Received: 22 October 2023 | Revised: 8 November 2023 | Accepted: 4 January 2024 | Published online: 16 January 2024

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 3(2) 168–178
DOI: 10.47852/bonviewAIA42021930

BON VIEW PUBLISHING

Toward Faster and Efficient Lightweight Image Super-Resolution Using Transformers and Fourier Convolutions

Vishal Ramesha[1], Yashas Kadambi[1], B. S. Abhishek Aditya[1], T. Vijay Prashant[1] and S. S. Shylaja[1,*]

[1]*Department of Computer Science and Engineering, PES University, India*

Abstract: Lightweight single-image super-resolution has seen many advances in recent times. Transformer-based methods have achieved great improvements over convolutional neural network-based methods. This is mainly driven by the transformer's ability to effectively model long-range dependencies and retain textures in images. However, these transformer-based approaches have many parameters and are computationally expensive during inference. In this work, we propose SWIFT, a hybrid of transformers and fast Fourier convolutions (FFC) for lightweight single-image super-resolution. We designed a novel dual spectrum frequency block (DSFB) that processes features in both the spatial domain and the Fourier domain. DSFB allows us to effectively maintain global context in features and extract high-frequency information. Additionally, to mitigate the frequency-erasing nature of transformers, we introduce SwinV2$^+$ transformers that use attention scaling to promote high-frequency information. Experimental results on popular benchmarking datasets show that SWIFT outperforms state-of-the-art transformer-based methods in the realm of lightweight SISR, using 34% fewer parameters and being up to 60% faster during inference.

Keywords: lightweight image super-resolution, transformers, Fourier convolutions

1. Introduction

Image super-resolution is a low-level computer vision task that involves generating a high-resolution (HR) image from a low-resolution (LR) image. This domain has become an active area of research in the computer vision community. Several models have been proposed [1–6] that address this task by using convolutional neural networks (CNNs). However, these CNN-based models work well when the size of the model is large and requires many training examples to produce decent results. Convolution-based methods concentrate on careful network designs, such as residual connections [2, 4] and dense networks [7, 8]. These models perform significantly compared to traditional upsampling methods, such as bicubic or bilinear interpolation. However, these models do not scale very well when applied to larger images. This is because convolutions work on local interactions which makes them less effective in modeling long-range dependencies. As a result, these models are not able to effectively maintain the global context while generating large HR images.

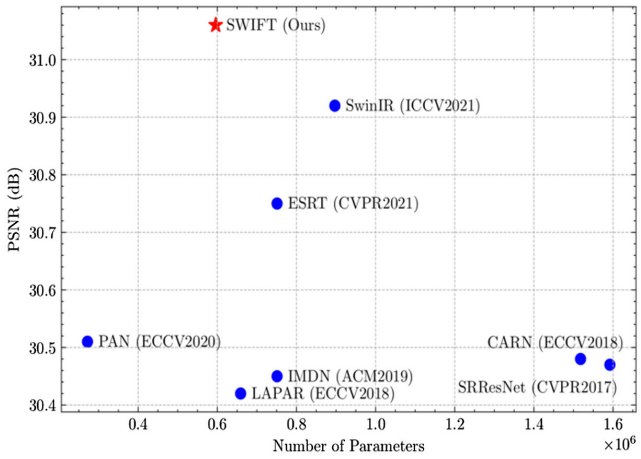
With the introduction of transformers in reference [9] and vision transformers [10] showing great success in high-level computer vision tasks, several works have been proposed that make use of transformers in low-level vision tasks. Transformer models employ a self-attention mechanism that helps the models to capture global context effectively and perform better in wide range of vision problems [1, 11–13]. To make transformers work with images, images are split into patches which are processed independently. This causes some artifacts around the borders of each patch. To mitigate this, the patches are usually overlapped, but this introduces extra computational overhead.

A recent work on SwinIR [14] proposed a transformer-based model that outperforms all the previous state-of-the-art methods in image super-resolution. SwinIR uses Swin transformers [15] which makes use of window attention to reduce complexity of attention computation on large images. This overcomes the major drawback of CNNs by being able to process images of larger sizes. Another notable model is ESRT [16] which uses a hybrid of CNNs and transformers to achieve results close to SwinIR using fewer computations. In order to increase the receptive field of convolutions, Chi et al. [17] propose a fast Fourier convolution (FFC) module. FFC converts the feature maps from the spatial domain to Fourier domain. This allows the model to capture global context without the need for an expensive self-attention module. SwinFIR proposed by Zhang et al. [18] improved on SwinIR by employing the FFC. SwinFIR [18] makes use of Spatial Frequency Block (SFB) on top of SwinIR and obtains state-of-the-art results in the field of image super-resolution.

Recently, the focus has shifted more toward efficient and lightweight methods for image super-resolution as shown in Figure 1. Typically, it is achieved by reducing model parameters, adopting re-parameterizable blocks, and carefully tuning model architectures. Several methods [8, 19–23] have been proposed over the years. However, the main challenge in choosing efficiency is that these methods often sacrifice on the quality of

***Corresponding author:** S. S. Shylaja, Department of Computer Science and Engineering, PES University, India. Email: shylaja.sharath@pes.edu

Figure 1
Comparison of model parameter VS PSNR scores of popular image SR methods on Manga109 for ×4 scale



the reconstructed image. In this work, we propose SWIFT, a hybrid of transformers and FFCs, for lightweight single-image super-resolution. Our method consists of three modules: a feature extraction module (FE), a high-frequency extraction module (HFE), and a parameter-efficient upsampling module for image reconstruction. In this work, we design a novel dual spectrum frequency block (DSFB) that uses FFCs as proposed by Chi et al. [17] for extracting features in Fourier domain and shared adaptive residual feature blocks (ARFB) layers [16] to extract features from the spatial domain. The combination of processing features in both the spatial domain and the Fourier domain allows DSFB to effectively model long-range dependencies and extract rich high-frequency information from features. Additionally, to overcome the frequency-erasing nature of transformers, we introduce SwinV2$^+$ transformers that modify the attention mechanism in SwinV2 transformers by using attention scaling to promote high-frequency information. Our method SWIFT achieves 0.10 ∼ 0.20 dB improvement in PSNR scores compared to other methods in lightweight single-image super-resolution. Our method has the lowest model size (by using ∼34% less parameters) and is upto 60% faster in inference compared to previous transformer-based methods.

