
S25 11785 Group 49 Project Proposal:

Image Super Resolution

Nisheet Das
Heinz College of Information Systems
and Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
nisheetd@andrew.cmu.edu

Vishakha Pathak
Heinz College of Information Systems
and Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
vmpathak@andrew.cmu.edu

Rishabh Devgon
Heinz College of Information Systems
and Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
rishabhd@andrew.cmu.edu

Vihan P. Tyagi
Heinz College of Information Systems
and Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
vihant@andrew.cmu.edu

1 Introduction

1.1 Abstract

Single Image Super-Resolution (SISR) aims to reconstruct high-resolution images from low-resolution inputs. While deep convolutional networks excel at capturing local patterns, they often miss global context and fine textures. Transformer-based methods can model long-range dependencies but tend to over-smooth high-frequency details and demand large computational resources. In this work, we propose an Efficient Mixed Transformer for SISR that balances local and global feature extraction, preserves sharp edges and textures, and remains lightweight enough for real-world deployment. We will also explore hyperparameter settings and iterative architecture refinements to push performance beyond existing baselines.

1.2 Motivation

High-frequency details—sharp edges, fine textures, and small structural elements—are critical in applications ranging from medical imaging to satellite remote sensing. Traditional CNN architectures struggle to reconstruct these features when the receptive field is limited, while pure Transformer models, although powerful in capturing global context, often produce overly smooth outputs and incur high memory costs. A hybrid approach that carefully combines convolutional feature extractors with a more efficient self-attention mechanism can close this gap and deliver both fidelity and speed.

1.3 Objective

Our goal is to design and implement a novel mixed Transformer architecture that:

- Integrates lightweight CNN backbones with adaptive Transformer blocks to capture both local texture and global structure.
- Incorporates specialized high-frequency filtering modules to retain sharp details without incurring excessive parameters.
- Leverages an efficient multi-head attention scheme to reduce GPU memory footprint.
- Systematically tunes hyperparameters and refines network components to maximize PSNR/SSIM on standard benchmarks (Set5, Set14, BSD100, Urban100, Manga109) while keeping the model compact.

2 Related Work

Single-image super-resolution has come a long way. Early deep-learning approaches relied on convolutional networks to map local low-resolution (LR) patches to high-resolution (HR) outputs, but they often missed the bigger picture—literally—when it came to capturing long-range image context. More recent Transformer-inspired designs bring self-attention to the table, and with it, a powerful way to model global dependencies.

2.1 Efficient Super-Resolution Transformer (ESRT)

In 2022, Lu et al. (2022) introduced ESRT, a thoughtful hybrid of a lightweight CNN backbone (LCB) and an equally lightweight Transformer backbone (LTB). In the LCB, High-Preserving Blocks (HPBs) equipped with High-frequency Filtering Modules (HFMs) and Adaptive Residual Feature Blocks (ARFBs) keep delicate textures intact. Meanwhile, the LTB’s Efficient Multi-Head Attention (EMHA) uses a clever feature-splitting trick to slash GPU memory usage to about 4.2 GB and pare parameters down to 751 K. Despite its compact size, ESRT hits 32.19 dB PSNR on Set5 and 28.69 dB on Set14 ($\times 4$), and it leads the pack on Urban100 and real-world benchmarks [2].

2.2 Real-Time 4K Image Super-Resolution

Zamfir et al. (2023) tackled the challenge of true real-time 4K upscaling (> 60 FPS) on commodity GPUs. Their RT4KSR framework separates out high-frequency detail via a Gaussian-blur branch, compresses features with PixelUnshuffle, then fuses everything back together with a structurally re-parameterized network. This two-branch design—High-Frequency Branch (HFB) plus Deep Feature Extraction Branch (DFEB)—delivers strong PSNR/SSIM on NTIRE 2023 4K RTSR, Set5, and Urban100, with scalable “S” and “XL” variants for different speed-vs-quality trade-offs [1].

