Noname manuscript No.
(will be inserted by the editor)

XTNSR: Xception-Based Transformer Network for Single Image Super Resolution

Jagrati Talreja[1] · Supavadee Aramvith[2] · Takao Onoye[3]

Received: date / Accepted: date

Abstract Single Image Super Resolution has significantly advanced by utilizing Transformers-based Deep Learning algorithms. However, challenges still need to be addressed in handling grid-like image patches with higher computational demands and addressing issues like over-smoothing in visual patches. This paper presents a Deep Learning model for single-image super-resolution. In this paper, we present the XTNSR model, a novel multi-path network architecture that combines Local Feature Window Transformers (LWFT) with Xception blocks for single-image super-resolution. The model processes grid-like image patches effectively and reduces computational complexity by integrating a Patch Embedding layer. Whereas the Xception blocks use depthwise separable convolutions for hierarchical feature extraction, the LWFT blocks capture long-range dependencies and fine-grained qualities. A multi-layer feature fusion block with skip connections, part of this hybrid architecture, guarantees efficient local and global feature fusion. The experimental results show better performance in Peak signal-to-noise ratio (PSNR), Structural Similarity Index Measure (SSIM), and visual quality than the state-of-the-art techniques. By optimizing parameters, the suggested architecture also lowers computational complexity. Overall, the architecture presents a promising approach for advancing image Super-Resolution capabilities.

Jagrati Talreja[1]
Department of Electrical Engineering,
Faculty of Engineering, Chulalongkorn University,
Bangkok, 10330, Thailand.

Supavadee Aramvith[2]
Multimedia Data Analytics and Processing Unit,
Department of Electrical Engineering,
Faculty of Engineering, Chulalongkorn University,
Bangkok 10330, Thailand.
E-mail: supavadee.a@chula.ac.th

Takao Onoye[3]
Graduate School of Information Science and Technology,
Osaka University,
1-5 Yamada-Oka, Suita, 565-0871 Japan.
E-mail: onoye@ist.osaka-u.ac.jp

Keywords Single Image super-resolution · Local Feature Window Transformer Block · Multi-Layer Feature Fusion Block · Xception Block

1 Introduction

Image super-resolution (ISR) is a fundamental challenge that aims to recover high-resolution details from lower-resolution images. Single-image super-resolution (SISR) is an ill-posed problem because of the generation of multiple High-Resolution (HR) images for every Low-Resolution (LR) image. Many architectures have contributed significantly to Super- Resolution in the Deep Learning panorama, each with advantages. The need for higher image resolution is motivated by its many applications, such as medical image analysis [1], forensics [2], and astronomical imagery [3]. In recent years, deep learning techniques have emerged as powerful tools for ISR, with Convolutional Neural Networks (CNNs) and Transformer-based architectures showing great promise in improving image quality. With varying applicability in image segmentation [4], reconstruction [5], estimation [6], anomaly detection [7], etc., these architectures recover the finer detail of any image.

Convolutional Neural Networks (CNNs) have led the way regarding image-related tasks. Dong et al. [8] were the first to propose a three-layer (extraction, mapping, and reconstruction) shallow convolutional neural

Fig. 1: Research Trends followed by Deep Learning based Image Super-Resolution Methods.

network-based architecture named Super-Resolution Convolution Neural Networks (SRCNN). Other models followed the SRCNN [8] model, like Accelerating the super-resolution convolutional neural network (FSRCNN) [9] to improve training efficiency. Deeper and more progressive networks were also introduced for accurate image super-resolution [10, 11, 12, 13, 14, 15, 16, 17] to improve the visual quality of the HR image, increase the model's capacity, reduce the computational complexity, and enhance the optimization strategy. However, the fundamental issues with conventional CNNs' ability to capture global contextual information and long-range dependencies led to exploring alternative architectures. As seen in Figure 1, the research trends shifted from CNNs to GANs, attention mechanisms, and transformer networks. To obtain more visually pleasing results, the researchers explored the direction of Generative Adversarial Networks (GANs) [18; 19; 20]. While Generative Adversarial Networks (GANs) are incredibly powerful for image generation tasks, their direct application to image super-resolution can sometimes pose challenges, especially regarding training stability and balancing the trade-off between generating realistic high-resolution images and avoiding model collapse. Due to this, the research moved to introduce attention mechanisms in CNNs' which can help the model focus on relevant image regions and capture complex spatial dependencies. Many recent models such as Image super-resolution using very deep residual channel attention networks (RCAN) [21], Image Super-Resolution with Cross-Scale Non-Local Attention and Exhaustive Self-Exemplars Mining (CSNL) [22], Image Super-Resolution with non-local sparse attention (NLSN) [23], Single image super-resolution via a Holistic Attention Networks (HAN) [24] and Efficient long-range network for image super-resolution (ELAN) [25] etc. have shown remarkable performance gains with the addition of different attention mechanisms in CNNs' making it possible to capture fine-grained detail and improve feature extraction. Nevertheless, these models proved to need to be improved when handling grid-like image data. Transformer stands out by leveraging attention mechanisms to capture better global dependencies and handle grid-like image data. The most recent models for demonstrating the efficient use of transformers in ISR are Image restoration using a Swin transformer (SwinIR) [26], Hierarchical Vision Transformer using Shifted Windows (Swin Transformer) [27], and Permuted Self-Attention for Single Image Super-Resolution (SRFormer) [28]. Even though they are cost-effective, there is still room to improve the balance between local and global context and inference time.

In the context of built-in systems and intelligent hardware with immediate handling, the model's size is crucial for attaining faster and more advanced results, particularly when dealing with higher scale factors in image super-resolution. Despite the advancements in state-of-the-art quantitative and qualitative methods, they still need to be improved.

(i) Images with high-resolution grid-like patches can be computationally demanding, resulting in higher memory resource needs and longer inference time.

(ii) Balancing local and global context while addressing potential issues such as over-smoothing and artifact generation.

(iii) When dealing with noisy low-resolution images, earlier approaches may introduce rugged patterns and uneven edges and may not effectively recover fine details.

An effective solution to all these issues is to combine the complementary abilities of Transformer [29] and Xception blocks [30] in a multi-path network [31] using skip connections [32]. This approach combines various types of extracted features from each path, enhancing both the output's perceptual quality and the SR model's operation time. It suggests that the network's layers don't need to wait for the results of earlier layers' computations and helps to reduce inference time. The multi-path network was introduced by Kim et al. in DRCN [13] to extract different types of features and combine them for performance enhancement and reduce parametric complexity.

Making use of a similar approach, we introduce a new Local Feature Window Transformer Block (LFWT) and a Multi-Layer Feature Fusion (MLFF) Block to effectively capture spatial relational features and combine complementary abilities of the Transformer with the Xception block in a multi-path network backbone. Patch embedding helps to address the challenge of processing grid-like image data by transforming input images into smaller patches, reducing the model's complexity. Using Local Feature Window Transformer (LFWT) Blocks in a multi-path framework helps strike a balance between the local and global context to address the issue of over-smoothing. Integration of the Xception Block via skip connections in a Multi-Layer Feature Fusion (MLFF) Block increases the model's ability to learn hierarchical features. It helps finely recover images with noise and improves training stability.

To summarize, the following list outlines the main contributions of our proposed method:

(1) Patch Embedding layer addresses scalability challenges, balancing computational efficiency and accuracy in adapting to varied computational resources.

(2) The method extracts features using LFWT Block. It processes them in a multi-path network, enabling the model to capture and balance local and global features to mitigate over-smoothing and artifact generation.

(3) Extracting hierarchical features from the Xception Block and integrating them in the MLFF Block helps recover fine details from noisy images, thus reducing noise.

The remaining sections of the article are organized as follows. Section II states the related work, Section III presents the proposed network architecture methodology, and Section IV demonstrates the experimental evaluations, ablation analysis, and visual results on experimental datasets. Finally, Section V gives a conclusion and recommendations for further work.

2 Related Work

In recent years, brilliant success in Single Image Super-Resolution has been driven by the ongoing development of deep learning architectures and techniques. This section summarizes the major contributions made to the field, with an emphasis on models that are similar to our suggested approach, including Xception blocks [30], transformer [29], patch embedding [33, 34], and multi-path networks [31]. For clarity, we categorize the related works into CNN-based approaches, GAN-based approaches, attention mechanisms, and transformer-based approaches.

2.1 CNN-based approaches

The pioneering work in the field of image Super-Resolution using the power of convolutional neural network was done by Dong et al. [8], which introduces a three-layered network for extraction features from low-dimensional space and mapping them to reconstruct higher dimensional space. SRCNN [8] was a remarkable introduction to Deep Learning Image Super-Resolution models. After this, much research was done to improve the deep learning models regarding performance, computational complexity, training efficiency, and stability. The better version of SRCNN [8] was later introduced by Dong et al. to reduce the computational complexity named FSRCNN [9]. While improving the deep learning algorithms, the researchers mainly focused on improving the methods regarding model frameworks, up-sampling methods, network designs, and learning strategies. Even though FSRCNN [9] improved the network's computational complexity, it still had limited capacity. The research shifted to creating deeper and more intricate architectures to improve the model capacity. A 20-layer residual network named VDSR [10] was introduced by Kim et al., and a pyramid structure network LapSRN [12] was proposed by Lai et al. Though the deeper network significantly improved performance, they had high computational costs. By introducing the recursive learning strategy in its network design, Kim et al. improved the deeper network introduced by DRCN [13] and rectified the concept of deeper networks. To further improve the training convergence, DnCNN [11] was introduced by Chen et al. Since the deeper network could not meet the demand for current cutting-edge devices, lightweight models such as IMDN [35] by Hui et al. and DRRN [14] by Tai et al. were introduced. When the progress shifted towards improving residual networks and introducing EDSR [17], substantial gains in performance were made, and the training efficiency of the model was boosted. EDSR [17] won the NTIRE (2017) challenge and became the widely used backbone of current research for creating Deep Learning Image Super-Resolution methods. To fit in the memory resources needed for the algorithms, a persistent memory network for Image Super-Resolution MemNet [15] was introduced by Tai et al. Another feedback network, the Super Resolution Feedback Network (SRFBN) [36], was also introduced to use fewer trainable parameters. To deal with noisy image methods such as SRMDNF [37] performed significantly great. Shallower Networks were also adapted to reduce the computational requirements, such as Super Sampling Network (SSNet) [38] by Hung et al. and Deep Recurrent Fusion Network (DRFN) [39]. Another approach that focused on quantization and compression technique was also introduced by Muhammad et al. in Squeeze-and-Excitation Next for Single Image Super-Resolution (SENext) [40].

However, since 2020, CNN-based approaches have faced challenges in handling global contextual information and long-range dependencies. This led to a shift towards integrating novel techniques such as recursive networks and multi-path architectures like IMDN [35] and DRRN [14], which aimed to optimize memory usage and computational efficiency.

Ozdemir (2024) [41] introduced an innovative approach to adapt transfer learning models to specific datasets through pruning techniques and Avg-TopK pooling. This method significantly reduces the model's complexity while retaining critical information, improving the network's generalization ability across diverse datasets. In addition, research by Ozdemir, Dogan, and Kaya (2024) introduced a new local pooling strategy known as Local Binary Pattern (LBP) pooling, which further refines feature extraction by focusing on local texture information within image patches [42]. This method demonstrates improved results in image processing tasks, especially when integrated into deeper CNN architectures.

2.2 GAN-based approaches

Although all these techniques performed better in quantitative measurements, the visual quality could have been significantly improved. Hence, to improve the visual patches of the images, the research trend shifted to Generative Adversarial Networks (GANS). SRGAN [18], EnhanceNet [19], and ESRGAN [20] are some good examples of GAN networks and have demonstrated significant visual gains in Image Super-Resolution. However, GANs showed great enhancement in the visual performance but were insufficient to generate better PSNR values. Hence, the Mean Opinion Score (MOS) [43], a manual estimate of human raters to measure the performance of the GAN model, was introduced. The research again shifted to the neural network when attention mechanisms were introduced in the Convolutional Neural Networks (CNNs), overcoming the drawbacks of both earlier approaches.

Post-2020, GAN-based methods, such as SPSR-GAN [44], have continued to push boundaries by incorporating perceptual losses and multi-scale discriminators to handle diverse image features. However, GANs are still challenged by stability issues and the need to tune hyperparameters to avoid mode collapse carefully.

2.3 Attention mechanism-based approaches

Attention mechanisms have greatly enhanced super-resolution models for images. Combining attention mechanisms and convolutional operations has demonstrated improved performance. Zhang et al. Residual Dense Network (RDN) [45] showcases the efficacy of integrating skip connections and residual learning for enhanced feature extraction and super-resolution quality. Works like Residual Attention Network (RAN) [46] by Wang et al. and Image Super-Resolution Using Very Deep Residual Channel Attention Networks (RCAN) [21] by Zhang et al. demonstrate the benefits of attention mechanisms when focusing on informative image regions. Since then, many works have been contributed with the use of different types of attention mechanisms, such as Image Super-Resolution with Cross-Scale Non-Local Attention and Exhaustive Self-Exemplars Mining (CSNL) [22], which used non-local attention and combined it with cross-scale features. Another similar work was introduced in Multi-FusNet of Cross Channel Network for Image Super-Resolution (MFCC) [47] and Fast Non-Local Attention network for light super-resolution FNLNET [48]. To deal with more intricate features, a second-order channel attention module was introduced in the Second-order Attention Network for Single Image Super-Resolution (SAN) [49]. Spatial Attention was introduced in the Residual Feature Aggregation Network (RFANet) [50]. Channel Split Image Super-Resolution (CSISR) [51] was introduced to improve learning capability. Another lightweight model to reduce the computational requirements was introduced in the Dynamic Residual Self-Attention Network (DRSAN) [52]. For larger and contextual information, the Context Reasoning Attention Network (CRAN) [53] and Information Growth Attention Network (IGAN) [54] were introduced. To better understand the correlation between layers and features and improve performance, a Holistic Attention Network (HAN) [24] was introduced. Sparsity, along with non-local attention, was also introduced to improve the computation complexity and preserve model capabilities in Non-Local Neural Networks (NLNN) [55] and Image Super-Resolution using Non-Local Sparse Attention Networks (NLSN) [23]. Architectural changes were made using the U-Net framework to non-local design to improve performance further and reduce computational burden in the Deep Attention Network for Single Image Super-Resolution (DANS) [56]. Although attention mechanisms have proven understand the complex contextual and finer details better, improvement is still needed.