2. Literature Review

In recent times, image super-resolution is an actively researched topic. Numerous models have evolved over the years fueled by the rapid advancements in deep learning technology. SRCNN [24] was the first to introduce CNN to the field of image super-resolution. Its simple network consisted of just three convolution layers. VDSR [25] improved upon this by using a deeper network to enable the model to converge faster. EDSR [4] removed some unnecessary modules in VDSR to make the model more efficient, and this enabled the model to have a large number of parameters. HAN [26] and SAN [27] further extend the usage of attention mechanisms to map interdependencies and enhance the expression and correlation of feature learning. CSNLN [28] proposed a non-local attention module that can give additional weight to features from all scales. NLSA [29] developed a sparse representation of non-local operations, preserving the robustness of the operations while being extremely efficient.

The limitation of a small receptive field in CNNs has been overcome by the introduction of self-attention mechanisms in transformers [9]. ViT [10] was the first transformer model which proved that transformers tend to work well on image tasks as well. IPT [30] leveraged the power of transformers to pre-train a transformer model for the underlying visual task in the feature mapping stage relying on large model size of more than 100 million parameters and huge datasets containing well over a million images. SwinIR [14] proposed a transformer-based model by making use of Swin transformers [15] that are stacked together. HAT [31] developed a model that combines channel attention, overlapping cross-attention, and self-attention to achieve great results but has a model size of ∼40 million parameters. Further SwinFIR [18] proposes the spatial frequency block (SFB) that uses Fourier transformations to extract comprehensively detailed and stable features. LaMa [32] proposes a new network focused on using FFC on image restoration problems. Inspired by these papers, we propose SWIFT, a hybrid of transformers and FFC, that tackles the problems of limited receptive field in CNN and to better model the long-range dependencies.

3. Research Methodology

3.1. Model design

The model architecture for SWIFT is shown in Figure 2. The architecture consists of three modules: the FE module, HFE module consisting of stacked FSTBs, and a parameter-efficient upsampling module for image reconstruction.

The LR images are first passed to the feature extraction module, containing a single convolution layer. This module extracts low-frequency details from raw input images and converts the input dimensions to a higher dimensional space, making it suitable for feature extraction at deeper layers. Let $I_{LR} \in R^{H\times W\times C_{in}}$ be the input raw LR images to FE module where H, W, and C_{in} are height, width, and channels, respectively, and the FE module can be represented as shown in Equation (1):

$$F_0 = H_{FE}(I_{LR}), \tag{1}$$

where $H_{FE}(\cdot)$ consists of a single 3 × 3 convolution layer for extracting feature maps from the LR input image I_{LR} and F_0 is the output features containing low-frequency information.

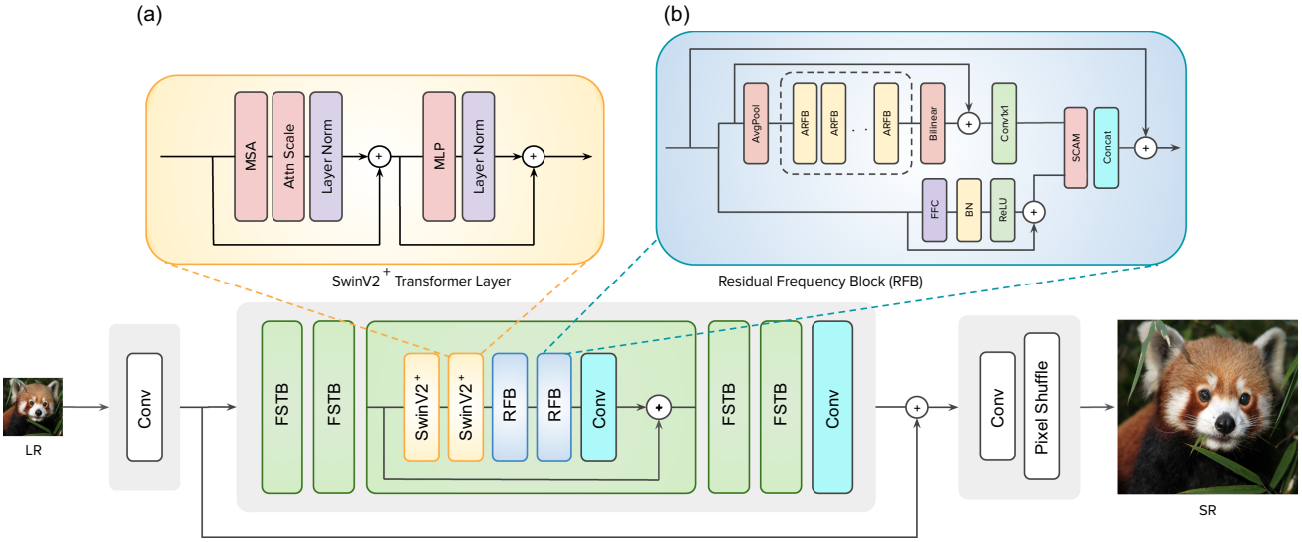
The feature maps F_0 extracted by the FE module are then passed to the HFE module that consists of several stacked Fourier-Swin transformer blocks (FSTBs). These FSTBs specialize in extracting high-frequency information from the low-frequency feature maps F_0 which helps to generate powerful feature representations. Additionally, a parameter-efficient convolution layer is used at the end of HFE module. The convolution operation is used to perform additional processing on the high-frequency features extracted before passing them to the upsampling module for image reconstruction. The HFE can be module can be represented as shown in Equation (2):

$$F_{HFE} = H_{HFE}(F_0), \tag{2}$$

where the output feature maps $F_{HFE} \in \mathbb{R}^{H\times W\times C}$ consist of rich information that helps the upsampler to perform better image reconstruction.