2.3 Efficient Mixed Transformer (EMT)

Zheng et al. (2023) took a slightly different tack with their Efficient Mixed Transformer (EMT). They blend parameter-free Pixel Mixer (PM) layers for local feature aggregation with Striped Window Self-Attention (SWSA) for global context. EMT’s three-stage pipeline (Shallow Feature Extraction Unit, Deep Feature Extraction Unit with stacked MTBs, and a final Reconstruction Unit) hits 32.64 dB on Set5 and 28.97 dB on Set14 ($\times 4$), all while keeping both parameters and memory use below those of SwinIR and ELAN-light [4].

2.4 Surveys on GAN- and Transformer-based SISR

Tian et al. (2024) recently surveyed the landscape of GAN-based, CNN-based, and Transformer-based SISR. They highlight ongoing challenges—preserving high-frequency details, achieving perceptual quality, and balancing computational load—that continue to drive new innovation [3].

3 Methodology

3.1 Model Architecture

ESRT combines a Lightweight CNN Backbone (LCB) with a Lightweight Transformer Backbone (LTB) to capture both local and long-range dependencies in images. LCB utilizes High-Preserving Blocks (HPBs) to extract deep features with reduced computational cost. HPBs incorporate a High-frequency Filtering Module (HFM) for texture detail preservation and Adaptive Residual Feature Blocks (ARFBs) for efficient feature extraction.

Here is what the architecture of HPBs looks like:

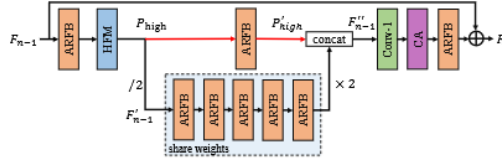


Figure 1: HPB Architecture

Here is what the architecture of an AFRB looks like:

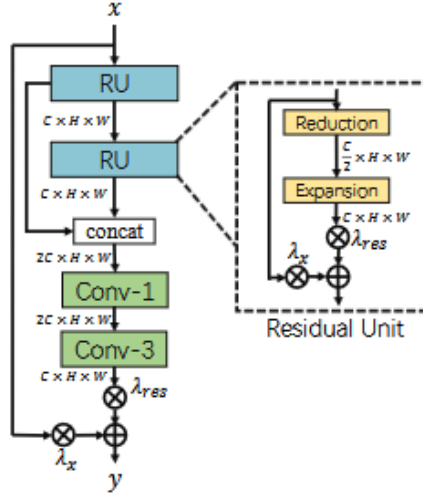


Figure 2: AFRB Architecture

LTB introduces Efficient Transformers (ETs) with an Efficient Multi-Head Attention (EMHA) mechanism to reduce GPU memory usage by focusing on local regions through a novel feature-splitting approach.

Here is the architecture of the LTB:

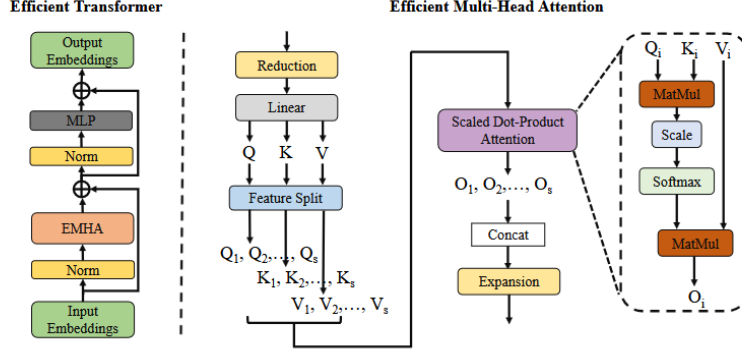


Figure 3: LTB Architecture

3.2 Model Description

Layer Structure	Parameters / Hyperparameters
- Shallow Conv (3×3 kernel, 32 channels) - LCB (3× High-Preserving Blocks): Each HPB contains a High-Frequency Filtering Module (downsamples ×2) and 5 stacked ARFBs (shared weights) - LTB (1× Efficient Transformer): ET (MLABlock): 1 block in LTB, with 8 heads, patch size 3×3, split factor 4 - Conv (3×3,32) + PixelShuffle upsample: All convs use 3×3 kernels with 32 filters (fusion conv uses 64 filters)	0.77M parameters - Input/Output: 3-channel LR image patch (3×48×48 for ×4 SR) → 3-channel HR output (3×192×192 for ×4) - Training: DIV2K dataset with random 48×48 LR patches, augmentation (flip/rot) - Optimizer: Adam ($\beta_1=0.9$) with initial LR=2e-4 (halved every 200 epochs) - Batch size: 16, Loss: L1, Epochs: 1000

Table 1: Model Description

3.3 Dataset

For training we used DIV2K noa (a) – a set of high-quality 2K images. For validation and testing, we leveraged the Set5 noa (b), Set14 noa (b), BSD100 noa (2021), Manga109 noa (2021), and Urban100 noa (c) benchmarks, which span natural scenes, textured photographs, manga illustrations, and urban landscapes. This rigorous evaluation protocol, together with DF2K’s high-fidelity ground truths, ensures strong generalization across diverse image domains. During training, we applied on-the-fly augmentations (random horizontal/vertical flips and 90° rotations) and normalized all pixel intensities to the $[0, 1]$ range. Data were loaded via PyTorch’s DataLoader with batch_size=16 (shuffle=True, drop_last=True, num_workers=4, pin_memory=True) during training, then with batch_size=1 (shuffle=False, drop_last=False, num_workers=2) for validation and testing to guarantee deterministic metrics.

3.4 Evaluation metrics

The PSNR Sara et al. (2019) is a widely used metric for measuring the quality of reconstructed images or videos after compression. It is calculated using the Mean Squared Error (MSE) between the original image I and the reconstructed image K .

Given two images I and K of size $m \times n$:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right)$$

where MAX_I is the maximum possible pixel value of the image (e.g., 255 for 8-bit images).

SSIM Sara et al. (2019) is designed to model the perceived change in structural information, incorporating luminance, contrast, and structure comparisons between two images x and y .

The general formula for SSIM is:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma$$

where typically $\alpha = \beta = \gamma = 1$

The three components are defined as:

Luminance:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

Contrast:

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

Structure:

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

where:

- μ_x, μ_y are the mean intensities of x and y
- σ_x^2, σ_y^2 are the variances of x and y
- σ_{xy} is the covariance between x and y
- C_1, C_2, C_3 are small constants to stabilize the division

The most common simplified form (with $C_3 = C_2/2$) is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

This value ranges from -1 to 1 , where 1 indicates perfect similarity.

3.5 Loss

We used the L1 loss in our implementation. This decision is based on the fact that L1 loss tends to preserve high-frequency details better than L2 loss, resulting in images with sharper edges and textures. Since L2 loss heavily penalizes mistakes, we could potentially risk the model generating blurry images due to the presence of outliers. L1 loss is less sensitive to outliers allowing our model to preserve more details.

4 Baseline

4.1 Baseline Selection

As our primary baseline, we adopt the Efficient Super-Resolution Transformer (ESRT) framework introduced by Zhisheng Lu et al. in “Transformer for Single Image Super-Resolution”[2]. This model establishes a hybrid CNN + Vision-Transformer architecture for SISR and demonstrates strong PSNR/SSIM performance making it a strong choice. One pitfall however is that it requires 16000 MB of GPU memory during training.

4.2 Baseline Model Description

The baseline consists of:

Lightweight CNN Backbone (LCB): Uses High Preserving Blocks (HPB) equipped with High-frequency Filtering Modules (HFM) and Adaptive Residual Feature Blocks (ARFB) to efficiently extract deep features with minimal computation.