Since 2020, there has been a growing emphasis on enhancing these attention mechanisms for SISR tasks. Works such as NLSN [23] and SAN [46] introduced non-local and second-order attention mechanisms, respectively, to better capture complex dependencies in images. These innovations substantially improved both quantitative and qualitative results, addressing challenges like over-smoothing and the loss of fine textures.

2.4 Transformer-based approaches

Patch embedding was first introduced in Attention is All You Need by Vaswani et al. [57], significantly improving performance when applied to attention mechanisms. This concept was introduced from patch embedding in transformers. Transformer-based models have become more popular for tasks involving image super-resolution. In their introduction of the Image Transformer, Dosovitskiy et al. [29] emphasized that transformers could capture long-range image dependencies. However, challenges persist in efficiently processing grid-like image structures, prompting the need for innovative architectural solutions. Since Transformer architectures have shown great success in Natural Language Processing (NLP) [58], their contribution to image super-resolution also achieves greater gain in qualitative and quantitative measures. The pioneering work of Vision Transformer (ViT) [29] has led to many developments in Image Super-Resolution using transformers. After that, much work in image super-resolution has been depicted using transformers. One is Image Restoration Using a Swin Transformer (SwinIR) [26]. It uses a fixed window size of 8 × 8 for extracting features simplified in the Efficient Long-Range Attention Network for Image Super-resolution (ELAN) [25] by different window sizes to improve self-attention in transformers showing better performance gains. Furthermore, improvements are shown in Hierarchical Vision Transformer using Shifted Windows (Swin Transformer) [27], which used the shifted window technique instead of sliding window to make the processing of image patches more compact. Recently, Permuted Self-Attention for Single Image Super-Resolution (SRFormer) [28] has been introduced, which performs self-attention in large window sizes to further improve the performance without increasing the computational cost of the model.

From 2020 onwards, transformer-based models such as SRFormer [28] and ELAN [25] have further refined this approach, introducing hierarchical structures and efficient attention mechanisms in hybrid approaches like DHTCUN [59] to balance computational cost and image quality. Recent methods like CTAFFNet [60] and TS:BEV [61] futher provide more valuable context and academic support on CNN-Transformer fusion techniques. These innovations represent a significant shift in SISR research, as transformers offer flexibility and scalability for high-resolution image reconstruction.

Table 1 presents the literature review in a comprehensive table format. This table summarizes key single-image super-resolution (SISR) models, along with their specialized indicators such as architecture type, key components, strengths, and limitations.

Chollet et al. introduced deep Learning with Depth-wise Separable Convolutions (Xception) [30]. It showed the efficacy of its customary design, which enhanced parameter efficiency and feature extraction capabilities by replacing depth-wise separable convolutions for conventional convolutional layers. The use of Xception for single-image super-resolution was extended by Lim et al. in Enhanced Deep Residual Networks for Single Image Super-Resolution (EDSR) [17]. By incorporating Xception blocks into the network, the study demonstrated enhanced feature extraction, making it easier to collect hierarchical information for reconstructing high-resolution images. As computational complexity decreased, Xception's depth-wise separable convolutions [30] assisted in preserving performance. Xception has demonstrated success in various image-processing tasks combined with other neural network components in hybrid architectures. Zhang et al. in Residual Dense Network (RDN) [45] demonstrated the advantages of combining depth-wise separable convolutions with residual learning for better super-resolution quality.

Significant progress has been made in image super-resolution using transformer-like architectures. Still, there is a need for techniques that can handle grid-like image data effectively while requiring less processing power. Furthermore, searching for the optimal possible balance between local and global information is still necessary to achieve realistic super-resolved results while mitigating general artifacts and over-smoothing. Research is still being conducted to address issues like computational efficiency and cross-domain generalization. Reaching the maximum potential of super-resolution images for a range of uses, such as multimedia [62], live streaming [63], medical [64], security [65], and super-resolution images in real time [66].

3 Mathematical Formulation of Problem Statement

There are various obstacles to achieving high-quality image super-resolution (ISR) with immediate handling in the context of built-in systems and intelligent hardware. More specifically, for higher scale factors, the model's performance and size are critical for faster results. Even with the development of cutting-edge techniques, the following areas still require improvement:

3.1 Computational Demands and Model Size

To maintain short inference times, the model must minimize memory usage and computational complexity. For higher scale factors in ISR, this is especially significant.

Let:

- F represents the total number of model parameters,
- C_F is the computational complexity (in FLOPs),
- M_F is the memory required to store the model,

Table 1: Literature Review of Key SISR Models

Model	Type	Key Components	Strengths	Limitations
SRCNN [8]	CNN-based	3-layer CNN, feature extraction, mapping	Simple architecture, foundational in SISR	Limited depth, struggles with fine details
EDSR [17]	CNN-based	Deep residual blocks, no batch normalization	High performance, NTIRE 2017 winner	Large model size, memory intensive
ESRGAN [20]	GAN-based	Residual-in-Residual Dense Block, GAN-based discriminator	Excellent visual quality, realistic texture generation	Lower PSNR, unstable training
SPSR-GAN [44]	GAN-based	Perceptual loss, multi-scale discriminator	Perceptual quality improvement	Stability issues, sensitive hyperparameter tuning
RCAN [21]	Attention-based	Channel attention, deep residual learning	Better attention on informative regions	Computationally intensive
RDN [45]	Attention-based	Dense residual blocks, skip connections	Enhanced feature extraction	High memory consumption
SwinIR [26]	Transformer-based	Swin Transformer, shifted windows	Efficient patch-based processing, strong performance	Relatively high computational cost
SRFormer [28]	Transformer-based	Permuted self-attention, hierarchical structure	Handles larger windows for self-attention	Still computationally demanding

– T_F is the inference time for processing an image.

Given a scale factor "s," the problem can be formulated as:

$$\begin{aligned} \min_F \quad & T_F \\ \text{subject to} \quad & M_F \leq M_{\max} \\ & C_F \leq C_{\max} \end{aligned}$$

$M_{\max}$ and $C_{\max}$ are the maximum allowable memory and computational complexity, respectively.

3.2 Balancing Local and Global Context

Balancing local detail preservation with global consistency is essential in super-resolution tasks. Over-smoothing or artifacts may arise if this balance is not achieved.

Let:

– I_{LR} is the low-resolution image,
– $\hat{I}_{HR}$ be the predicted high-resolution image,
– I_{HR} be the ground-truth high-resolution image,
– $L_{\text{local}}(\hat{I}_{HR}, I_{HR})$ be the loss function for capturing local features (e.g., edges),
– $L_{\text{global}}(\hat{I}_{HR}, I_{HR})$ be the loss function for capturing global features (e.g., structure).

The optimization problem can be defined as:

$$\min_F \left[\lambda_{\text{local}} L_{\text{local}}(\hat{I}_{HR}, I_{HR}) + \lambda_{\text{global}} L_{\text{global}}(\hat{I}_{HR}, I_{HR})\right]$$

where λ_{local} and λ_{global} are weights balancing local and global losses.

3.3 Handling Noisy Low-Resolution Images

When processing noisy low-resolution images, earlier approaches may introduce rugged patterns or uneven edges and may not recover fine details effectively. The goal is to minimize noise while preserving fine details.

Let:

– I_{LR} is the noisy low-resolution image, modeled as $I_{LR} = I_{LR}^{\text{clean}} + n$, where n is the noise,
– $\hat{I}_{HR}$ be the predicted high-resolution image,
– $L_{\text{noise}}(\hat{I}_{HR}, I_{HR})$ is the loss function for noise reduction,
– $L_{\text{detail}}(\hat{I}_{HR}, I_{HR})$ be the loss function for recovering fine details.

The optimization problem can be formulated as:

$$\min_F \left[\lambda_{\text{noise}} L_{\text{noise}}(\hat{I}_{HR}, I_{HR}) + \lambda_{\text{detail}} L_{\text{detail}}(\hat{I}_{HR}, I_{HR})\right]$$

where λ_{noise} and λ_{detail} balance the trade-off between noise reduction and detail recovery.

3.4 Pseudo-code for Xception-Based Transformer Network (XTNSR)

The following is a pseudo-code for the suggested method, which combines Local Feature Window Transformer Blocks (LFWT) and Multi-Layer Feature Fusion Blocks (MLFF) with Transformer and Xception blocks in a multi-path network.

```
# Function Definitions for Various Components

Function XceptionBlock(Input):
# Apply depthwise separable convolutions followed by pointwise convolution
DWConv1 = DepthwiseConvolution(Input)
Conv1 = PointwiseConvolution(DWConv1)
Output1 = ReLU(Conv1)
DWConv2 = DepthwiseConvolution(Input)
Conv2 = PointwiseConvolution(DWConv2)
Output2 = ReLU(Conv2)
Return Output1 + Output2

Function LocalFeatureWindowTransformerBlock(Input):
# Apply Layer Normalization and Shifted Window Multi-Head Self-Attention
LN = LayerNormalization(Input)
SW_MSA = ShiftedWindowMultiHeadSelfAttention(LN)
Out = SW_MAS + Input LN = LayerNormalization(Out)
MLP = MultiLayerPerceptron(LN)
Output = MLP + Out
Return Output

Function MultiLayerFeatureFusion(X, LFWT):
# Fuse features from Xception and LFWT outputs
Concatenated = Concatenate(A1,A2, A3, A4, A5)
Fused = FullyConnectedLayer(Concatenated)
DWConv = DepthwiseConvolution(Fused)
Concat = Concatenate(DWConv, ReLU(DWConv))
Fused = FullyConnectedLayer(Concat)
Return Fused

Function XTNSRModel(LRImage):
# Initial Patch Embedding
Patched = PatchEmbedding(LRImage)

# Feature extraction
LFWT1 = LocalFeatureWindowTransformerBlock
(Patched)
X1 = XceptionBlock(LFWT1)
LFWT2 = LocalFeatureWindowTransformerBlock(X1)
X2 = XceptionBlock(LFWT2)
LFWT3 = LocalFeatureWindowTransformerBlock(X2)
LFWT4 = LocalFeatureWindowTransformerBlock
(LWFT3)
X3 = XceptionBlock(LFWT4)
LFWT5 = LocalFeatureWindowTransformerBlock(X3)
X4 = XceptionBlock(LFWT5)

# Multi-Layer Feature Fusion and Final Output
MLFFOutput = MultiLayerFeatureFusion(LFWT1,
LFWT2, LFWT3, LFWT4, LFWT5)
HR = Deconvolution(MLFFOutput) # Upscale to high-resolution
Return HR

# Main Execution Function
Function Main():

# Load Low-Resolution Image
LRImage = LoadImage("LowResolutionInput.jpg")

# Generate High-Resolution Image
HRImage = XTNSRModel(LRImage)

# Save the High-Resolution Image
SaveImage(HRImage, "HighResolutionOutput.jpg")

# Run the main function
Main()
```

Explanation:

XceptionBlock: Applies a series of depthwise separable convolutions and ReLU activations.
LocalFeatureWindowTransformerBlock: Uses layer normalization, shifted window multi-head self-attention, and a multi-layer perceptron to process features.
MultiLayerFeatureFusion: Combines and fuses feature from both Xception and Transformer blocks.
XTNSRModel: Extracts features through different (S1, S2, S3) shallow paths and (D1, D2, D3) dense feature paths as depicted.
Main: Drives the entire process from loading an image to saving the processed output.

4 Proposed Method

This section introduces our unique method for image super-resolution, which involves integrating a novel Local Feature Window Transformer (LFWT) Block and the Xception block into a multi-path framework and applying patch embedding at the input side. Patch embedding reduces the complexity of the model by transforming input images into smaller patches, which helps to address the difficulty of processing grid-like image data. To address the issue of over-smoothing, a multi-path framework that uses Local Feature Window Transformer (LFWT) Blocks helps to balance the local and global context. When the Xception Block is integrated into a Multi-Layer Feature Fusion (MLFF) Block through skip connections, the model's capacity to learn hierarchical features is enhanced, and it also aids in the fine recovery of noisy images and enhances training stability. The architecture of the Xception-Based Transformer Network for SISR (XTNSR) in Figure 2 (a) consists of six Local Feature Window Transformer (LFWT) Blocks and four Xception blocks connected in a multi-path framework. The initial features are extracted using a standard 3 × 3 convolution following the patch embedding at the input side. It utilizes six paths for input flow: Dense Feature Path for dense feature transmission and Shallow Feature Path for shallow features. Dense features traverse through three Dense Feature Paths (D1, D2, D3), and shallow features traverse through three shallow feature paths (S1, S2, S3). These features are fused in the Multi-Layer Feature Fusion (MLFF) Block. Finally, all features pass through the deconvolution layer for HR image reconstruction.

Here are some effective & concise highlights from the proposed XTNSR:

Efficient Patch Embedding Layer: The method incorporates a patch embedding layer that enhances scalability and balances computational efficiency with accuracy across different hardware platforms.

LFWT Block for Feature Extraction: The Local Feature Window Transformer (LFWT) Block captures both local and global features, addressing the limitations of over-smoothing and artifact generation during image super-resolution.

Hierarchical Feature Extraction with Xception Block: The Xception Block extracts hierarchical features, which, when integrated with the MLFF Block, enhances the recovery of fine details and mitigates noise in the super-resolution output.

Motivations for the proposed XTNSR:

The motivation for this article stems from the need to address several critical challenges in real-time image super-resolution, particularly in the context of built-in systems and intelligent hardware. These challenges include:

Computational Efficiency: Processing high-resolution photos as grid-like patches requires a significant amount of processing power, which increases memory usage and inference times. Models that balance efficiency without sacrificing performance are desperately needed.

Preserving Local and Global Context: Conventional techniques frequently fail to adequately capture local and global image features, resulting in problems such as artifacts and over-smoothing. To ensure high-quality image reconstruction while mitigating these issues, new methodologies must be developed.

Detail Recovery in Noisy Inputs: Previous approaches have shown to be challenging when handling noisy, low-resolution images, as they frequently introduce rough patterns and miss minute details. This calls for more sophisticated models to efficiently refine the image and extract hierarchical features, particularly for high-scale components in image super-resolution.