Finally, a parameter-efficient upsampling module is used for upsampling the features maps to the HR size. This module consists of a convolution layer and parameter-efficient pixel shuffle upsampling layers [33] that takes in the feature maps F_0,

Figure 2
SWIFT architecture for lightweight image super-resolution



extracted by FE module, and the high features F_{HFE} as input and upsamples to the desired scales. The parameter-efficient upsampling module can be represented as shown in Equation (3):

$$I_{SR} = H_{UP}(F_0 + F_{HFE}), \tag{3}$$

where $I_{SR} \in \mathbb{R}^{H\times W\times C_{in}}$ is the super-resolved image generated by the upsampling module H_{UP}.

3.2. FSTB

The design of our novel FSTB is shown in Figure 2. Each FSTB consists of a series of SwinV2$^+$ transformer layers followed by a series of DSFBs. The number of SwinV2$^+$ transformers and DSFBs can be configured based on the model type. A parameter-efficient convolution layer is used at the end of each FSTB to add the inductive bias back, which helps the model learn better features.

Let the number of SwinV2$^+$ transformers and DSFBs be represented by M and N, respectively, in the i-th FSTB, and let $F_{i,0}$ represent the input feature maps. The operation performed by the FSTB can be represented as shown in Equation (4):

$$\begin{aligned} F_{i,j} &= H_{SwinV2^+_{i,j}}(F_{i,j-1}), \quad j = 1,2,\ldots,M, \\ F_{i,k} &= H_{DSFB_{i,k}}(F_{i,k-1}), \quad k = 1,2,\ldots,N, \\ F_{i,out} &= H_{conv_i}(F_{i,N}) + F_{i,0}, \end{aligned} \tag{4}$$

where $H_{SwinV2^+_{i,j}}(\cdot)$ is the j-th SwinV2$^+$ transformer in the i-th FSTB and $H_{DSFB_{i,k}}(\cdot)$ is the k-th DSFB in the i-th FSTB. A residual connection is used in each FSTB, which helps to aggregate features on different levels and propagate features directly to the high-quality image reconstruction, adding to the stability and faster convergence of the model.

SwinV2$^+$ transformer. SwinV2 transformer proposed in Liu et al. [34] builds upon the original Swin transformer [15] by altering the shifted window self-attention mechanism to enhance the model's capacity and window resolution. The modified attention module in SwinV2 uses *scaled cosine attention* between keys and queries. This helps in reducing the influence of attention heads on few pixel pairs in features that was present when using *dot product* between keys and queries. Another noticeable change in SwinV2 transformers is the use of post-normalization in place of pre-normalization. Post-normalization decreases the average feature variance in deeper layers which in turn increases the numerical stability of the model during training. SwinV2 transformer further enhances the Swin transformer by using a *log-spaced continuous* relative position bias which helps the model to generalize to higher input resolutions.

However, the self-attention module in these transformers predominantly acts as a low-pass filter [35], which erases the high-frequency information which makes the model lose its expressiveness at deeper layers. Using the methods proposed in Wang et al. [35], we modify the self-attention module of SwinV2 transformer by adding *AttnScale*. In AttnScale, the self-attention matrix is decomposed into two sub-matrices: a high-pass filter and a low-pass filter, and a trainable parameter is introduced to rescale the high-pass filter to match the magnitude of the low-pass filter. This makes the self-attention module behave as an all-pass filter which helps in preserving more high-frequency information. In this work, we call this modified transformer as SwinV2$^+$ transformer. Let A be the attention scores of SwinV2$^+$ transformer, and the operation performed by AttnScale can be represented as shown in Equation (5):

$$\begin{aligned} A_{LP} &= \frac{1}{n} 1 1^T, \\ A_{HP} &= A - A_{LP}, \\ \hat{\mathrm{A}} &= A_{LP} + (\lambda + 1) * A_{HP}, \end{aligned} \tag{5}$$

where λ is the rescaling parameter and 11^T is the largest possible low-pass filter. Subtracting the largest possible low-pass filter A_{LP} from the attention scores A gives the high-pass component A_{HP} of A. The rescaling parameter λ is initialized to zero and is jointly tuned by other network parameters during training. Computing *AttnScale* is relatively lightweight and adds minimal parameter overhead to the window attention module.

3.3. DSFB

The design of our novel DSFB is shown in Figure 2(b). Each DSFB consists of two branches, one that processes information in the Fourier domain and the other in the image domain. The input $F_{i,M} \in R^{H\times W\times C}$ is split into two halves along the channels. The first half is passed to the first branch, while the second half is passed to the second branch.

3.3.1. Processing in Fourier domain

The first branch of DSFB takes the input $F_{i,M} \in R^{H\times W\times \frac{C}{2}}$, where $F_{i,M}$ is the feature maps of the last SwinV2$^+$ transformer layer in the i-th FSTB. Inspired by the success of FFCs in image inpainting tasks [32], we apply the FFC in image super-resolution. The features $F_{i,M}$ are passed through an FFC layer, succeeded by batch normalization and ReLU activation. The FFC operation increases the receptive field and allows the model to preserve the global context better. The FFC layer applies a fast Fourier transform (FFT) along the channels, converting the feature maps from the spatial domain to the Fourier domain. The feature maps in the Fourier domain comprise of both low and high frequencies present in the spatial feature maps. The FFC splits the channels into two branches: a local branch that uses traditional convolutions to extract information in the spatial domain and a global branch that uses real FFT to account for the global context. The FFC operation is comprised of the following steps: Let $F_{in} \in R^{H\times W\times C}$ be the input to FFC layer. Real FFT2d is applied to F_{in} which converts as shown in Equation (6):

$$F_{in} = Real\ FFT2d(F_{in})$$
$$F_{in} \in \mathbb{R}^{H\times W\times C} \rightarrow F_{in} \in \mathbb{C}^{H\times \frac{W}{2}\times C}, \tag{6}$$

