mult-	Set5		Set14		B100		Urban100		Manga109	
		Adds	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$
STSN	881.9k	197.7G	38.16	0.9611	33.74	0.9191	32.29	0.9012	32.58	0.9326	39.06	0.9775
STSN W/O ESA	855.3k	195.9G	38.10	0.9609	33.78	0.9192	32.27	0.9009	32.43	0.9314	39.00	0.9777
STSN W/O conv3	791.7k	176.8G	38.12	0.9609	33.73	0.9194	32.26	0.9008	32.44	0.9313	39.00	0.9777

Table 5. The Ablation Study on Warm-Start Strategy at Scale × 2

Method	#Params	#Mult-	Set5		Set14		B100		Urban100		Manga109	
		Adds	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$PSNR$	$SSIM$
STSN	881.9k	197.7G	38.16	0.9611	33.74	0.9191	32.29	0.9012	32.58	0.9326	39.06	0.9775
STSN W Warm-Start	881.9k	197.7G	38.19	0.9611	33.78	0.9199	32.30	0.9013	32.68	0.9336	39.13	0.9778

results show that the ESA block could impact the performance due to its ability to use spatial attention to improve the performance.

The impact of using the 3 × 3 convolution. In this task, the 3 × 3 convolution is removed from the Conv2FormerGroup to show the impact of the 3 × 3, as indicated in Fig. 3f (Model 6). The obtained results are shown in Table 4, where 1^{st} row represents the results of using the full model, and 3^{rd} row represents the results without using 3 × 3 convolution. The results show that the 3 × 3 convolution impacts performance greatly. For instance, the PSNR decreased from 32.58 dB to 32.44 dB on the Urban100 dataset. So, these results show that the 3 × 3 convolution could impact the performance due to its ability to extract local features.

4.4.4 The Ablation Study on Warm-Start Strategy

In this task, the warm-start strategy [23] is used to retrain the model again, starting from the pre-train model on the same scale. The obtained results are indicated in Table 5, where 1^{st} row represents the results of using the full model, and 2^{nd} row represents the results with this strategy. The results indicate the model performance on PSNR improved from 32.58 dB and 39.06 dB to 32.68 dB and 39.13 dB on the Urban100 and Manga109, respectively. So, these results show that this strategy can impact the performance without any additional parameters and Multi-Adds.

4.5. STSN for NTIRE 2023 Challenge

We took part in NTIRE 2023 Image Super-Resolution Challenge [51], and our model achieved a good result, as shown in Table 6. Our STSN model is changed from the STSN model in the paper; it 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 [32], and we set the RGB range to 255, not to 1, as in the paper. In our training, we used DIV2K and LSDIR [26] to train the model. After that, the model is trained in the following steps. At the starting stage, the model is trained from scratch using the DIV2K and LSDIR [26] datasets, with a patch size of 192 × 192. We train our model using a batch size of 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 the warm-start strategy [23]. In this training, L_1 loss function is used with the Adam optimizer. After the previous stage, we trained the model starting from the previous pre-trained weights using the DIV2K and Flickr2K datasets with an learning rate of 5 $\times 10^{-5}$ for 200 epochs using L_1 loss. Using this model design and training strategy, we got among the best 10 teams of the competition.

Table 6. The Results of the Top 10 Teams on NTIRE 2023 Challenge

Rank	Team Name	PSNR	SSIM
1	ZZPM	31.232	0.8750
2	Graphene	31.200	0.8665
3	IPLAB	31.181	0.8660
4	Samsung Research China - Beijing (SRC-B)	31.163	0.8656
5	LDCC	31.155	0.8655
6	NTU607_SR	30.966	0.8617
7	Swin2SR	30.859	0.8603
8	TUK-IKLAB	30.804	0.8595
9	GarasSjtu (Our)	30.780	0.8582
10	AhRightRightRight	30.649	0.8555

5. Conclusion

In this paper, we propose a simple Transformer-style network (STSN) for single image super-resolution (SISR). The STSN is designed to be similar to the Transformers block but with linear complexity. The idea of this method is based on using convolutional modulation (Conv2Former), which is a very simple block with a linearly compared to quadratically in Transformers. This model simplified the self-attention mechanism based on using only convolutions and Hadamard product. Our methods achieved faster run time based on the experimental result in SR models.