4.1 Initial Feature Extraction and Patch Embedding

Figure 2 (a) shows that the input's initial features are extracted using a normal 3 × 3 convolution, and then patch embedding is applied to the convoluted input. Equation 1 demonstrates the initial feature extraction stage.

$$H_0 = H_{Conv}(H_{LR}), \tag{1}$$

Here, $H_{Conv}(.)$ represents a 3 × 3 convolution operation. H_{LR} is the input Low-Resolution (LR) image, fed to the normal convolution for extracting initial features, and H_0 is the output of the convolution layer.

$$H_P = P_{embed}(H_0), \tag{2}$$

$P_{embed}(.)$ represents Patch Embedding, and H_P results from embeddings obtained after applying Patch embedding. After obtaining the initial features, patch embedding is applied on H_0, as depicted in Equation 2. Equation 2, depicts the Patch embedding over the convoluted output. The higher-dimensional embeddings obtained are input tokens to the subsequent Transformer. They help retain spatial information by capturing local patterns within each patch. The embedding tokens have sequence lengths of L and P as embedding dimension.

4.2 Local Feature Window Transformer Block (LFWT)

The Local Feature Window Transformer (LFWT) Block is constructed using the Shifted Window Multi-head Self-Attention (SW-MSA) [27] module and the Multi-layer Perceptron (MLP) [67] with Rectified Linear Unit (ReLU) [68] activation. Each module applies a layer norm (LN) before the MLP and SW-MSA modules. Following every module is another residual Skip connection.

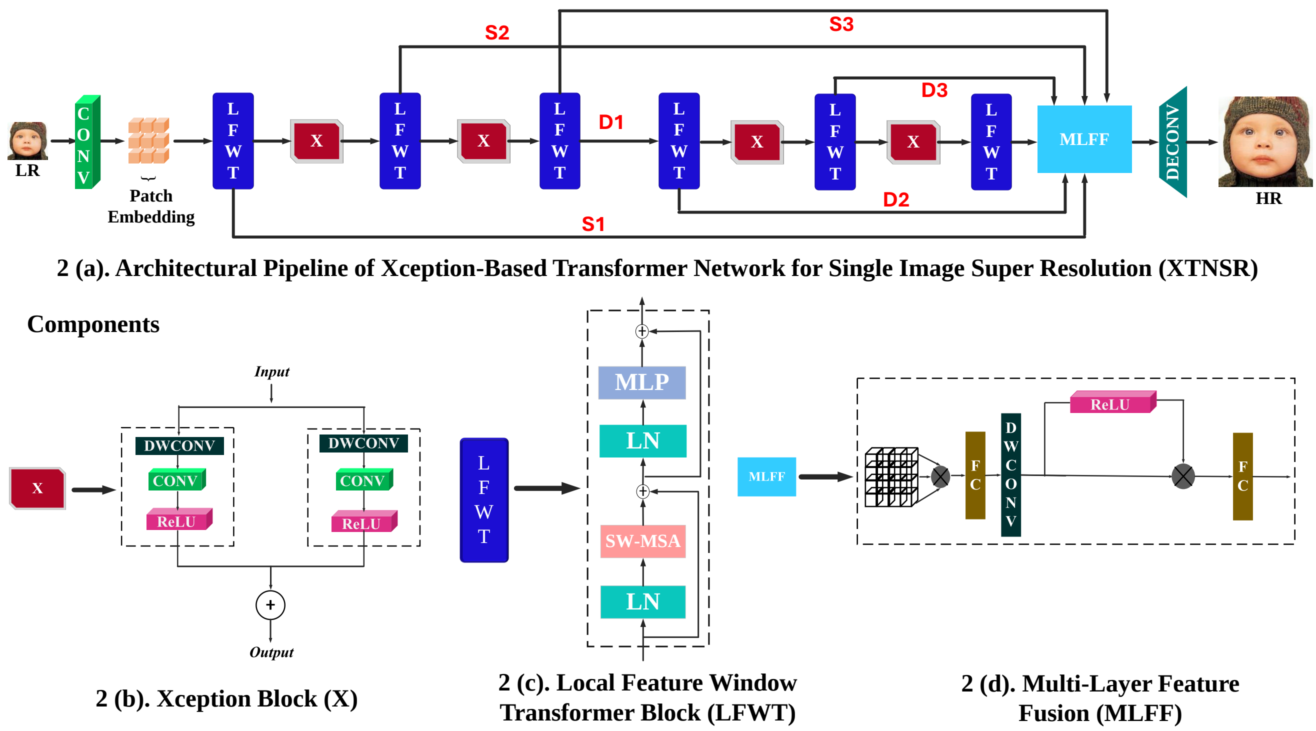

Fig. 2: The design of the suggested network structure of the Xception-Based Transformer Network for Single Image Super-Resolution (XTNSR).

As seen in Figure 2(c), the embeddings from the previous layer are fed to the SW-MSA module present in the LFWT Block, followed by Layer Norm. This module's output is then added to the embeddings passed through the residual connection and transferred to the next MLP module. The output of the MLP module is then added to the summed output of the previous module. Equations 3 and 4 show the SW-MSA and MLP module operation in the LFWT Block.

$$M_{SWMSA} = H_{SWMSA}(H_{LN}(H_{I1})) + H_{I1}, \tag{3}$$

Here, H_{I1} is the input to the Local Feature Window Transformer Block, $H_{LN}(.)$ is the Layer Norm function, $(H_{SWMSA})(.)$ is the Shifted Window Multi-head Self-Attention function, M_{SWMSA} is the output of the SW-MSA module in the LFWT Block.

$$H_{LFWT} = H_{MLP}(H_{LN}(M_{SWMSA})) + M_{SWMSA}, \tag{4}$$

In Equation 4, $H_{MLP}(.)$ is the output function of the Multi-layer Proton, and H_{LFWT} represents the output of the Local Feature Window Transformer (LFWT) Block.

4.3 Xception Block (X)

As presented in Figure 2(b), the Xception Block is designed to be relatively lightweight. The Xception block introduces depthwise separable convolution to make it more computationally efficient and reduces the number of parameters while maintaining strong performance.

It consists of a series of depthwise separable and pointwise convolution combinations followed by ReLU non-linearity. This series combination is then connected in parallel. The output from the two parallelly connected modules is then summed up to obtain the output from the Xception Block. This enables the model to learn hierarchical features within the same layer.

$$\begin{aligned} H_X = {} & \mathrm{ReLU}\big(H_{\mathrm{Conv}}(H_{\mathrm{DWConv}}(H_{\mathrm{SWT}}))\big) \\ & + \mathrm{ReLU}\big(H_{\mathrm{Conv}}(H_{\mathrm{DWConv}}(H_{\mathrm{SWT}}))\big), \end{aligned} \tag{5}$$

Here, $H_{\mathrm{DWConv}}(.)$ is the Depthwise Separable convolution operation function, $H_{\mathrm{Conv}}(.)$ is the Pointwise convolution operation function, $ReLU(.)$ represents the non-linearity, and H_X represents the output of the Xception Block.

Including Xception Block and Transformers in a multi-path framework allows the network to balance an image's local and global details. Since the Xception

Block operates both spatial and channel dimensions simultaneously. It also incorporates residual connections to help with the flow of gradients during training and mitigate the vanishing gradient problem.

4.4 Multi-Layer Feature Fusion Block (MLFF)

As seen in Figure 2(d), the Multi-Layer Feature Fusion Block (MLFF) is designed to merge features from the multiple paths present in the network. This block helps to merge the dense and shallow features transferring through different paths in the network. It concatenates the features from multiple paths, and then the fused features pass through the Depthwise Separable convolution followed by a Fully Connected layer. After that, the feature is passed through a residual connection and a ReLU non-linearity and fused again to pass through the Fully Connected layer. It helps to point the spatial region information from different paths in a network. Furthermore, it creates a multi-path representation of the input image and highlights the important features that contribute to improving the model's performance. Finally, all the features are aggregated and passed through a deconvolution layer to generate the High-Resolution image.

Equations 6, 7, 8, 9, and 10 stepwise describe how the Multi-Layer Feature Fusion (MLFF) Block merges the features from multiple paths in the network. As the features from multiple paths enter the MLFF block, they are concatenated, as shown in Equation 6.

$$MLFF_{Cat} = cat(H_{S1}, H_{S2}, H_{S3}, H_{D1}, H_{D2}, H_{D3}), \quad (6)$$

In Equation 6, $MLFF_{Cat}$ denotes the output of the concatenation of features from multiple paths in the MLFF Block, $cat(.)$ is the concatenation operation, $H_{S1}, H_{S2}, H_{S3}, H_{D1}, H_{D2}, andH_{D3}$ are the shallow and dense features from paths S1, S2, S3, and dense feature paths D1, D2, and D3.

After this, these concatenated features are passed through a Depthwise Separable convolution layer and a Fully Connected layer to reduce the computational burden.

$$MLFF_{DW} = H_{DWConv}(H_{FC}(MLFF_{Cat})), \quad (7)$$

In Equation 7, $H_{FC}(.)$ is the Fully Connected layer function, $MLFF_{DW}$ is the result obtained in the MLFF Block by the Depthwise Separable convolution.

The generated features are concatenated together after applying the non-linearity and residual connection.

$$MLFF_{FC} = cat(ReLU(MLFF_{DW}), MLFF_{DW}), \quad (8)$$

$MLFF_{FC}$, as seen in Equation 8, is the input of the last Fully Connected layer in the MLFF Block.

The concatenated output is then passed to the Fully Connected layer to better generalize the data.

$$MLFF_{O} = H_{FC}(MLFF_{FC}), \quad (9)$$

$MLFF_{O}$ in Equation 9 is the output of the MLFF Block.

Finally, the Multi-Layer Feature Fusion Block (MLFF) output is fed to the deconvolution layer to reconstruct the high-resolution image.

$$HR_{O} = H_{DConv}(MLFF_{O}), \quad (10)$$

Lastly, in Equation 10, $H_{DConv}(.)$ is the deconvolution operation, and HR_{O} is the generated high-resolution output.

5 Experimental Findings

Our proposed XTNSR model's performance was verified through experiments conducted on benchmark test datasets, resulting in quantitative and qualitative visual results. Analysis based on average PSNR (dB) SSIM has also been graphically demonstrated for state-of-the-art methods. Additionally, this section discusses the computational cost in terms of execution time and network parameters. Analysis for different numbers of Local Feature Window Transformers (LFWT) and Xception Block has also demonstrated which combinations give the best qualitative and quantitative results. Furthermore, time and space complexity analysis with different state-of-the-art models is also shown. Moreover, the ablation analysis includes the Flops v/s PSNR computation, the results of changing the LFWT Block's window size during testing, comparison with traditional denoising methods to show the model's performance on noisy images, changing the non-linearity function in the Xception and Multi-Layer Feature Fusion (MLFF) Block, its PSRN and SSIM convergence, and loss analysis.

5.1 Experiment Configuration

This section presents the datasets, evaluation metrics, and training specifics used to test and train our suggested structure on publicly available datasets. It is

Table 2: Hyperparameters and settings for the model

Hyperparameter	Value
Patch Size	48 × 48
Window Size	24 × 24
Scale Factors	×2, ×3, ×4, ×8
Training Samples	DIV2K
Testing Datasets	Set5, Set14, BSD100, Urban100, Manga109
Batch Size	4
Optimizer	Adam
Optimizer β_1	0.90
Optimizer β_2	0.99
Learning Rate Schedule	Halved every 200 epochs, maintained at 10^{-4}
Total Iterations	1000
GPU	NVIDIA GeForce GTX 2080ti with 24GB of RAM
Programming Language	Python 3.6
Platform	PyTorch 1.7.0
Data Augmentation	Random rotation (90°, 180°, 270°) and flipping
Downsampling Method	Bicubic kernels

mentioned that the testing and training sets are different. The hyperparameters used in the experiments are shown in Table 2.

5.1.1 Datasets

The DIV2K [69] dataset has been used for network training and validation. DIV2K [69] is divided into 800 high-quality images, 100 validation images, and 100 test images for the training phase. Out of these 1000 images, we have used 800 images for training. We tested using five benchmark datasets: Set5 [70], Set14 [71], BSD100 [72], Urban100 [73], and Manga109 [74].

5.1.2 Evaluation Metrics

For a fair comparison with earlier research, the standard metrics Peak Signal-to-Noise Ratio (PSNR) (dB) and the structural similarity index (SSIM) were calculated for quantitative measures. Equations 11 and 12 show the mathematical expression of PSNR and SSIM measures.

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{M^2}{\frac{1}{N}\sum_{j=1}^{N}(I_G(j) - \hat{I}_S(j))^2}\right), \quad (11)$$

Here, PSNR is the Peak Signal-to-Noise Ratio measured in decibels (dB), M is the maximum pixel value having N number of pixels, I_G is the ground truth image, and $\hat{I}_S$ is the reconstructed image.

$$\mathrm{SSIM}(I; \hat{I}) = \left[\mathrm{C}_l(I; \hat{I})\right]^{\alpha} \left[\mathrm{C}_c(I; \hat{I})\right]^{\beta} \left[\mathrm{C}_s(I; \hat{I})\right]^{\gamma}, \quad (12)$$

Here, SSIM represents the Structural Similarity Index, I, is the ground truth image, $\hat{I}$ is the reconstructed HR image, α, β, γ are exponents that control the relative importance of the luminance, contrast, and structure, and C_l, C_c, C_s are the luminance, contrast, and structure.

5.1.3 Training and Testing Specifics

To train our model, we used 48 × 48 low-resolution patches. The window size is chosen to be 24 × 24. MATLAB R2022b was used to create low-resolution images for scale ×2, ×3, ×4, and ×8. An NVIDIA GeForce GTX 2080ti GPU with 24GB of RAM is used to train the suggested network. The proposed model's algorithm was coded using the programming language Python 3.6 and the platform PyTorch 1.7.0. Eight hundred samples are taken from the DIV2K [69] dataset for model training. We use an Adam optimizer with $\beta_1 = 0.90$ and $\beta_2 = 0.99$ for optimization. Every 200 epochs, the learning rate of the suggested model is halved and maintained at 10^{-4}. The model has been trained for 1000 iterations. Five common benchmark datasets, namely Set5 [70], Set14 [71], BSD100 [72], Urban100 [73], and Manga109 [74], were used to test our suggested model. Bicubic kernels are used to downsample the HR images to produce the LR image. The window size during testing is also set for 24 × 24. The training samples are randomly cropped into 48 × 48 patches, and the batch size of training images is set as 4. Additionally, data augmentation [75] generates additional samples for the algorithm through random rotation and flipping for 90, 180, and 270 degrees.