Lightweight Transformer Backbone (LTB): Composed of Efficient Transformers (ET) that employ an unfolding–folding pre/post-processing strategy and an Efficient Multi-Head Attention (EMHA) mechanism with local-window splitting to capture long-range patch dependencies while reducing attention matrix size.

In a $\times 4$ upscaling setting, this model achieves PSNR = 32.14dB on Set5 with 971 K parameters

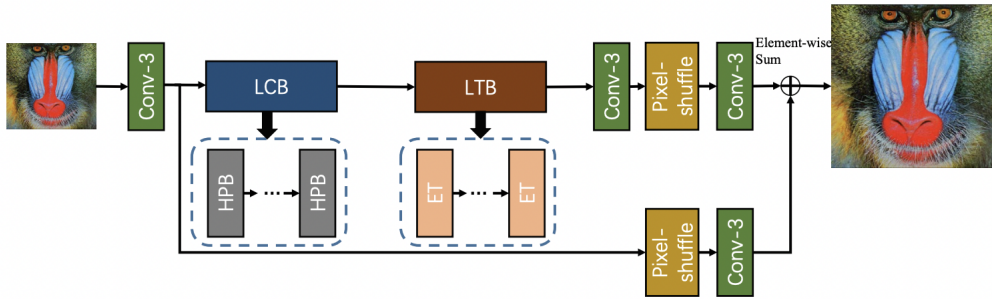


Figure 4: ESRT Architecture

4.3 Baseline Implementation Completeness

To demonstrate the fidelity of our baseline, we compare its super-resolved outputs against the original high-resolution ground-truth images and the low-resolution images.



Figure 5: SET 5: HR Image, Model Output, LR Image



Figure 6: SET 14: HR Image, Model Output, LR Image



Figure 7: Manga 109: HR Image, Model Output, LR Image



Figure 8: BSD 100: HR Image, Model Output, LR Image

4.4 Proposed Model Extensions

Building on the ESRT baseline, we plan to implement and evaluate the following extensions to further improve reconstruction quality and efficiency:

- **Multi-Scale High-Frequency Filtering.** Instead of a single downsampling factor in the HFM, integrate parallel HF modules operating at different spatial scales (e.g. $2\times$, $4\times$). This will allow the network to preserve both fine and coarser texture details simultaneously.
- **Adaptive Attention Windowing.** Extend the EMHA mechanism with a dynamic split factor: at each layer, compute a local “complexity” score for each patch (e.g. based on gradient magnitude) and adjust the number of splits so that high-detail regions receive finer attention (more, smaller windows) while smooth regions use fewer, larger windows.
- **Grouped High-Preserving Blocks.** Organize HPBs into residual groups (similar to RCAN’s residual groups) with a shared set of weights per group. This reduces overall parameter count while retaining depth in each group, encouraging cross-block feature reuse.
- **Perceptual and Adversarial Losses.** Add a VGG-based perceptual loss term to the training objective, encouraging the model to match deep feature statistics of HR images. Optionally explore a lightweight GAN discriminator to enhance realism in highly textured regions.
- **Knowledge Distillation.** Train a small “student” version of the extended ESRT by distilling both feature-level and output-level knowledge from the full model. This can yield a faster inference model with minimal quality degradation.
- **Dynamic Channel Pruning.** Incorporate an ℓ_1 -based channel importance regularizer during training and prune low-importance channels in convolutional layers and attention projections, achieving further runtime savings.

5 Results

5.1 Final Results

We successfully implemented the ESRT model for Image Super Resolution. Figure 9 shows the train loss, SSIM and PSNR across datasets. Table 2 compares our model against the benchmarks numbers from the paper.