Since convolution can operate on real values, FFC converts the above complex tensor to real by concatenating the real and imaginary parts. This conversion converts as shown in Equation (7):

$$F_{in} \in \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow F_{in} \in \mathbb{R}^{H\times \frac{W}{2}\times 2C}, \tag{7}$$

A 1×1 convolution followed by batch normalization and ReLU activation is applied on the above feature maps. This operation operates on feature maps that are represented in the Fourier domain. FFC layer then applies Inverse Real FFT2d to convert the feature maps back to the spatial domain and can be represented as shown in Equation (8):

$$F_{in} \in \mathbb{R}^{H\times \frac{W}{2}\times 2C} \rightarrow F_{in} \in \mathbb{C}^{H\times \frac{W}{2}\times C}$$
$$F_{out} = Inverse\ \ FFT2d(F_{in}) \tag{8}$$
$$F_{in} \in \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow F_{out} \in \mathbb{R}^{H\times W\times C}$$

Results in Suvorov et al. [32] show that FFC works very well on images that have repeated structures and patterns and preserve the global context while being efficient during training and inference.

3.3.2. Processing in the spatial domain

The second branch of DSFB takes in input $F_{i,M} \in R^{H\times W\times \frac{C}{2}}$, where $F_{i,M}$ is the feature maps of the last SwinV2$^+$ transformer layer in the i-th FSTB. This branch extracts information in the spatial domain. The feature maps are passed through a pooling layer that extracts the average intensities in the input. The features are then passed through a series of weight-shared ARFBs [16], which adaptively select and scale features, helping the model to propagate high-frequency details. A skip connection is added after the ARFB layers to stabilize training and improve gradient flow. A 1×1 convolution is then applied to the features for additional processing.

3.3.3. Merging both branches using stereo cross attention module (SCAM)

We use SCAM [36] to merge both the branches into a single branch. SCAM allows the model to selectively combine features from two branches using the attention mechanism. The output of SCAM is then concatenated along channel dimension to output a single branch.

4. Results and Discussions

4.1. Datasets and evaluation metrics

In this work, we use the DIV2K dataset [37] for training. The DIV2K dataset comprises of 800 HR images, and the corresponding LR images are generated through bicubic downsampling from the HR images. To evaluate the performance of the model, we use five widely used benchmark datasets: Set5 [38], Set14 [39], BSD100 [40], Urban100 [41], and Manga109 [42]. We use PSNR and SSIM metrics to quantitatively evaluate the performance of the SR images.

4.2. Experimental setup

4.2.1. Training setting

We randomly crop LR images of size 64×64 as inputs to the model. Data augmentations, like random rotations, random vertical and horizontal flips, and RGB channel shuffling, are used to increase the training data. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and the $L1$ loss function for training. The initial learning rate is set to $2e-4$, and it is reduced by 50% at [384000, 534000, 584000, 609000] iterations. Model was trained for a total of 700,000 iterations. We use PyTorch for building models, and all training is performed using one NVIDIA Tesla A100 GPU.

4.2.2. Implementation details

In SWIFT, we set the number of FSTB to be 4. Inside each FSTB, we use 2 SwinV2$^+$ transformer layers and 2 DSFB layers. Model channel count is set to 64. We use a window size of 8, and number of attention heads is set to 8 in each SwinV2$^+$ transformer layer. In DSFB, we set the number of shared ARFB to 5. The convolution layer at the end of each FSTB is implemented as three 3×3 convolutions followed by PReLU activation.

4.3. Results on lightweight image SR

In this work, we focus primarily on the task of lightweight image super-resolution. Table 1 shows the quantitative comparisons between the proposed SWIFT and eight well-known methods such as CARN [19], IMDN [20], ESRT [16], PAN [23], LAPAR [43], LatticeNet [44], and SwinIR [14]. In keeping comparisons fair, the models selected have model sizes similar to that of our proposed method and the models have been trained using only the DIV2K training set.

From the table, it can be seen that SWIFT outperforms other famous state-of-the-art methods like SwinIR, LatticeNet, and LAPAR. SWIFT achieves higher scores in both PSNR and SSIM metrics on most benchmarking datasets while being the smallest model in terms of model parameters. SWIFT uses 33.55% fewer parameters compared to the previous best SwinIR. The proposed SWIFT architecture allows models to be smaller and run faster during inference. Methods like PAN have much lesser parameters

Table 1
Quantitative comparison of state-of-the-art methods in lightweight image SR on popular benchmarking datasets. Red and blue indicate the top two results

			Set 5		Set14		BSD100		Urban100		Manga109	
Scale	Method	#Param($\times 10^3$)	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
×2	CARN	1,592	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
	IMDN	694	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
	ESRT	677	38.03	0.9600	33.75	0.9184	32.25	0.9001	32.58	0.9318	39.12	0.9774
	PAN	261	38.00	0.9605	33.59	0.9181	32.18	0.8997	32.01	0.9273	38.70	0.9773
	LAPAR	548	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
	LatticeNet	756	38.15	0.9610	33.78	0.9193	32.25	0.9005	32.43	0.9302	–	–
	SwinIR	878	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
	SWIFT (ours)	**579**	38.16	0.9614	33.88	0.9207	32.29	0.9012	32.60	0.9328	39.15	0.9784
×3	CARN	1,592	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
	IMDN	703	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
	ESRT	770	34.42	0.9268	30.43	0.8433	29.15	0.8063	28.46	0.8574	33.95	0.9455
	PAN	261	34.40	0.9271	30.36	0.8423	29.11	0.8050	28.11	0.8511	33.61	0.9448
	LAPAR	594	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
	LatticeNet	765	34.53	0.9281	30.39	0.8424	29.15	0.8059	28.33	0.8538	–	–
	SwinIR	886	34.62	0.9289	30.54	0.8463	29.20	0.8082	28.66	0.8624	33.98	0.9478
	SWIFT (ours)	**600**	34.55	0.9288	30.57	0.8464	29.21	0.8082	28.61	0.8612	34.18	0.9483
×4	CARN	1,592	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
	IMDN	715	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
	ESRT	751	32.19	0.8947	28.69	0.7833	27.69	0.7379	26.39	0.7962	30.75	0.9100
	PAN	272	32.13	0.8948	28.61	0.7822	27.59	0.7363	26.11	0.7854	30.51	0.9095
	LAPAR	659	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
	LatticeNet	777	32.30	0.8962	28.68	0.7830	27.62	0.7367	26.25	0.7873	–	–
	SwinIR	897	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
	SWIFT (ours)	**596**	32.39	0.8978	28.82	0.7870	27.71	0.7411	26.52	0.7992	31.06	0.9153