Data Pre Processing

Normalization: To guarantee a consistent pixel intensity distribution throughout the dataset and promote quicker and more stable model convergence, all images are normalized.

Data Collection and Preprocessing: Bicubic interpolation is first employed to process the High-Resolution (HR) images used in this study to produce comparable Low-Resolution (LR) images. Cubic polynomials are used in bicubic interpolation, a popular resampling method in image processing, to improve the resulting image's smoothness. For our purposes, this approach is very helpful because it maintains edge clarity more effectively than more basic methods like bilinear or nearest neighbor interpolation.

Details about Bicubic Downsampling: Using a bicubic kernel, each HR picture is methodically downsampled to produce its LR counterpart. To ensure that the resulting LR images preserve as much contextual and textural information as possible from their HR beginnings, this kernel modifies pixel values to account for the cubic interpolation of surrounding pixel values.

Patch extraction and Data Augmentation: HR and LR pictures are split into patches after downsampling. HR photos are specifically cropped into 48×48 patches. This dimension was selected to compromise acceptable processing overhead and adequate spatial information. In testing, corresponding LR patches are produced with a window size of 24×24, matching the input dimensions used in training the model.
Extensive data augmentation approaches are utilized to improve our model's robustness and diversity of training data. Training samples are randomly rotated 90, 180, and 270 degrees, as well as flipped both vertically and horizontally. To improve generalization across unknown data during testing, these transformations are essential for the model to learn invariant features from different orientations and mirrorings of image data.

5.1.4 Software/Hardware Specifications

We have given full software/hardware systems used in simulations and implementations for our proposed XTNSR:

Software Stack

Framework: PyTorch 1.7.0: Used for building and training deep learning models, including the CNN and Transformer components.
Torchvision 0.12.0: Supports image transformations and dataset management.
Torchaudio 0.11.0: Facilitates audio processing in any auxiliary applications related to the ISR model.
CUDA Toolkit 11.3: Facilitates GPU-accelerated computations necessary for training deep learning models.
cuDNN (NVIDIA): Enhances performance by providing optimized primitives for deep learning.
Matplotlib: Used for generating visualizations and plotting results.
TQDM: For creating progress bars and monitoring model training.

Operating System: Ubuntu 18.04 LTS: Provides the Linux environment required for high-performance computing, software compatibility, and system stability.

Programming Languages and Tools:
Python 3.6: The primary programming language for implementing the XTNSR model, facilitating integration with libraries like PyTorch.
MATLAB R2022b: Utilized for data preprocessing and generating low-resolution images for training and validation.

Hardware Systems

GPU:
NVIDIA GeForce GTX 2080ti GPU (24GB RAM): Used for training the models due to its capability to handle large-scale computations required by Transformers.

5.2 Evaluations based on quantitative metrics in state-of-the-art methods

The standard quantitative metric comparison of five benchmark test datasets for scale factors $\times 2$, $\times 3$, $\times 4$, and $\times 8$ is presented in **Table 3**. We have used 15 state-of-the-art algorithms widely accepted for Single Image Super-Resolution, including Bicubic, SRCNN [8], FSRCNN [9], VDSR [10], MemNet [15], LapSRN [12], SENext [40], RCAN [21], MFCC [47], EDSR [17], HAN [24], NLSN [23], SwinIR [26], ELAN [25], and SRFormer [28] to show comparison with our proposed XTNSR. Regarding PSNR and SSIM, our suggested XTNSR analytical outcomes have considerably surpassed the state-of-the-art techniques. In addition, our suggested approach outperformed other SOTA models in terms of PSNR and SSIM on all averages for benchmark datasets.

5.3 Comparative study of network parameters and execution time

Parametric and performance comparison of the network on the Set5 [70] test dataset with up-sampling factor

Table 3: Standard metric assessment of our suggested XTNSR against state-of-the-art SR methods for up-scaling factors ×2, ×3 ×4, and ×8. The highest score is bolded and colored **Red** . The second-greatest score is highlighted and displayed in Blue.

Method	Factor	#Param	Set5 [70]		Set14 [71]		BSD100 [72]		Urban100 [73]		Manga109 [74]		Average	
			PSNR↑	SSIM↑	PSNR↑	SSIM ↑	PSNR↑	SSIM ↑	PSNR↑	SSIM ↑	PSNR↑	SSIM ↑	PSNR↑	SSIM ↑
Bicubic	×2	-/-	33.68	0.9304	30.24	0.8691	29.56	0.8435	26.88	0.8405	31.05	0.9349	30.23	0.8832
SRCNN [8]	×2	57K	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.51	0.8946	35.72	0.9680	33.11	0.9219
FSRCNN [9]	×2	12K	36.98	0.9556	32.62	0.9087	31.50	0.8904	29.58	0.9009	36.62	0.9710	33.56	0.9260
VDSR [10]	×2	665K	37.53	0.9587	33.05	0.9127	31.90	0.8960	30.77	0.9141	37.16	0.9740	33.24	0.9314
MemNet [15]	×2	677K	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740	34.43	0.9330
LapSRN [12]	×2	812K	37.52	0.9591	32.99	0.9124	31.80	0.8949	30.41	0.9101	37.53	0.9740	33.87	0.9302
SENext [40]	×2	97K	38.04	0.9608	34.24	0.9181	32.21	0.8997	32.43	0.9287	38.79	0.9774	35.14	0.9369
RCAN [21]	×2	16,000K	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786	35.52	0.9405
MFCC [47]	×2	1,861K	38.16	0.9606	33.85	0.9195	32.28	0.9010	32.65	0.9331	39.11	0.9780	35.21	0.9384
EDSR [17]	×2	43,000K	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773	35.28	0.9386
HAN [24]	×2	3,230K	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785	35.53	0.9405
NLSN [23]	×2	4,475K	38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789	35.57	0.9412
SwinIR [26]	×2	878K	38.38	0.9620	34.24	0.9233	32.47	0.9032	33.51	0.9401	39.70	0.9794	35.66	0.9416
ELAN [25]	×2	621K	38.36	0.9620	34.20	0.9228	32.45	0.9030	33.44	0.9391	39.62	0.9793	35.61	0.9412
SRFormer [28]	×2	853K	38.45	0.9622	34.21	0.9236	32.51	0.9038	**33.86**	**0.9426**	39.69	0.9786	35.74	0.9422
XTNSR (Ours)	×2	1,875K	**38.58**	**0.9626**	**34.27**	**0.9255**	**32.58**	**0.9043**	33.64	0.9408	**39.78**	**0.9799**	**35.76**	**0.9426**
Bicubic	×3	-/-	30.40	0.8686	27.54	0.7741	27.21	0.7389	24.46	0.7349	26.95	0.8566	27.31	0.7945
SRCNN [8]	×3	57K	32.75	0.9090	29.29	0.8215	28.41	0.7863	26.24	0.7991	30.48	0.9117	29.44	0.8455
FSRCNN [9]	×3	12K	33.16	0.9140	29.42	0.8242	28.52	0.7893	26.41	0.8064	31.10	0.9210	29.70	0.8516
VDSR [10]	×3	665K	33.66	0.9213	29.78	0.8318	28.83	0.7976	27.14	0.8279	32.01	0.9340	30.28	0.8624
MemNet [15]	×3	677K	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369	30.62	0.8669
LapSRN [12]	×3	812K	33.82	0.9227	29.79	0.8320	28.82	0.7973	27.07	0.8271	32.21	0.9350	30.36	0.8631
SENext [40]	×3	54K	34.32	0.9255	**31.08**	0.8419	29.11	0.8047	28.60	0.8519	33.63	0.9451	31.35	0.8738
RCAN [21]	×3	16,000K	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499	31.64	0.8818
MFCC [47]	×3	2,230K	34.67	0.9294	30.51	0.8456	29.22	0.8080	28.64	0.8616	34.15	0.9478	31.43	0.8793
EDSR [17]	×3	43,000K	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476	31.48	0.8792
HAN [24]	×3	3,230K	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500	31.66	0.8819
NLSN [23]	×3	4,475K	34.85	0.9306	30.70	0.8485	29.34	0.8117	29.25	0.8726	34.57	0.9508	31.74	0.8824
SwinIR [26]	×3	886K	34.89	0.9312	30.77	0.8503	29.37	0.8124	29.29	0.8744	34.74	0.9518	31.81	0.8840
ELAN [25]	×3	629K	34.90	0.9313	30.80	0.8504	29.38	0.8124	29.32	0.8745	34.73	0.9517	31.82	0.8841
SRFormer [28]	×3	861K	34.94	0.9318	30.81	0.8518	29.41	0.8142	**29.52**	**0.8786**	34.78	0.9524	31.89	**0.8857**
XTNSR (Ours)	×3	1,875K	**35.02**	**0.9322**	30.84	**0.8519**	**29.46**	**0.8144**	29.38	0.8755	**34.90**	**0.9525**	**31.92**	0.8853
Bicubic	×4	-/-	28.43	0.8109	26.00	0.7023	25.96	0.6678	23.14	0.6574	25.15	0.7890	25.68	0.7250
SRCNN [8]	×4	57K	30.48	0.8628	27.50	0.7513	26.90	0.7103	24.52	0.7226	27.66	0.8580	27.40	0.7785
FSRCNN [9]	×4	12K	30.70	0.8657	27.59	0.7535	26.96	0.7128	24.60	0.7258	27.89	0.8590	27.57	0.7850
VDSR [10]	×4	665K	31.35	0.8838	28.02	0.7678	27.29	0.7252	25.18	0.7525	28.82	0.8860	28.13	0.8031
MemNet [15]	×4	677K	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942	28.46	0.8094
LapSRN [12]	×4	812K	31.54	0.8866	28.09	0.7694	27.32	0.7264	25.21	0.7553	29.09	0.8900	28.27	0.8060
SENext [40]	×4	54K	31.50	0.8947	28.99	0.7812	**28.49**	0.7357	26.64	0.7839	30.48	0.9084	29.22	0.8208
RCAN [21]	×4	16,000K	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173	29.46	0.8317
MFCC [47]	×4	2,157K	32.42	0.8973	28.73	0.7849	27.67	0.7399	26.48	0.7977	30.98	0.9131	29.25	0.8265
EDSR [17]	×4	43,000K	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148	29.32	0.8289
HAN [24]	×4	3,230K	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177	29.52	0.8321
NLSN [23]	×4	4,475K	32.59	0.9000	28.87	0.7891	27.78	0.7444	26.96	0.8109	31.27	0.9184	29.49	0.8325
SwinIR [26]	×4	897K	32.72	0.9021	28.94	0.7914	27.83	0.7459	27.07	0.8164	31.67	0.9226	29.64	0.8356
ELAN [25]	×4	621K	32.75	0.9022	28.96	0.7914	27.83	0.7459	27.13	0.8167	31.68	0.9226	29.67	0.8357
SRFormer [28]	×4	873K	32.81	0.9029	29.01	0.7919	27.85	0.7472	**27.20**	**0.8189**	31.75	**0.9237**	29.72	0.8369
XTNSR (Ours)	×4	1,875K	**32.82**	**0.9030**	**29.03**	**0.7931**	27.99	**0.7473**	27.36	0.8192	**31.76**	0.9232	**29.79**	**0.8372**
Bicubic	×8	-/-	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500	22.68	0.5876
SRCNN [8]	×8	57K	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440	22.46	0.6950	23.42	0.5739
FSRCNN [9]	×8	12K	25.60	0.6970	24.00	0.5990	24.31	0.5720	21.45	0.5500	22.72	0.6920	23.46	0.5696
VDSR [10]	×8	665K	25.93	0.7240	24.26	0.6140	24.49	0.5830	21.70	0.5710	23.16	0.7250	23.50	0.5800
MemNet [15]	×8	677K	26.16	0.7414	24.38	0.6199	24.58	0.5842	21.89	0.5825	23.56	0.7387	24.11	0.6529
LapSRN [12]	×8	812K	26.15	0.7380	24.35	0.6200	24.54	0.5860	21.81	0.5810	23.39	0.7350	24.04	0.6520
MSRN [76]	×8	6,226K	26.59	0.7254	24.88	0.5961	24.70	0.5610	22.37	0.6077	24.30	0.7701	24.56	0.6520
EDSR [17]	×8	43,000K	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.69	0.7841	24.74	0.6824
AWSRN [77]	×8	2,348K	26.97	0.7747	24.96	0.6414	24.80	0.5967	22.45	0.6174	24.69	0.7842	24.77	0.6828
DBPN [78]	×8	10,000K	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.60	0.7732	24.75	0.6824
MFCC [47]	×8	2,453K	27.07	0.7762	25.01	0.6412	24.84	0.5980	22.54	0.6196	24.63	0.7791	24.81	0.6828
RDN [45]	×8	21,900K	27.21	0.7840	25.13	0.6480	24.88	0.6010	22.73	0.6312	25.14	0.7897	25.02	0.6907
RCAN [21]	×8	16,000K	27.31	0.7878	25.23	0.6511	24.98	0.6058	23.00	0.6452	25.24	0.8029	25.15	0.6985
SENext [40]	×8	97K	26.87	0.7415	**25.73**	0.6200	**26.79**	0.5847	21.90	0.5829	23.96	0.7389	25.05	0.6536
HAN [24]	×8	3,230K	27.33	0.7884	25.24	0.6510	24.98	0.6059	22.98	0.6437	25.20	0.8011	25.14	0.6980
XTNSR (Ours)	×8	1,875K	**27.62**	**0.7910**	25.34	**0.6519**	25.16	**0.6069**	**23.22**	**0.6461**	**25.42**	**0.8038**	**25.35**	**0.6999**

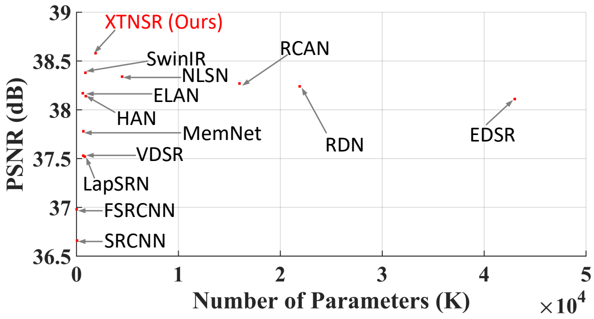

Fig. 3: Inspection of PSNR for model parameters on ×2 up-scaling factor using the Set5 [70] image test dataset.