Acknowledgment. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2204500, in part by the National Natural Science Foundation of China under Grant. 62074097.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 4

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 252–268, 2018. 5

[3] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 1

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 4

[5] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 4

[6] Haoyu Chen, Jinjin Gu, and Zhi Zhang. Attention in attention network for image super-resolution. *arXiv preprint arXiv:2104.09497*, 2021. 2, 5, 6

[7] X Chen, X Wang, J Zhou, and C Dong. Activating more pixels in image super-resolution transformer. arxiv 2022. *arXiv preprint arXiv:2205.04437*. 2

[8] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xin Yuan, et al. Cross aggregation transformer for image restoration. *Advances in Neural Information Processing Systems*, 35:25478–25490, 2022. 2

[9] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. *arXiv preprint arXiv:2211.11436*, 2022. 1, 2

[10] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019. 1, 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014. 1, 5

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 391–407. Springer, 2016. 5

[13] Garas Gendy, Guanghui He, and Nabil Sabor. Lightweight image super-resolution based on deep learning: State-of-the-art and future directions. *Information Fusion*, 94:284–310, 2023. 1

[14] Garas Gendy, Nabil Sabor, Jingchao Hou, and Guanghui He. Balanced spatial feature distillation and pyramid attention network for lightweight image super-resolution. *Neurocomputing*, 509:157–166, 2022. 2

[15] Garas Gendy, Nabil Sabor, Jingchao Hou, and Guanghui He. Real-time channel mixing net for mobile image super-resolution. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 573–590. Springer, 2023. 1

[16] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. 2

[17] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019. 2

[18] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022. 4

[19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 4

[20] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th acm international conference on multimedia*, pages 2024–2032, 2019. 2, 5

[21] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 723–731, 2018. 2, 5

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[23] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 766–776, 2022. 2, 5, 8

[24] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 5

[25] Wenjie Li, Juncheng Li, Guangwei Gao, Jiantao Zhou, Jian Yang, and Guo-Jun Qi. Cross-receptive focused inference network for lightweight image super-resolution. *arXiv preprint arXiv:2207.02796*, 2022. 1, 2

[26] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. 8

[27] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual network for efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 833–843, 2022. 2

[28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1, 2, 5, 6

[29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2

[30] Zudi Lin, Prateek Garg, Atmadeep Banerjee, Salma Abdel Magid, Deqing Sun, Yulun Zhang, Luc Van Gool, Donglai Wei, and Hanspeter Pfister. Revisiting rcan: Improved training for image super-resolution. *arXiv preprint arXiv:2201.11279*, 2022. 2

[31] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 41–55. Springer, 2020. 2, 5

[32] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2359–2368, 2020. 3, 8

[33] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[34] Zhisheng Lu, Hong Liu, Juncheng Li, and Linlin Zhang. Efficient transformer for single image super-resolution. *arXiv preprint arXiv:2108.11084*, 2021. 1

[35] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 4

[36] Alexander Panaetov, Karim Elhadji Daou, Igor Samenko, Evgeny Tetin, and Ilya Ivanov. Rdrn: Recursively defined residual network for image super-resolution. In *Proceedings of the Asian Conference on Computer Vision*, pages 4110–4125, 2022. 2

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[38] Jinye Ran and Zili Zhang. Lightweight wavelet-based transformer for image super-resolution. In *PRICAI 2022: Trends in Artificial Intelligence: 19th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2022, Shanghai, China, November 10–13, 2022, Proceedings, Part III*, pages 368–382. Springer, 2022. 2

[39] Bin Sun, Yulun Zhang, Songyao Jiang, and Yun Fu. Hybrid pixel-unshuffled network for lightweight image super-resolution. *arXiv preprint arXiv:2203.08921*, 2022. 2

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[41] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020. 1

[42] Bin Xia, Yucheng Hang, Yapeng Tian, Wenming Yang, Qingmin Liao, and Jie Zhou. Efficient non-local contrastive attention for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2759–2767, 2022. 2

[43] Haoran Yang, Gwanggil Jeon, Kai Liu, Yiguang Liu, and Xiaomin Yang. Feature similarity rank-based information distillation network for lightweight image superresolution. *Knowledge-Based Systems*, page 110437, 2023. 2

[44] Jinsu Yoo, Taehoon Kim, Sihaeng Lee, Seung Hwan Kim, Honglak Lee, and Tae Hyun Kim. Enriched cnn-transformer feature aggregation networks for super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4956–4965, 2023. 1, 2

[45] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal processing*, 128:389–408, 2016. 1

[46] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 4

[47] Jiale Zhang, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Accurate image restoration with attention retractable transformer. *arXiv preprint arXiv:2210.01427*, 2022. 2

[48] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 649–667. Springer, 2022. 2, 5

[49] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 1, 2

[50] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 2

[51] Yulun Zhang, Kai Zhang, Zheng Chen, Yawei Li, Radu Timofte, et al. Ntire 2023 challenge on image super-resolution (x4): Methods and results. In *Computer Vision and Pattern Recognition Workshops*, 2023. 8

[52] Lin Zhou, Haoming Cai, Jinjin Gu, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Yu Qiao, and Chao Dong. Efficient image super-resolution using vast-receptive-field attention. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 256–272. Springer, 2023. 2

[53] Lin Zhou, Haoming Cai, Jinjin Gu, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Yu Qiao, and Chao Dong. Efficient image super-resolution using vast-receptive-field attention. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 256–272. Springer, 2023. 2

[54] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. *Advances in neural information processing systems*, 33:3499–3509, 2020. 2