Figure 3
Qualitative comparisons of various methods in lightweight image SR on Set14 and Urban100 datasets for ×4 scale

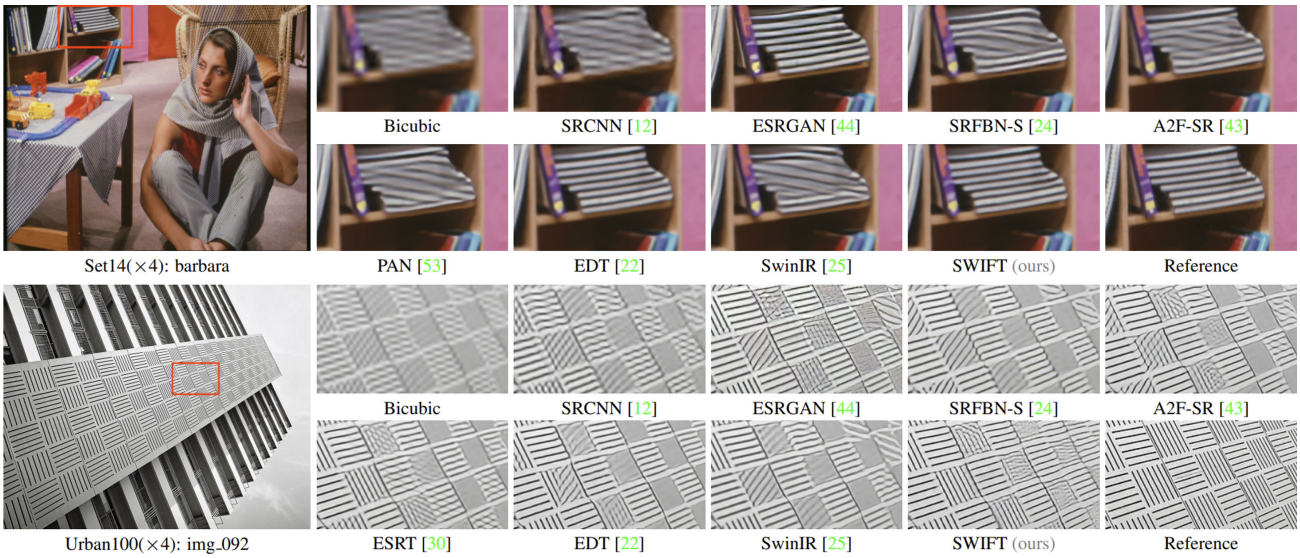


Table 2
Impact of increasing channel count on Set5 (×4) scores

Model	#Channels	#Param (×10^3)	Manga109	
			PSNR↑	SSIM↑
SWIFT	32	187	30.57	0.9089
	64	596	31.06	0.9153
	128	2,218	31.54	0.9221

compared to SWIFT; however, the consequence of reducing parameters is that these methods tend to score less in both PSNR and SSIM metrics. PAN scores ~0.6 dB less than SWIFT in PSNR on Urban100 (×2). The maximum PSNR gain achieved by SWIFT over the previous best SwinIR method is ~0.20 dB in Manga109 for ×3 scale.

In Figure 3, we provide the visual comparisons between SWIFT and other lightweight models on ×4 scale. It is evident from the output of SWIFT that our proposed architecture effectively extracts high-quality features from LR images and uses them to reconstruct SR images with good details.

The quality of reconstructed images produced by SWIFT is on par with that of EDT [21] output. EDT uses additional datasets and pretraining on ImageNet [45] (consists of ~1.3 M images) to obtain higher PSNR and SSIM scores. SWIFT uses only 800 training images from DIV2K for training and achieves similar results qualitatively.

4.4. Ablation studies and discussions

For the ablation study, we train SWIFT on DIV2K [37] datasets and evaluate how each component of the model affects the performance of SR images.

4.4.1. Impact of channel number

Table 2 shows the number of channels used for model training directly impacts the scores obtained by the model. Most lightweight models, like ESRT [16], LAPAR [43], use 32 channels. This decreases the number of parameters in the model, but it sacrifices on both PSNR and SSIM scores. Increasing the channels to 64 increases the parameters drastically and will make the model no longer applicable to lightweight SR. Keeping this in mind, we design SWIFT to work with 64 channels and have fewer parameters than most lightweight models that use 32 channels. Increasing channels further increases the performance of the model but this extra performance diminishes gradually.

4.4.2. Impact of the number of FSTB, SwinV2$^+$ and DSFB

Results in Figure 4 show a noticeable positive correlation between the number of FSTB in the model and the PSNR scores obtained. However, using more FSTB in the model increases both parameters and inference time. For the purpose of lightweight SR, SWIFT uses 4 FSTB blocks. The number of SwinV2$^+$ transformer layers and DSFB inside FSTB also tends to affect PSNR scores. In our experiments, an equal number of SwinV2$^+$ and DSFB tend to perform better than other configurations. More specifically, for lightweight SR, we set the number of SwinV2$^+$ and DSFB to be 2 and 2, respectively. To have a smaller model, we found that using 4 FSTB each with 2 SwinV2$^+$ and 2 DSFB gave the model enough capacity to learn better while staying low on parameters.