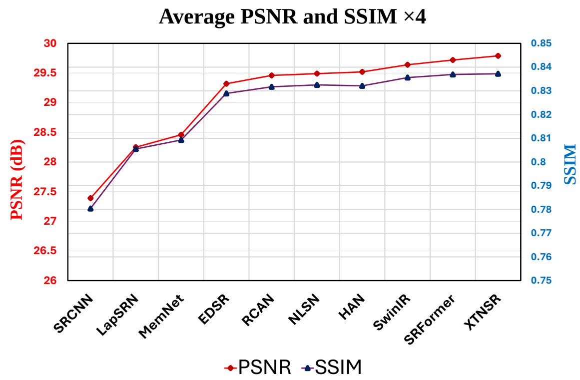

Fig. 5: Average PSNR and SSIM evaluation on up-scale factor ×4.

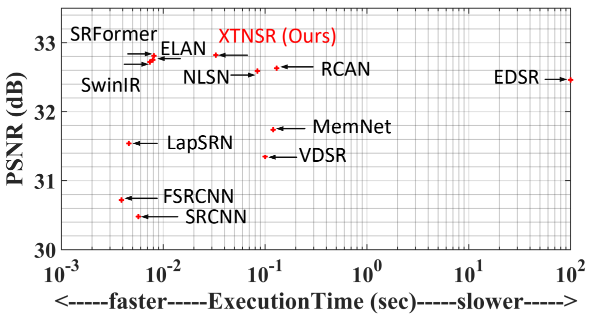

Fig. 4: Inspection of PSNR for execution time on ×4 up-scaling factor using the Set5 [70] image test dataset.

×2 has been demonstrated in Figure 3. The parameters of XTNSR are approximately 96% lower than those of EDSR [17], 88% lower than RCAN [21], 74% lower than RDN [45], and 66% lower than NLSN [23]. Percentage reduction in the parameters with an increase in performance shows that compared to alternative deep learning techniques, the XTNSR model contributes to a more effective model size reduction with significantly higher performance gains. Furthermore, performance versus execution has been shown over the Set5 [70] test dataset on up-scaling factor ×4 in Figure 4. From Figure 4, it can be observed that our suggested approach shows the maximum PSNR value, i.e., 32.82 (dB), and in terms of execution time, it is faster than five state-of-the-art methods (VDSR [10], MemNet [15], EDSR [17], RCAN [21] and NLSN [23]).

5.4 Assessment of average PSNR and SSIM on up-scaling factors ×4 and ×8 on the Image Test Datasets

Reporting average PSNR and SSIM scores has become standard practice in image super-resolution. This standardization facilitates a consistent and easy interpretation of the quality of super-resolved images. By calculating and reporting these metrics, researchers can demonstrate the effectiveness of their proposed approach compared to existing techniques or baselines. As seen in Figure 5 and Figure 6, our proposed XTNSR has the best average PSNR and SSIM compared to other state-of-the-art methods. The average PSNR and SSIM values are generated by taking an average of PSNR and SSIM values of all the Image SR test Dataset (i.e., Set5 [70], Set14 [71], BSD100 [72], Urban100 [73], Manga109 [74]). The average values for all the state-of-the-art models are also shown in Table 1. Figure 5 and Figure 6 represent the average values for up-scaling factors ×4 and ×8. For the up-scaling factor ×4, our suggested XTNSR achieves the most favorable outcome in comparison to RCAN [21], NLSN [23], SwinIR [26], HAN [24], and SRFormer [28]. Similarly, for the up-scaling factor ×8, it shows higher outcomes as compared to MemNet [15], DBPN [78], EDSR [17], RCAN [21], and HAN [24], respectively.

5.5 Analysis with different combinations of Local Feature Window Transformer (LFWT) and Xception Blocks

This study checked combinations of several Local Feature Window Transformers and Xception Block. Starting with using 5 LFWT and 5 Xception Blocks and try-

Table 4: Different Combinations of LFWT and Xception Block

Number of LFWT Blocks	Number of Xception Blocks	Average PSNR	Average SSIM
5	5	37.58	0.9598
5	6	37.62	0.9602
6	5	37.78	0.9609
6	4	**37.82**	**0.9611**
4	6	37.76	0.9606
4	4	37.49	0.9587
6	6	37.68	0.9604

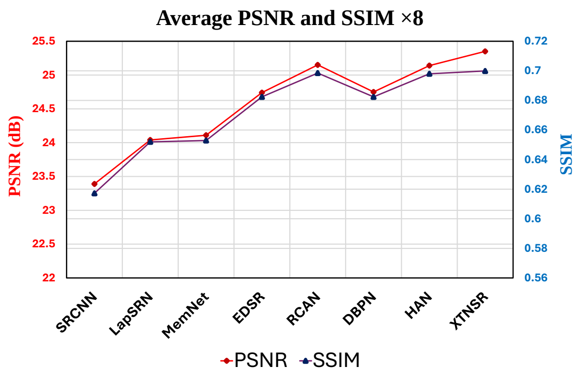

Fig. 6: Average PSNR and SSIM evaluation on up-scale factor ×8.

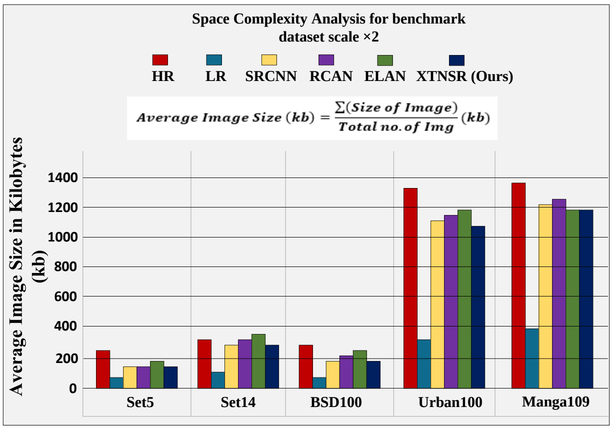

Fig. 7: Space Complexity assessment of network on Image SR Test Data Sets on up-scaling factor ×2.

ing different combinations, we observed that the best average PSNR and SSIM are obtained with the network having 6 LFWT Blocks and 4 Xception blocks. This analysis was done by training the model for 500 epochs at scale ×2. Then, the combination of 6 LFWT blocks and 4 Xception blocks was adopted in the final training of the network for 1000 epochs to obtain the best performance gains at limited computational expense. Table 4 shows that the best value of PSNR and SSIM is observed to be with 6 LFWT Blocks and 4 Xception Blocks and is highlighted in **red**, and the second-best performance is observed with six LFWT Blocks and five Xception Blocks highlighted in blue.

5.6 Network complexity analysis in terms of space and time

Space complexity analysis of a deep learning model typically refers to the amount of memory or storage required by the model during training and inference. The space complexity depends on various factors, including the model architecture, parameters, and input data size. In this section, the space complexity for image data has been presented. We assess the space complexity on the Image SR text datasets with an up-scaling factor of ×2. The space complexity of our proposed XTNSR, along with HR, LR, SRCNN [8], RCAN [21], and ELAN [25], is shown in Figure 7.

A deep learning model's time complexity can be seen in the time needed to complete each epoch during training. The model's overall complexity affects the time complexity, including the neural network's depth and width. The ability to parallelize computations across multiple devices or nodes can impact time complexity. Optimization techniques, such as model quantization or pruning, may impact training and inference time complexity. Figure 8 compares our suggested XTNSR with state-of-the-art NLSN [23] and ELAN [25] regarding time complexity. From Figure 8, it can be observed that in comparison to other methods, our proposed XTNSR network takes less time per epoch for 100 training epochs. This shows our model is also better in terms of time complexity.

5.7 Perceptual Quality Comparison

The perceived accuracy at up-scaling factors ×4 and ×8 for image SR testing datasets, such as Set5 [70], Set14 [71], BSD100 [72], Urban100 [73], and Manga109 [74], is presented in Figure 9, Figure 10, Figure 11, Fig-

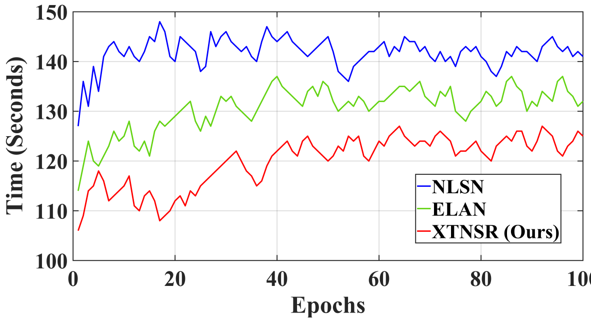

Fig. 8: Evaluation of run time against PSNR on Set5 [70] for up-scale factor ×4.

ure 12, Figure 13, Figure 14, Figure 15, and Figure 16. While improving an image for an enlargement factor of ×8 is challenging, our suggested XTNSR reconstructs the finer details in an image, efficiently removes general artifacts, and provides realistic visual results by reducing over-smoothing. Some examples of the generated high-quality images from each test dataset at scale factor ×4 and ×8 have been presented in Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 147, Figure 15, and Figure 16. On up-scaling factor ×4, for Set14 [71], we used the Barbara image, and for the BSD100 [72] dataset, we used the Img_78004 image. For the Urban100 [73] dataset, we used Img_004, and for the Manga109 [74] dataset, we used the GakuenNoise Image. For these images, we have shown the comparison of our proposed method with eight existing state-of-the-art methods, including Bicubic, MSRN [76], EDSR [17], AWSRN [77], RCAN [21], NLSN [23], SwinIR [26] and SRFormer [28].

5.8 Ablation assessment

This section demonstrates controlled experimentation by changing algorithm hyperparameters, significantly affecting the model's performance in the ablation assessment section. The model comprises 6 Local Feature Window Transformer (LFWT) Blocks, 4 Xception Blocks, and 1 Multi-layer Feature Fusion Block. Controlled experiments have been performed on each block to determine their effect on the network's overall performance.

Three types of ablation assessment have been performed on the model: (1) Ablation study for FLOPs versus PSNR, (2) Ablation study for changing the window size of the Local Feature Window Transformer Block, (3) Ablation study using traditional denoising methods, (4) Ablation study by changing activation functions to ReLU, PReLU, and CReLU in the Xception Block, (5) Ablation study by changing non-linearity in the Multi-Layer Feature Fusion (MLFF) Block,

5.8.1 Ablation study for FLOPS v/s PSNR

A study comparing PSNR (Peak Signal-to-Noise Ratio) against FLOPs (Floating Point Operations) in ISR systematically analyzes the trade-off between computational complexity and reconstruction quality. Using ten state-of-the-art image SR methods, we compared the computational cost of floating-point operations per second (FLOPs) versus PSNR for Set 5 [70] on scale factor ×4 shown in Figure 17. Figure 17 shows that, compared to SRFormer and ELAN, our suggested (XTNSR) method has fewer FLOPs.

5.8.2 Analysis with different window sizes in the Local Feature Window Transformer (LFWT) Block

To assess the effect of different window sizes, we set the window sizes to be 8 × 8, 16 × 16, and 24 × 24 and compare our proposed model with existing state-of-the-art transformer-based methods like SwinIR [26] and SRFormer [28] on image SR test dataset for up-scaling factor ×4. Table 5 compares our model with other Image SR models evaluated for different window sizes. Larger window sizes increase the super-resolution model's accuracy. As seen from Table 5's quantitative values, our model performs better in PSNR and SSIM on publicly available datasets.

5.8.3 Ablation assessment using traditional denoising methods

In this section, we present a comparative analysis of our XTNSR model applied to the Urban100 [73] Dataset at a scale of ×2 against traditional denoising approaches including Block Matching and 3D Filtering (BM3D) [79], a Fast and Flexible Solution for CNN-Based Image Denoising (FFDNet) [80], Nonlocally Centralized Sparse Representation (NCSR) [81], and Denoising Convolutional Neural Network (DnCNN) [11]. The evaluation is based on PSNR metrics under Gaussian noise with varying noise levels (σ), specifically $\sigma = 5$, $\sigma = 10$, and $\sigma = 15$, as summarized in Table 6.

5.8.4 Ablation assessment for color image denoising

In this section, we show a quantitative evaluation of color image denoising on more critical datasets like Kodak24 [82], General100 [9], and McMaster [83] for Gaussian noise having noise levels $\sigma = 5$, $\sigma = 10$, and $\sigma = 15$, over a scale factor of ×2. We have used seven

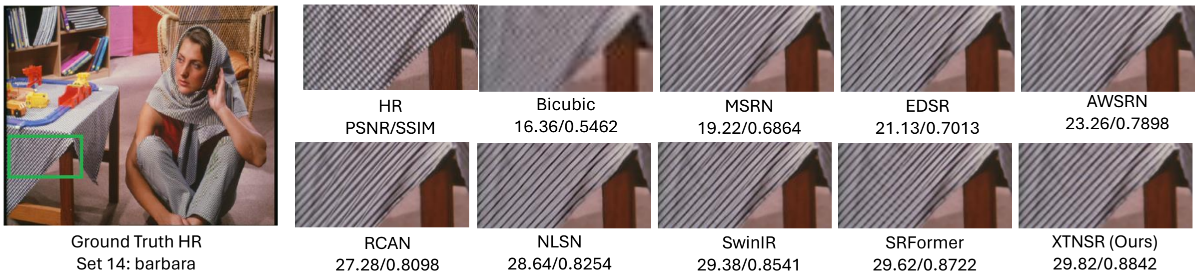

Fig. 9: Barbara's image from the Set14 [71] dataset perceptual improvement on a ×4 up-scaling factor.

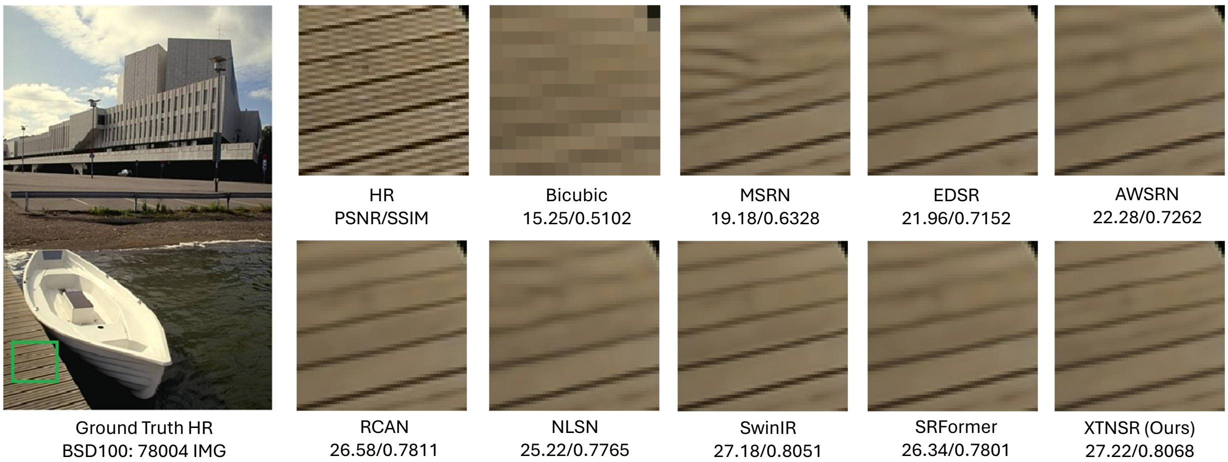

Fig. 10: Img_78004 image from the BSD100 [72] dataset perceptual improvement on a ×4 up-scaling factor.

Table 5: Impact of varying window sizes on performance for ×4 upscaling factor. **Red** indicates the best quantitative value, whereas blue indicates the second-best quantitative value.

Method	Window Size	Set5 [70]		Set14 [71]		BSD100 [72]		Urban100 [73]		Manga109 [74]	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR [26]	8 × 8	38.24	0.9615	33.94	0.9212	32.39	0.9023	33.09	0.9373	39.34	0.9784
	16 × 16	38.32	0.9618	34.00	0.9212	32.44	0.9030	33.40	0.9394	39.53	0.9791
	24 × 24	38.35	0.9620	34.04	0.9214	**32.48**	**0.9034**	33.54	0.9402	39.71	**0.9798**
SRFormer [28]	8 × 8	38.20	0.9611	34.06	0.9214	32.36	0.9021	32.92	0.9361	39.10	0.9777
	16 × 16	38.31	0.9617	34.10	0.9217	32.43	0.9026	33.26	0.9385	39.36	0.9785
	24 × 24	38.38	0.9621	34.13	**0.9228**	32.44	0.9030	33.51	0.9405	39.49	0.9788
XTNSR (Ours)	8 × 8	38.26	0.9616	34.08	0.9215	32.38	0.9020	33.06	0.9368	39.22	0.9782
	16 × 16	38.34	0.9620	34.11	0.9217	32.44	0.9028	33.44	0.9398	39.51	0.9793
	24 × 24	**38.40**	**0.9622**	**34.14**	0.9222	**32.48**	**0.9034**	**33.68**	**0.9409**	**39.86**	0.9796

state-of-the-art methods to compare our proposed approach and show its effectiveness in image denoising. Table 7 shows the comparison of our proposed approach against SOTA methods like BM3D [79], FFDNet [80], NCSR [81], DnCNN [11], Swin IR [26], SRFormer [28], and RDN [45]. The results demonstrate that XTNSR consistently outperforms the other methods regarding peak signal-to-noise ratio (PSNR), with the best values highlighted in bold red and the second-best values underlined in blue. For Kodak24 [82], XTNSR achieves the highest PSNR values across all noise levels, indicating superior performance in preserving image quality while removing noise. Similarly, on the General100 [9] dataset, XTNSR shows the best performance at noise levels $\sigma = 5$ and $\sigma = 10$ and competitive performance at $\sigma = 15$. On the McMaster [83] dataset, XTNSR also

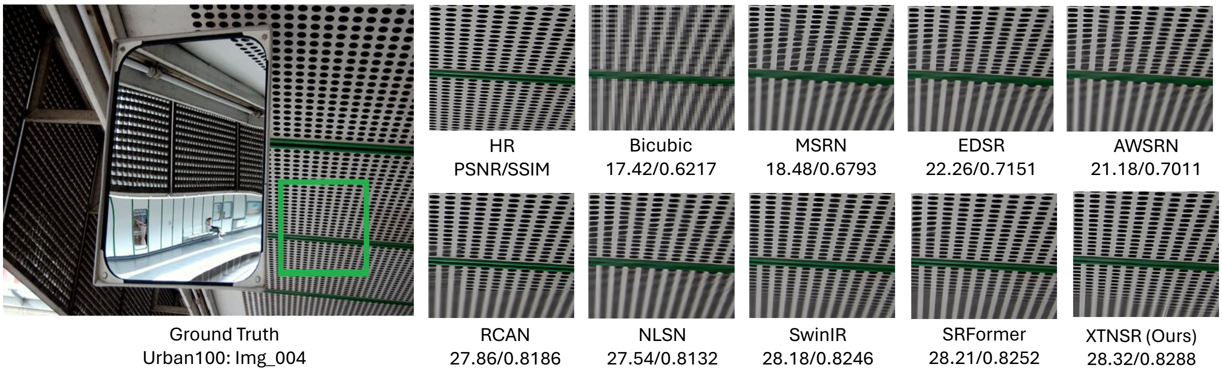

Fig. 11: Img_004 image from the Urban100 [73] dataset perceptual improvement on a ×4 up-scaling factor.

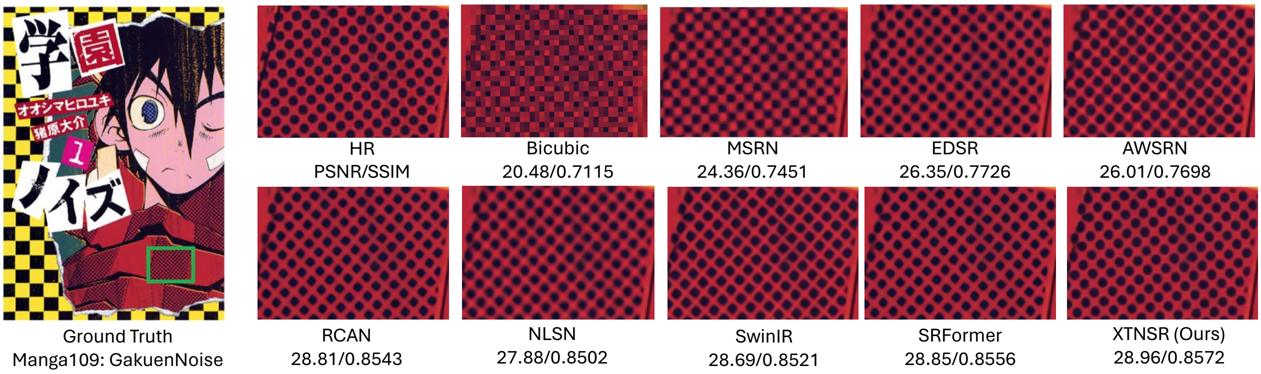

Fig. 12: GakuenNoise image from the Manga109 [74] dataset perceptual improvement on a ×4 up-scaling factor.

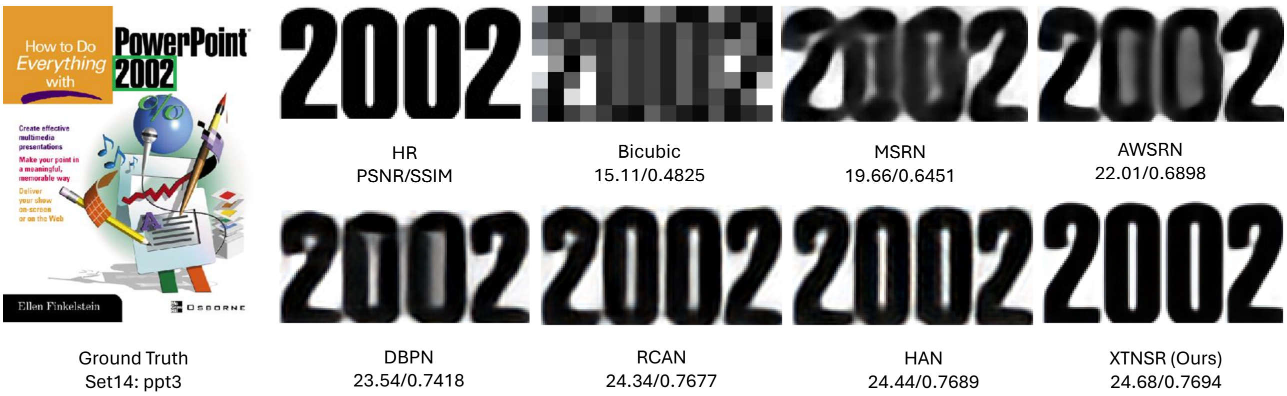

Fig. 13: ppt3 image from the Set14 [71] dataset perceptual improvement on a ×8 up-scaling factor.

Table 6: Performance evaluation for noise degradation of images on Urban100 [73] for scale factor ×2. The best quantitative value has been recorded as bold with **Red** color. The second best quantitative value is shown in blue color with an underline.

Methods / Noise Level	BM3D[79]	FFDNet [80]	NCSR [81]	DnCNN [11]	XTNSR (Our)
$\sigma = 5$	31.18	31.34	31.56	31.67	**31.78**
$\sigma = 10$	29.61	29.76	29.44	29.82	29.89
$\sigma = 15$	28.12	28.48	28.64	28.58	28.77

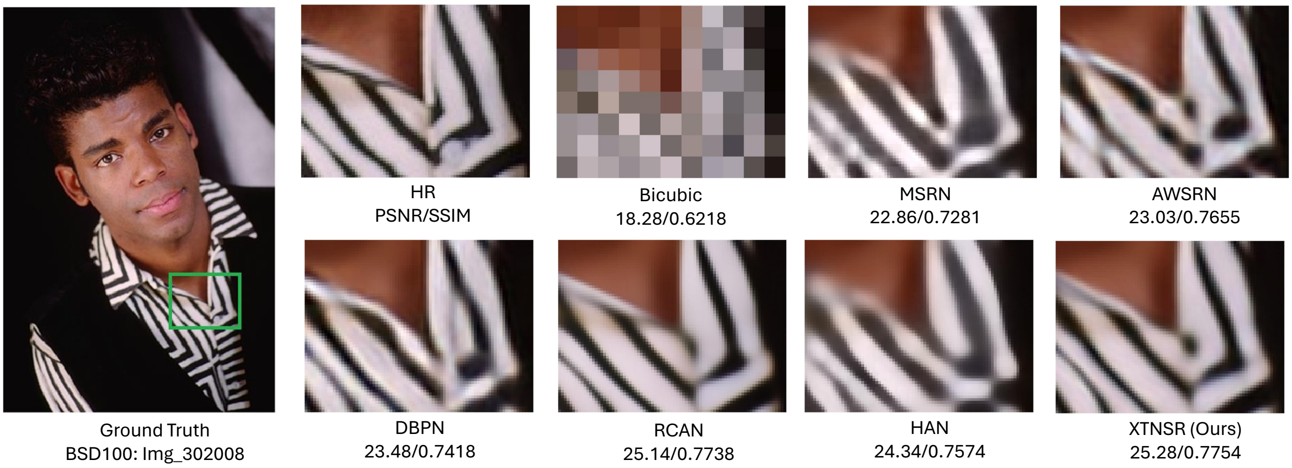

Fig. 14: Img_302008 image from the BSD100 [72] dataset perceptual improvement on a ×8 up-scaling factor.

Fig. 15: Img_096 image from the Urban100 [73] dataset perceptual improvement on a ×8 up-scaling factor.

Fig. 16: MAD_STONE image from the Manga109 [74] dataset perceptual improvement on a ×8 up-scaling factor.

Table 7: Performance evaluation for color image denoising on Kodak24 [82], General100 [9], and McMaster [83] for scale factor ×2. The best quantitative value has been recorded as bold with **Red** color. The second best quantitative value is shown in blue color with an underline.

Dataset	σ	BM3D [79]	FFDNet [80]	NCSR [81]	DnCNN [11]	RDN [45]	SRFormer [28]	SwinIR [26]	XTNSR (Ours)
Kodak24 [82]	5	34.39	34.72	34.74	34.69	34.84	35.54	35.52	**35.56**
	10	34.32	34.68	34.72	34.64	34.78	35.44	35.46	**35.48**
	15	34.28	34.63	34.67	34.60	34.68	**35.38**	35.34	35.36
General100 [9]	5	33.62	33.86	34.58	34.50	34.60	35.47	**35.55**	35.53
	10	33.56	33.83	34.53	34.45	34.57	35.42	**35.52**	35.51
	15	33.52	33.78	34.49	34.41	34.52	35.36	35.48	**35.50**
McMaster [83]	5	34.16	34.76	34.43	33.51	34.48	35.32	35.67	**35.68**
	10	34.12	34.70	34.39	33.48	34.45	35.28	**35.63**	35.61
	15	34.06	34.66	34.35	33.45	34.42	35.22	35.56	**35.58**

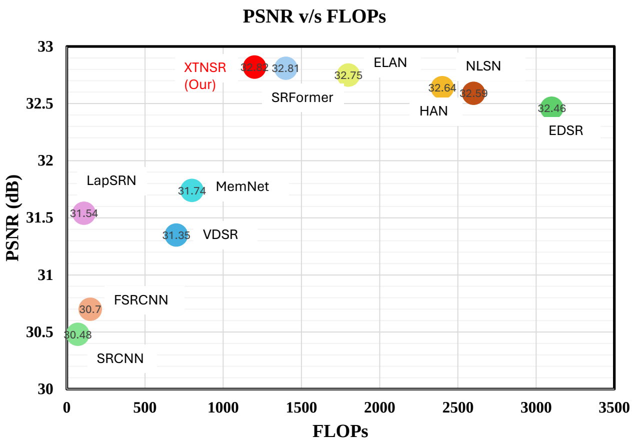

Fig. 17: FLOPs v/s PSNR for Set5 [70] scale ×4.

excels, particularly at $\sigma = 5$ and $\sigma = 10$, reinforcing its robustness across diverse imaging conditions.