4.4.3. Impact of datasets and training patch size

CNN-based models often tend to use DIV2K [37] for image SR. Recently, several methods have been proposed that train models on more images by combining images from DIV2K [37] and Flickr2K [46] datasets called the DF2K dataset. As there are more training examples, it is expected that models trained on DF2K dataset achieve better scores in PSNR and SSIM metrics as shown in Table 3. However, to keep a fair model comparison, we train SWIFT only on the DIV2K dataset. Models like SwinFIR [18] and IPT [30] pre-train the models on ImageNet [45] dataset and then fine-tune it for DIV2K and DF2K datasets. These models, although they outperform most other models, tend to require a lot

Figure 4
Impact of the number of FSTB layers in SWIFT on Set5 (×2) scores

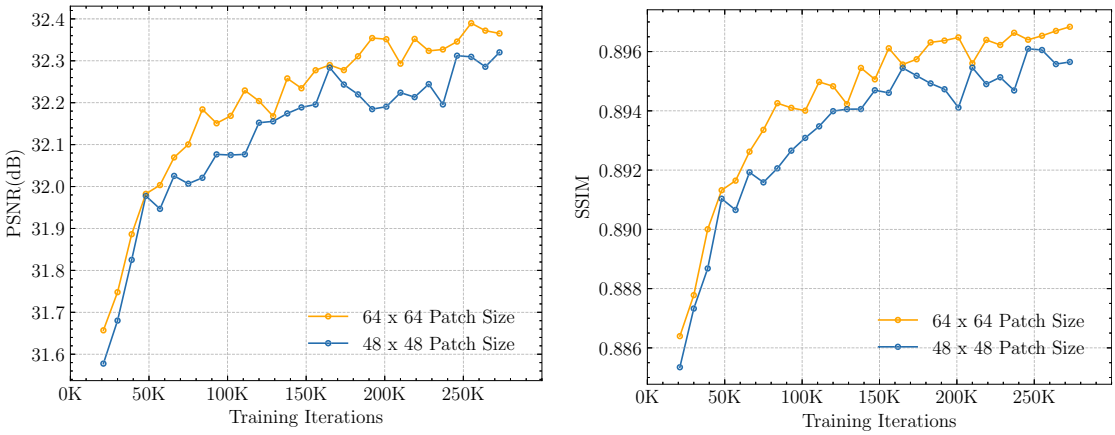


Table 3
Impact of different training datasets on Set5 (×4) scores

Model	Parameters	Dataset	Set5	
			PSNR ↑	SSIM ↑
SWIFT	596K	DIV2K	32.39	0.8978
		DIV2K+Flickr2K	32.43	0.8983

of resources for training. With respect to the training patch size, as shown in Figure 5, training on larger patch size (64 × 64 instead of traditional 48 × 48) tends to allow the model to converge better and perform better in both PSNR and SSIM scores.

4.4.4. Study of components in DSFB

DSFB is an important component of SWIFT as it is capable of extracting high-frequency information during feature extraction. Table 4 shows the contribution of each component in DSFB. In the table, ARFB and FFC indicate the spatial branch and Fourier branch, respectively. According to Cases 1 and 2, we observe that the model scores less in PSNR scores when only one of the two branches in DSFB is used. Case 3 shows that model performs better in PSNR scores when both branches are used together for feature extraction. Case 4 shows that using SCAM [36] in DSFB gives a gain of ~0.05 dB in PSNR scores as it helps to combine features from both branches effectively. Although SCAM is traditionally used in stereo image super-resolution, using SCAM in our architecture helps SWIFT to score higher in both PSNR and SSIM metrics.

4.5. Comparisons with latest SwinIR advancements

Since the introduction of SwinIR [14], several improvements have been proposed, namely SwinFIR [18] and Swin2SR [47]. SwinFIR proposes SFBs that use Fourier transformation to capture global context. The DSFB in SWIFT uses FFCs [17] to extract high-frequency information while maintaining global context in feature maps. The SFB in SwinFIR just applies Fourier transformation and a few convolution operations to extract features. The comparison between SWIFT and SwinFIR is shown in Table 5. SwinFIR uses advanced techniques such as feature ensembles, ImageNet pretraining, and fine-tuning on the DIV2K dataset. Comparing SWIFT and SwinFIR is not fair as SwinFIR has been pre-trained on large datasets, and the lack

Figure 5
Impact of training patch size on Set5 (×2) scores

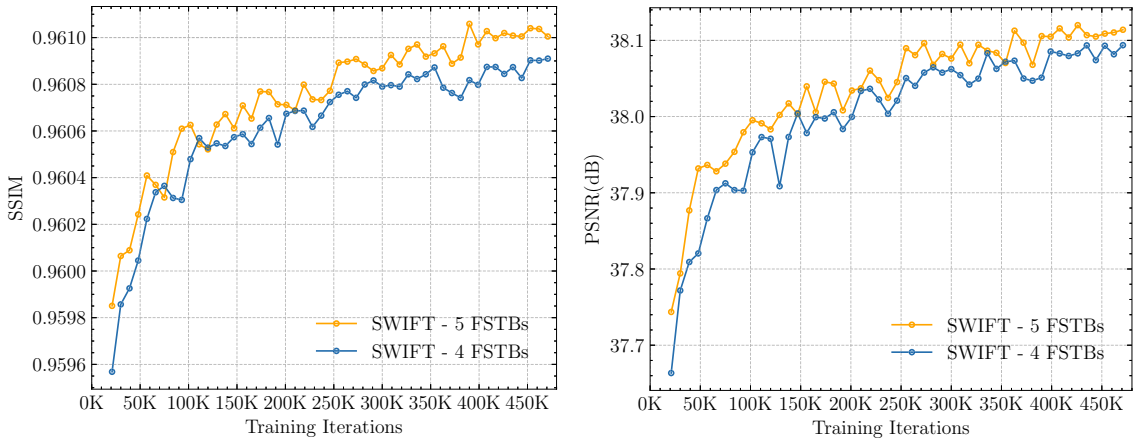


Table 4
Study of each component in DSFB on Set5 (×2)

Case Index	1	2	3	4
ARFB		✓	✓	✓
FFC	✓		✓	✓
SCAM				✓
Parameters	462K	482K	544K	579K
PSNR	38.09 dB	38.08 dB	38.11 dB	**38.16 dB**

Table 5
Comparison of SWIFT with advances in SwinIR architecture for ×2 scale

Model	#Param ($\times 10^3$)	BSD100		Urban100	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
SwinFIR-T	872	32.38	0.9024	33.14	0.9374
Swin2SR	1000	32.35	0.9024	32.85	0.9349
SWIFT (ours)	579	32.29	0.9012	32.60	0.9328

of a public codebase for SwinFIR inhibits us from performing comparisons on inference time and model size.