These results demonstrates the effectiveness of the XTNSR method in color image denoising, highlighting its ability to outperform existing techniques and handle varying noise levels effectively. The observed improvements suggest that our approach significantly advances image denoising technology, particularly for scenarios involving high noise levels and complex image content.

5.8.5 Ablation assessment using different non-linearity functions in Xception Block

We changed the ReLU activation to PReLU and CReLU, as seen in Figure 18 in the Xception block of the model. The model was trained for 400 epochs and on scale ×2, and the average PSNR and SSIM were calculated on image test datasets. Table 8 shows that the best PSNR and SSIM are obtained with ReLU activation and the second best with CReLU.

Table 8: Assessment of the Xception block using various activations, such as PReLU, CReLU, and ReLU. The average PSNR and SSIM quantitative values are computed using the Image SR test datasets with an up-scaling factor of ×2 across 100 epochs. **Red** indicates the best quantitative value, whereas blue indicates the second-best quantitative value.

Non-Linearity	Xception Block			Average PSNR	Average SSIM
ReLU	✓	×	×	**32.66**	**0.8876**
CReLU	×	✓	×	32.62	0.8872
PReLU	×	×	✓	32.58	0.8868

Figure 19, Figure 20, and Figure 21 show the model's PSNR, SSIM, and Loss convergence with ReLU, CReLU, and PReLU activations, respectively. The values are plotted using the DIV2K [69] training Dataset on the up-scaling factor of ×2 for 100 epochs, keeping other hyper-parameters the same.

Figure 19 and Figure 20 show that ReLU non-linearity gives better PSNR and SSIM training convergence for the model. Figure 21 presents that ReLU gives minimal loss convergence during training. Henceforth, it is better to use ReLU activation for our proposed XTNSR Network.

5.8.6 Ablation assessment using different non-linearity functions in Multi-Layer Feature Fusion (MLFF) Block

We did a similar assessment in the Multi-Layer Feature Fusion (MLFF) Block, just as in the Xception Block. We performed experiments by changing the non-linearity in the Multi-Layer Feature Fusion (MLFF) Block. Figure 22 shows the MLFF blocks with different activation functions. The quantitative values shown in Table 9 reveal that ReLU activation gives the best average PSNR and SSIM values of Image SR test datasets. The second-best value is given by CReLU activation.

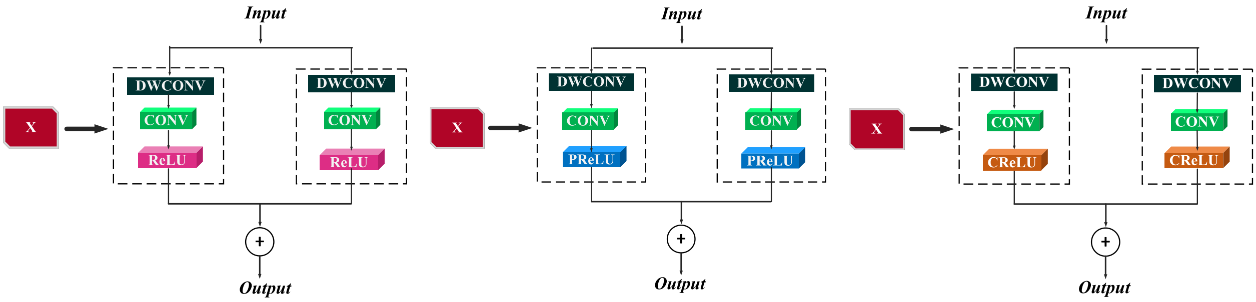

Fig. 18: Xception Block with different non-linearity.

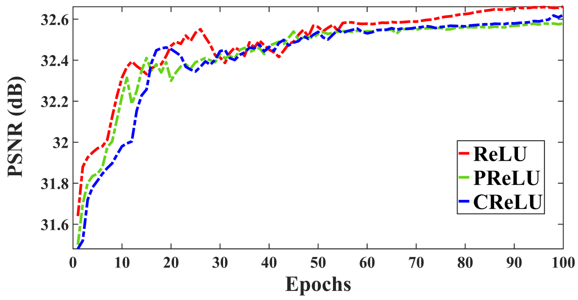

Fig. 19: PSNR convergence using different non-linearity on the DIV2K [69] dataset for up-scaling factor ×2.

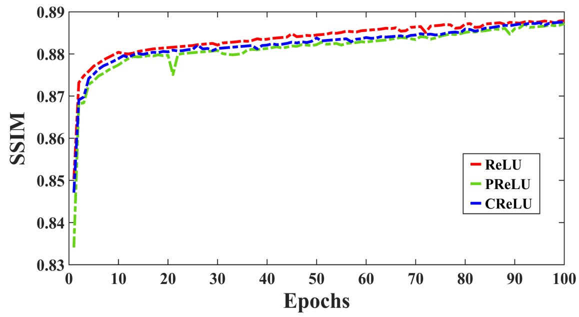

Fig. 21: Loss convergence using different non-linearity for up-scaling factor ×2.

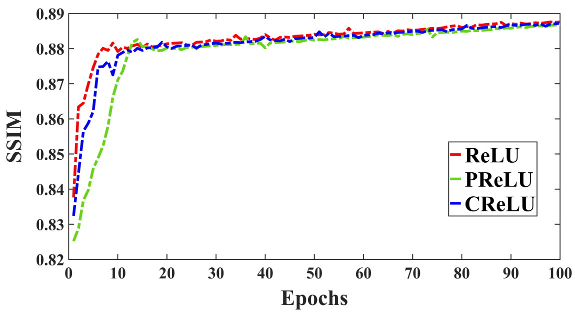

Fig. 20: SSIM convergence using different non-linearity on the DIV2K [69] dataset for up-scaling factor ×2.

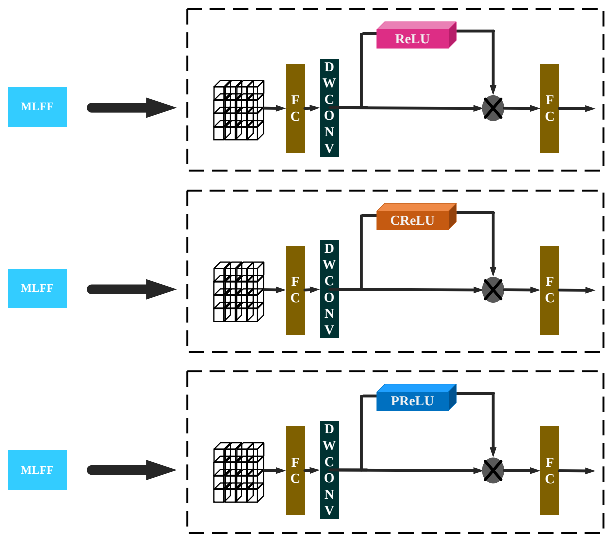

Fig. 22: MLFF Block with different non-linearity.

Figures 23, 24, and 25 also show PSNR, SSIM, and loss convergence for the MLFF Block of the model with different ReLU, CReLU, and PReLU activations. The DIV2K [69] training dataset has been used to plot the curves on the up-scaling factor of ×2 for 100 epochs, keeping other hyper-parameters the same.

Figures 23 and 24 show that ReLU shows better PSNR and SSIM versus epoch convergence than CReLU and PReLU. Figure 25 further shows that ReLU gives better loss convergence, proving that ReLU is the most suitable activation to use in our proposed XTNSR model.

Table 9: Assessment of the MLFF block using various activations, such as PReLU, CReLU, and ReLU. The average PSNR and SSIM quantitative values are computed using the Image SR test datasets with an up-scaling factor of ×2 across 100 epochs. **Red** indicates the best quantitative value, whereas blue indicates the second-best quantitative value.

Non-Linearity	MLFF Block			Average PSNR	Average SSIM
ReLU	✓	×	×	**32.68**	**0.8879**
CReLU	×	✓	×	<u>32.65</u>	<u>0.8876</u>
PReLU	×	×	✓	32.60	0.8872

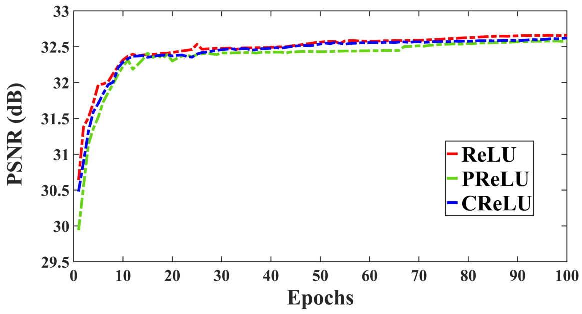

Fig. 23: PSNR versus epoch convergence using different non-linearity on the DIV2K [69] dataset for up-scaling factor ×2.

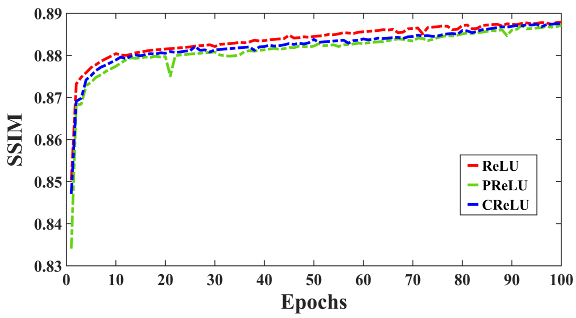

Fig. 24: SSIM versus epoch convergence using different non-linearity on the DIV2K [69] dataset for up-scaling factor ×2.

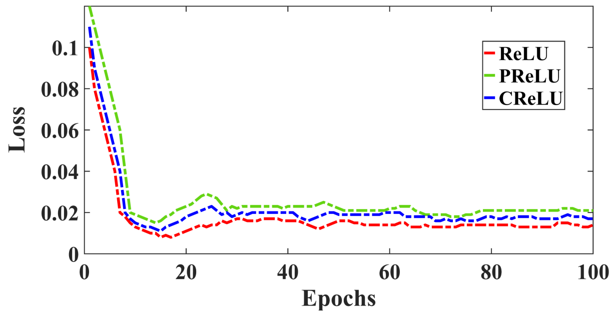

Fig. 25: Loss versus epoch convergence using different non-linearity for up-scaling factor ×2.

6 Discussion

Significant improvements in image super-resolution tasks are demonstrated by the proposed Xception-Based Transformer Network for Single Image Super-Resolution (XTNSR), which introduces a novel architecture that combines Xception Blocks and Local Feature Window Transformer (LFWT) Blocks within a multi-path framework. Based on various benchmark datasets, the experiments' results show that XTNSR performs better than several cutting-edge techniques in terms of PSNR, SSIM, and visual quality.

Specificity of Proposed Method: Although the architecture of XTNSR exhibits encouraging results, certain scenarios are more conducive to its effectiveness. With their ability to capture long-range dependencies, the suggested LFWT Blocks are very helpful for images with intricate textures and patterns, like those found in the Urban100 and Manga109 datasets. Though still competitive, the method's performance is less noticeable in smoother regions, like those found in simpler images from the Set5 and Set14 datasets. The combination of LFWT and Xception Blocks allows the network to balance local and global feature extraction, effectively reducing over-smoothing artifacts and improving the recovery of fine details in noisy images.

Better quantitative and qualitative results are observed in scenarios where very high-resolution grid-like image data require significant computational resources despite certain limitations. The approach demonstrates notable improvements in image quality and preservation of detail, especially when working with large-scale factors (e.g., ×4 and ×8). This indicates that the technique is particularly well-suited for settings with limited resources or real-time, such as forensic analysis or medical imaging, where a high degree of visual fidelity is essential.

Impact on the ISR Field: By tackling the trade-off between computing efficiency and image quality, the XTNSR architecture marks a substantial breakthrough in the ISR industry. Combining Xception Blocks and LFWT opens the door to more complex models that can adjust to different image complexity levels without sacrificing efficiency. This approach not only expands on the current knowledge of transformer applications in ISR but also creates new opportunities for future research aimed at producing even more compact and effective models for real-time use. Furthermore, the architecture's ability to handle both local and global features with flexibility implies that it could be useful for tasks related to image restoration other than ISR, like deblurring or denoising.

Overall, the suggested XTNSR approach offers both technological innovation and useful implications for high-resolution imaging tasks in various fields, significantly contributing to the evolution of deep learning-based ISR methods. This method's applicability in domains requiring high-precision image restoration may be enhanced by further investigation and refinement.

7 Conclusions and Future Work

In conclusion, our work presents the XTNSR model, which combines novel Local Feature Window Transformers with Xception blocks for single-image super-resolution. We effectively handle patches to preserve computational efficiency while guaranteeing accuracy using a Patch Embedding layer. We efficiently balance local and global information by combining the LWFT and Xception blocks in a multi-path network backbone. Thus, mitigating over-smoothing and artifact generation. This method also helps to recover noisy images by capturing the hierarchical feature through integrating the Xception Block in the network. The model's efficacy is demonstrated across various up-scaling factors by evaluating five benchmark datasets. A detailed analysis of five benchmark datasets reveals that the suggested XTNSR approach also improves restoration effects concerning both statistical and subjective requirements for up-scaling factors of $\times 2$, $\times 3$, $\times 4$, and $\times 8$.

The XTNSR model's performance demonstrates a notable advancement in single-image super-resolution compared to previous methods. By combining LWFT and Xception blocks, our approach enhances local detail preservation and global consistency, addressing common issues observed in earlier models such as Swin IR and SRFormer. This combination proves effective in reducing artifacts and improving overall image quality, aligning with recent advancements in transformers where similar techniques were shown to be beneficial. Future research will refine the model for real-time and high-definition video applications.

8 Limitations

Even though our proposed XTNSR model exhibits promising results, several limitations are needed to be acknowledged:

Training Data Dependency: The model's effectiveness is contingent on the quality and diversity of the training dataset. The current model may not generalize well to images with characteristics significantly different from those in the DIV2K dataset.

Real-Time Applications: Although the XTNSR model performs well in static image scenarios, further refinement is needed to optimize the model for real-time and high-definition video applications.

Scalability: The model has been tested effectively up to $\times 8$ scaling factors. However, its performance for even higher scaling factors (e.g., $\times 16$) remains unexplored.

Acknowledgements This research is funded by the Second Century Fund (C2F), and the Ratchadapiseksompotch Fund Chulalongkorn University, Bangkok, 10330, Thailand. This reasearch is also supported by the 90th Anniversary of Chulalongkorn University Scholarship under the Ratchadapisek Somphot Endowment Fund and also funded by the Thailand Science Research and Innovation Fund Chulalongkorn University.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. Greenspan, H., *Super-resolution in medical imaging.* The computer journal, 2009. 52(1): p. 43-63
2. Lorch, B. and Riess, C., *Image forensics from chroma subsampling of high-quality JPEG images.* In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. July, 2019. pp. 101-106.
3. Zhang, Z., Barbary, K., Nothaft, F.A., Sparks, E.R., Zahn, O., Franklin, M.J., Patterson, D.A. and Perlmutter, S., *Kira: Processing astronomy imagery using big data technology.* IEEE Transactions on Big Data, 2016. 6(2): pp.369-381.
4. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation.* in International Conference on Medical image computing and computer-assisted intervention. 2015. Springer.
5. Zhang, L., et al., *A super-resolution reconstruction algorithm for surveillance images.* Signal Processing, 2010. 90(3): p. 848-859.
6. Olsen, S.I., *Estimation of noise in images: An evaluation.* CVGIP: Graphical Models and Image Processing, 1993. 55(4): p.319-323.
7. A. Basharat, A. Gritai, and M. Shah, *"Learning object motion patterns for anomaly detection and improved object detection,"* IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587510.
8. C. Dong, C. C. Loy, K. He and X. Tang, *"Image super-resolution using deep convolutional networks"*, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, pp. 295-307, Feb. 2015.
9. Dong, C., C.C. Loy, and X. Tang. *Accelerating the super-resolution convolutional neural network. in European conference on computer vision.* 2016. Springer.

10. J. Kim, J. K. Lee and K. M. Lee, *"Accurate image super-resolution using very deep convolutional networks"*, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1646-1654, Jun, 2016.
11. Chen, J. and Li, F., *Denoising convolutional neural network with mask for salt and pepper noise.* IET Image Processing, 2019. 13(13), pp.2604-2613.
12. W.-S. Lai, J.-B. Huang, N. Ahuja and M.-H. Yang, *"Deep Laplacian pyramid networks for fast and accurate super-resolution"*, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5835-5843, Jul. 2017.
13. Kim, J., J.K. Lee, and K.M. Lee. *Deeply-recursive convolutional network for image super-resolution.* in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016.
14. Tai, Y., Yang, J. and Liu, X., *Image super-resolution via deep recursive residual network.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. (pp. 3147-3155).
15. Y. Tai, J. Yang, X. Liu and C. Xu, *"MemNet: A persistent memory network for image restoration"*, *Proc. IEEE Conf. Int. Conf. Comput. Vis.*, Oct 2017. pp. 4539-4547.
16. Ahn, N., Kang, B. and Sohn, K.A., *Fast, accurate, and lightweight super-resolution with cascading residual network.* In Proceedings of the European conference on computer vision (ECCV), 2018. (pp. 252-268).
17. Lim, B., *Enhanced deep residual networks for single image super-resolution.* in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 2017.
18. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang et al., *Photorealistic single image super-resolution using a generative adversarial network* in CVPR, 2017.
19. M. S. Sajjadi, B. Scholkopf, and M. Hirsch, *Enhancenet: Single image super-resolution through automated texture synthesis* in ICCV, 2017.
20. X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, *Esrgan: Enhanced super-resolution generative adversarial networks* in ECCV Workshop, 2018.
21. Zhang, Y., et al. *Image super-resolution using very deep residual channel attention networks.* in *Proceedings of the European conference on computer vision (ECCV).* 2018.
22. Mei, Y., et al. *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.* 2020.
23. Mei, Y., Y. Fan, and Y. Zhou. *Image super-resolution with non-local sparse attention.* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021.
24. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X. and Shen, H., 2020. *Single image super-resolution via a holistic attention network.* In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16* (pp. 191-207). Springer International Publishing.
25. Zhang, X., Zeng, H., Guo, S. and Zhang, L., *Efficient long-range attention network for image super-resolution.* In European Conference on Computer Vision (pp. 649-667). Cham: Springer Nature Switzerland, Oct, 2022.
26. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L. and Timofte, R. *SwinIR: Image restoration using swin transformer.* In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. (pp. 1833-1844).
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. *Swin Transformer: Hierarchical vision transformer using shifted windows.* In Proceedings of the IEEE/CVF international conference on computer vision, 2021. (pp. 10012-10022).
28. Zhou, Y., Li, Z., Guo, C.L., Bai, S., Cheng, M.M. and Hou, Q. *SRFormer: Permuted Self-Attention for Single Image Super-Resolution.* arXiv preprint, 2023. arXiv:2303.09735.
29. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., and Houlsby N. *An image is 11worth 16×16 words: Transformers for image recognition at scale.* In International Conference on Learning Representations, 2021.
30. Chollet, F. *Xception: Deep learning with depthwise separable convolutions.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1251-1258). 2017.
31. Mehri, A., Ardakani, P.B. and Sappa, A.D. *MPRNet: Multi-path residual network for lightweight image super-resolution.* In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021. pp. 2704-2713.
32. Tong, T., Li, G., Liu, X. and Gao, Q. *Image super-resolution using dense skip connections.* In Proceedings of the IEEE International Conference on Computer Vision, 2017. pp. 4799-4807.
33. Trockman, A. and Kolter, J.Z. *Patches are all you need?.* 2022. arXiv preprint arXiv:2201.09792.
34. Chun-Le G., Yan Q., Anwar S., Cong R., Ren W., and Li C. *Image dehazing transformer with transmission-aware 3d position embedding.* In CVPR, 2022.
35. Hui, Z., Gao, X., Yang, Y. and Wang, X., *Lightweight image super-resolution with information multi-distillation network.* In *Proceedings of the 27th ACM international conference on multimedia*, October 2019. (pp. 2024-2032).
36. Li, Z., et al. *Feedback network for image super-resolution.* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2019.
37. K. Zhang, W. Zuo and L. Zhang, *"Learning a single convolutional super-resolution network for multiple degradations"*, *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018. vol. 6, no. 1, pp. 3262-3271.
38. K.-W. Hung, Z. Zhang and J. Jiang, *"Real-time image super-resolution using recursive depthwise separable convolution network"*, *IEEE Access*, vol. 7, 2019. pp. 99804-99816.
39. X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, et al., *"DRFN: Deep recurrent fusion network for single-image super-resolution with large factors"*, *IEEE Trans. Multimedia*, Feb. 2019. vol. 21, no. 2, pp. 328-337.
40. Wazir M., Aramvith S., and Onoye T. *"SENext: Squeeze-and-ExcitationNext for Single Image Super-Resolution."* IEEE Access (2023).
41. Ozdemir, C. *Adapting transfer learning models to dataset through pruning and Avg-TopK pooling.* *Neural Computing and Applications*, 2024. 36(11), pp.6257-6270.
42. Ozdemir, C., Dogan, Y. and Kaya, Y. *A new local pooling approach for convolutional neural network: local binary pattern.* *Multimedia Tools and Applications*, 2024. 83(12), pp.34137-34151.
43. Z. Wang, J. Chen, and S. C. H. Hoi, *'Deep learning for image superresolution: A survey.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, Oct. 2021. pp. 3365–3387.

44. Ma, C., Rao, Y., Cheng, Y., Chen, C., Lu, J. and Zhou, J. *Structure-preserving super resolution with gradient guidance.* In Proceedings of the *IEEE/CVF conference on computer vision and pattern recognition*, 2020. (pp. 7769-7778).
45. Zhang, Y., et al. *Residual dense network for image super-resolution.* in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018.
46. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X. and Tang, X. *Residual attention network for image classification.* In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. pp. 3156-3164.
47. Ruangsang W., Aramvith S., and Onoye T. "Multi-FusNet of Cross Channel Network for Image Super-Resolution." IEEE Access (2023).
48. Hong, J., Lee, B., Ko, K. and Ko, H. *Fast non-local attention network for light super-resolution.* Journal of Visual Communication and Image Representation, 2023. p.103861.
49. Dai, T., et al. *Second-order attention network for single image super-resolution.* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2019.
50. Liu, J., et al. *Residual feature aggregation network for image super-resolution.* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020.
51. Prajapati, K., Chudasama, V., Upla, K., Raia, K., Ramachandra, R. and Busch, C. *Channel Split Convolutional Neural Network for Single Image Super-Resolution (CSISR).* In 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), December 2021. pp. 1-8.
52. Park, K., Soh, J.W. and Cho, N.I. *Dynamic residual self-attention network for lightweight single image super-resolution.* IEEE Transactions on Multimedia, 2021.
53. Zhang, Y., Wei, D., Qin, C., Wang, H., Pfister, H. and Fu, Y. *Context reasoning attention network for image super-resolution.* In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. pp. 4278-4287.
54. Li, Z., Li, G., Li, T., Liu, S. and Gao, W. *Information-growth attention network for image super-resolution.* In Proceedings of the 29th ACM International Conference on Multimedia. October, 2021. pp. 544-552.
55. Wang, X., Girshick, R., Gupta, A. and He, K. *Non-local neural networks.* In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. (pp. 7794-7803).
56. Talreja, J., Aramvith, S. and Onoye, T. *DANS: Deep Attention Network for Single Image Super-Resolution.* IEEE Access, 2023.
57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I.*Attention is all you need.* Advances in neural information processing systems, 2017. 30.
58. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J. *Transformers: State-of-the-art natural language processing.* In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, October 2020. pp. 38-45.
59. J. Talreja, S. Aramvith and T. Onoye, "*DHTCUN: Deep Hybrid Transformer CNN U Network for Single-Image Super-Resolution*," in IEEE Access, vol. 12, pp. 122624-122641, 2024,
60. Dong, X., Shi, P., Liang, T. and Yang, A., 2024. CTAFFNet: CNN–Transformer Adaptive Feature Fusion Object Detection Algorithm for Complex Traffic Scenarios. Transportation Research Record, p.03611981241258753.
61. Dong, X., Shi, P., Qi, H., Yang, A. and Liang, T., 2024. TS-BEV: BEV object detection algorithm based on temporal-spatial feature fusion. Displays, 84, p.102814.
62. Malczewski, K. and Stasiński, R. *Super-resolution for multimedia, image, and video processing applications.* In Recent Advances in Multimedia Signal Processing and Communications Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. pp. 171-208.
63. Chen, Y., Li, Q., Zhang, A., Zou, L., Jiang, Y., Xu, Z., Li, J. and Yuan, Z. *Higher quality live streaming under lower uplink bandwidth: an approach of super-resolution based video coding.* In Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, July 2021. pp. 74-81.
64. Isaac, J.S. and Kulkarni, R. *Super-resolution techniques for medical image processing.* In 2015 International Conference on Technologies for Sustainable Development (ICTSD) IEEE, Feb 2015. pp. 1-6.
65. Kong, Y., Ren, X. and Zhou, R. *Deep learning for image super-resolution and its application in urban security.* In 2nd International Conference on Applied Mathematics, Modelling, and Intelligent Computing (CAMMIC 2022) (Vol. 12259, pp. 1459-1464). SPIE.
66. Zamfir, E., Conde, M.V. and Timofte, R. *Towards real-time 4k image super-resolution.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023. pp. 1522-1532.
67. Taud, H. and Mas, J.F. *Multilayer perceptron (MLP).* Geomatic approaches for modeling land change scenarios, 2018. pp.451-455.
68. Arora, R., Basu, A., Mianjy, P. and Mukherjee, A. *Understanding deep neural networks with rectified linear units.* arXiv preprint, 2016. arXiv:1611.01491.
69. Agustsson, E., Timofte, R. *"NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study."* IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2017). 126-135.
70. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. *Low-complexity single-image super-resolution based on nonnegative neighbor embedding.* In *Proceedings of the British Machine Vision Conference, Surrey*, UK, 3–7 September 2012.
71. Zeyde, R.; Elad, M.; Protter, M. *On Single Image Scale-Up Using Sparse-Representations.* In *Proceedings of the International Conference on Curves and Surfaces*, Oslo, Norway, 28 June–3 July 2012; pp. 711–730.
72. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.* In *Proceedings of the Eighth International Conference On Computer Vision* (ICCV-01), Vancouver, BC, Canada, 7–14 July 2001.
73. Huang, J.-B.; Singh, A.; Ahuja, N. *Single image super-resolution from transformed self-exemplars.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.
74. Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. *Sketch-based manga retrieval using manga109 dataset.* Multimedia. Tools Appl. 2017, 76, 21811–21838.
75. Ozdemir, C., Dogan, Y. and Kaya, Y. *RGB-Angle-Wheel: A new data augmentation method for deep learning models. Knowledge-Based Systems*, 2024. 291, p.111615.

76. J. Li, F. Fang, K. Mei and G. Zhang, *"Multi-scale residual network for image super-resolution", Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep 2018. pp. 517-532.
77. Wang, C., Li, Z. and Shi, J., *Lightweight image super-resolution with adaptive weighted learning network*, 2019. arXiv preprint arXiv:1904.02
78. Haris, M., Shakhnarovich, G. and Ukita, N. *Deep back-projection networks for super-resolution.* In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. pp. 1664-1673.
79. Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. *Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space.* In 2007 IEEE international conference on image processing, September 2007. (Vol. 1, pp. I-313). IEEE.
80. Zhang, K., Zuo, W. and Zhang, L., 2018. *FFDNet: Toward a fast and flexible solution for CNN-based image denoising.* IEEE Transactions on Image Processing, 27(9), pp.4608-4622.
81. Dong, W., Zhang, L., Shi, G. and Li, X. *Nonlocally centralized sparse representation for image restoration.* IEEE Transactions on Image Processing, 22(4), 2012. pp.1620-1630.
82. Rich Franzen. Kodak lossless true color image suite. source: http://r0k. us/graphics/kodak, 4(2), 1999.
83. L. Zhang, X. Wu, A. Buades, and X. Li, "Color Demosaicking by Local Directional Interpolation and Non-local Adaptive Thresholding," Journal of Electronic Imaging 20(2), 023016 (Apr-Jun 2011), DOI:10.1117/1.3600632.