Swin2SR improves over SwinIR by using SwinV2 transformers while keeping the overall architecture similar to SwinIR. Similar to SwinFIR, Swin2SR uses additional datasets such as a combination of DIV2K and Flickr2K [46] for training. SWIFT uses 800 training images from DIV2K dataset and achieves scores closer to Swin2SR. Swin2SR uses ∼68% more parameters compared to SWIFT but gains ∼0.06 dB in PSNR score for BSD100 [40] dataset and takes 84% longer during inference (Table 6).

4.6. Comparisons on computational costs

A detailed comparison of inference times for state-of-the-art methods has been shown in Table 6. From the table, it is evident that SWIFT has the lowest inference time across all the benchmarking datasets compared to other transformer-based methods like SwinIR [14], ESRT [16], and Swin2SR [47]. SWIFT consistently achieves fast inference time for images of smaller size as present in benchmarking datasets like Set5 [38], Set14 [39], and BSD100 [40].

However, in DIV2K Validation [37], Urban100 [41], and Manga109 [42], where the images are comparatively larger, inference using SWIFT is much faster compared to other transformer-based methods. More specifically, SWIFT is ∼53.3% faster on average in inference time compared to the SwinIR method. Inference can also be carried out by recursively dividing images into smaller patches and stitching the model predictions on individual patches to reconstruct the SR image. This method is useful for models like ESRT [16], which are efficient for small input images (64 × 64) but are computationally expensive to run when the size of input images are large (256 × 256). Our method easily scales to larger image sizes and can run efficiently on both types of inference methods. To compare various models on model size and computational complexities, we report the total number of parameters and multiply-accumulate operations, evaluated on an input of size 1280 × 720 for ×4 scale.

Table 6
Comparison of inference time of popular methods on benchmarking datasets for ×4 scale. The ▼ symbol indicates improvement and ▲ symbol indicates deterioration of inference time compared to the reference model. The reference model used for comparisons is indicated by * (asterisks). Inference for all models was carried out on NVIDIA RTX 3080 GPU

Method	Architecture type	#Param	Multi-Adds	Inference time (ms) ↓				
				DIV2K Val	Set5	Set14	BSD100	Urban100	Manga109
Transformer-Based Methods									
SWIFT* (ours)	SwinV2$^+$ Transformer + FFC	**596K**	**49.2G**	**256.48**	**23.62**	**28.47**	**24.14**	**67.77**	**77.07**
ESRT	CNN + Efficient Transformer	751K▲26%	67.7G	457.31▲44%	64.84▲174%	66.60▲134%	63.64▲164%	129.77▲92%	166.34▲116%
SwinIR	Swin Transformer	897K▲50%	49.6G	632.80▲147%	25.93▲10%	48.30▲70%	38.21▲58%	149.11▲120%	178.56▲132%
Swin2SR	SwinV2 Transformer	1,000K▲68%	50.5G	694.68▲171%	28.61▲21%	53.97▲90%	44.33▲84%	166.97▲146%	200.19▲160%
Convolution-Based Methods (with Hardware Support)									
CARN	CNN	1,592K	90.9G	55.24	3.50	7.09	4.23	21.46	25.20
IMDN	CNN	715K	40.9G	27.27	4.19	4.77	3.74	10.58	14.18

5. Limitations

Although SWIFT is designed to be smaller and capable of running faster at inference compared to other transformer-based methods, CNN-based methods like CARN [19] and IMDN [20] still run much faster than SWIFT. This difference in speed is mainly because CNN-based methods can leverage hardware acceleration on modern GPUs. The presence of optimized convolution kernels in hardware allows for significantly faster convolution operations. While CNN-based models may have faster inference times, their performance is limited by their reliance on local receptive fields. They lack the ability to effectively model long-range dependencies and struggle to preserve fine textures compared to transformer-based methods like SwinIR [14] and ESRT [16]. This limitation is evident from the qualitative results presented in Figure 3. To address the challenge of long-range dependencies, SWIFT utilizes FFC instead of traditional convolutions. By incorporating Fourier convolutions, SWIFT mitigates the limitations of local receptive fields and enables the modeling of global context. Additionally, SWIFT leverages the SwinV2$^+$ transformer, which enhances its ability to map textures effectively. With the increasing popularity of transformer-based architectures, modern accelerators such as NVIDIA H100 provide hardware support for running transformers. Presence of such hardware support can provide inference closer to CNN-based methods.

6. Conclusion

In this work, we introduce SWIFT, a hybrid model consisting of transformers and FFCs, for lightweight single-image super-resolution. We design a new block called DSFB that extracts high-frequency information by processing features in both the spatial domain and the Fourier domain. DSFB effectively models the long-range dependencies in feature maps while staying low on parameters. Additionally, we also extend transformers to promote high-frequency information with the usage of SwinV2$^+$ transformers. Experimental results show that SWIFT achieves state-of-the-art results on popular benchmarking datasets in lightweight SISR. SWIFT architecture allows models to be smaller in terms of a total number of model parameters and offers faster inference compared to other transformer-based lightweight image super-resolution methods.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Vishal Ramesha: Conceptualization, Methodology, Software, Formal analysis, Resources, Data curation, Writing – original draft, Writing – review & editing. **Yashas Kadambi:** Conceptualization, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **B. S. Abhishek Aditya:** Methodology, Software, Investigation, Data curation, Visualization. **T. Vijay Prashant:** Software, Formal analysis, Data curation, Visualization. **S. S. Shylaja:** Conceptualization, Validation, Resources, Writing – review & editing, Supervision, Project administration.

References

[1] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. In *35th Conference on Neural Information Processing Systems*.

[2] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., . . . , & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4681–4690.

[3] Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., & Wu, W. (2019). Feedback network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3867–3876.

[4] Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 136–144.

[5] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., . . . , & Change Loy, C. (2018). ESRGAN: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision Workshops*.

[6] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision*, 286–301.

[7] Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, *16*(8), 2080–2095. https://doi.org/10.1109/TIP.2007.901238

[8] Gu, S., Sang, N., & Ma, F. (2012). Fast image super resolution via local regression. In *Proceedings of the 21st International Conference on Pattern Recognition*, 3128–3131.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . , & Polosukhin, I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems*.

[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . , & Houlsby, N. (2020). An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv Preprint: 2010.11929*.

[11] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6836–6846.

[12] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer Vision-ECCV 2020: Proceedings of 16th European Conference*, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[13] Esser, P., Rombach, R., & Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12868–12878. https://doi.org/10.1109/CVPR46437.2021.01268

[14] Liang, J., Cao, J., Sun, G., Zhang, K., van Gool, L., & Timofte, R. (2021). SwinIR: Image restoration using Swin transformer.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 1833–1844.

[15] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . , & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

[16] Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., & Zeng, T. (2022). Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 457–466.

[17] Chi, L., Jiang, B., & Mu, Y. (2020). Fast Fourier convolution. In *34th Conference on Neural Information Processing Systems*.

[18] Zhang, D., Huang, F., Liu, S., Wang, X., & Jin, Z. (2022). SwinFIR: Revisiting the SwinIR with fast Fourier convolution and improved training for image super-resolution. *arXiv Preprint:2208.11247*. https://doi.org/10.48550/arXiv.2208.11247

[19] Ahn, N., Kang, B., & Sohn, K. A. (2018). Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision*, 252–268.

[20] Hui, Z., Gao, X., Yang, Y., & Wang, X. (2019). Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2024–2032. https://doi.org/10.1145/3343031.3351084

[21] Li, W., Lu, X., Qian, S., & Lu, J. (2023). On efficient transformer-based image pre-training for low-level vision. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 1089–1097. https://doi.org/10.24963/ijcai.2023/121

[22] Wang, X., Wang, Q., Zhao, Y., Yan, J., Fan, L., & Chen, L. (2020). Lightweight single-image super-resolution network with attentive auxiliary feature learning. In *Proceedings of the Asian Conference on Computer Vision*.

[23] Zhao, H., Kong, X., He, J., Qiao, Y., & Dong, C. (2020). Efficient image super-resolution using pixel attention. In *Computer Vision–ECCV 2020 Workshops*, 56–72.

[24] Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. https://doi.org/10.1109/TPAMI.2015.2439281

[25] Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1646–1654.

[26] Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., . . . , & Shen, H. (2020). Single image super-resolution via a holistic attention network. In *Computer Vision–ECCV 2020: 16th European Conference*, 191–207.

[27] Dai, T., Cai, J., Zhang, Y., Xia, S. T., & Zhang, L. (2019). Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11065–11074.

[28] Mei, Y., Fan, Y., Zhou, Y., Huang, L., Huang, T. S., & Shi, H. (2020). Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5690–5699.

[29] Mei, Y., Fan, Y., & Zhou, Y. (2021). Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3517–3526.

[30] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., . . . , & Gao, W. (2021). Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12299–12310.

[31] Chen, X., Wang, X., Zhou, J., Qiao, Y., & Dong, C. (2023). Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22367–22377.

[32] Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., . . . , & Lempitsky, V. (2022). Resolution-robust large mask inpainting with Fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2149–2159.

[33] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., . . . , & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1874–1883.

[34] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., . . . , & Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12009–12019.

[35] Wang, P., Zheng, W., Chen, T., & Wang, Z. (2022). Anti-oversmoothing in deep vision transformers via the Fourier domain analysis: From theory to practice. *arXiv Preprint: 2203.05962*. https://doi.org/10.48550/arXiv.2203.05962

[36] Chu, X., Chen, L., & Yu, W. (2022). NAFSSR: Stereo image super-resolution using NAFNet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1239–1248.

[37] Timofte, R., Agustsson, E., van Gool, L., Yang, M. H., & Zhang, L. (2017). NTIRE 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 114–125.

[38] Bevilacqua, M., Roumy, A., Guillemot, C., & Alberi-Morel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the 23rd British Machine Vision Conference*, 1–10.

[39] Zeyde, R., Elad, M., & Protter, M. (2012). On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference*, 711–730. https://doi.org/10.1007/978-3-642-27413-8_47

[40] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, *2*, 416–423. https://doi.org/10.1109/ICCV.2001.937655

[41] Huang, J. B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5197–5206.

[42] Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., & Aizawa, K. (2017). Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, *76*(20), 21811–21838. https://doi.org/10.1007/s11042-016-4020-z

[43] Li, W., Zhou, K., Qi, L., Jiang, N., Lu, J., & Jia, J. (2020). LAPAR: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. In *34th Conference on Neural Information Processing Systems*.

[44] Luo, X., Xie, Y., Zhang, Y., Qu, Y., Li, C., & Fu, Y. (2020). LatticeNet: Towards lightweight image super-resolution with lattice block. In *Computer Vision–ECCV 2020: 16th European Conference*, 272–289. https://doi.org/10.1007/978-3-030-58542-6_17

[45] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F. F. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[46] Wang, Y., Wang, L., Yang, J., An, W., & Guo, Y. (2019). Flickr1024: A large-scale dataset for stereo image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[47] Conde, M. V., Choi, U. J., Burchi, M., & Timofte, R. (2023). Swin2SR: Swinv2 transformer for compressed image super-resolution and restoration. In *Proceedings of Computer Vision-ECCV 2022 Workshops*, 669–687. https://doi.org/10.1007/978-3-031-25063-7_42

How to Cite: Ramesha, V., Kadambi, Y., Abhishek Aditya, B. S., Prashant, T. V., & Shylaja, S. S. (2025). Toward Faster and Efficient Lightweight Image Super-Resolution Using Transformers and Fourier Convolutions. *Artificial Intelligence and Applications*, *3*(2), 168–178. https://doi.org/10.47852/bonviewAIA42021930