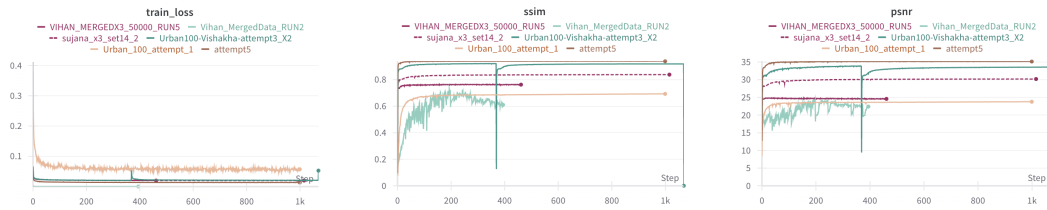


Figure 9: Ablation Plots

5.2 Completeness

We have been successfully able to implement the ESRT model and trained it for getting comparable results as compared to the benchmark numbers.

Eval Dataset	Benchmark SSIM	Our SSIM	Benchmark PSNR	Our PSNR
Set5	0.9268	0.92172	34.42	33.89
Set14	0.8424	0.83849	30.39	30.20
BSD100	0.8063	0.79601	29.15	28.74
Manga109	0.9455	0.84466	33.95	21.18

Table 2: SSIM and PSNR metrics between benchmark and our model across datasets

5.3 Discussion

We believe training on the Div2K dataset was a good initial step. Training the model on Div8K would have led to better generalization given a larger dataset and higher-resolution images. Dividing each image into a greater number of patches would allow the model to learn both local and global dependencies more comprehensively, albeit at the cost of requiring a higher-memory GPU. Although our super-resolution outputs on Manga109 look visually convincing, we were unable to match the benchmark PSNR and SSIM for that dataset. We also experimented with regenerating brain-scan images; the results were not satisfactory, highlighting limitations for medical-imaging applications. Nonetheless, the implementation is robust with respect to input-image resolution and can effectively handle edge cases.

6 Future Work

For our upcoming efforts, please refer to the *Proposed Model Extensions* subsection to see the specific enhancements and directions we plan to pursue next.

In terms of the dataset we plan to use a merged dataset of Urban100, BSD100, and Manga109 to evaluate whether the model generalizes better across diverse image domains and to further improve PSNR and SSIM metrics.

7 Acknowledgments

We would like to acknowledge our Professor Bhiksha Raj for sharing his expertise in the field and enabling us with this opportunity. We would also like to thank our mentor Vishan Oberoi for his mentorship and guidance. His support, insights in formalising and closing the problem statement, and his expertise in deep learning were invaluable in our accomplishments thus far.

References

- Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, “Transformer for Single Image Super-Resolution,” 2022, pp. 457–466. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022W/NTIRE/html/Lu_Transformer_for_Single_Image_Super-Resolution_CVPRW_2022_paper.html
- E. Zamfir, M. V. Conde, and R. Timofte, “Towards Real-Time 4K Image Super-Resolution,” 2023, pp. 1522–1532. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023W/NTIRE/html/Zamfir_Towards_Real-Time_4K_Image_Super-Resolution_CVPRW_2023_paper.html
- L. Zheng, J. Zhu, J. Shi, and S. Weng, “Efficient Mixed Transformer for Single Image Super-Resolution,” Jun. 2023, arXiv:2305.11403 [cs]. [Online]. Available: <http://arxiv.org/abs/2305.11403>
- C. Tian, X. Zhang, Q. Zhu, B. Zhang, and J. C.-W. Lin, “Generative Adversarial Networks for Image Super-Resolution: A Survey,” Dec. 2024, arXiv:2204.13620 [eess]. [Online]. Available: <http://arxiv.org/abs/2204.13620>

“DIV2K Dataset.” [Online]. Available: <https://data.vision.ee.ethz.ch/cvl/DIV2K/>

“Set 5 & 14 Super Resolution Dataset.” [Online]. Available: <https://www.kaggle.com/datasets/ll01dm/set-5-14-super-resolution-dataset>

“eugenesiow/BSD100 · Datasets at Hugging Face,” Jul. 2021. [Online]. Available: <https://huggingface.co/datasets/eugenesiow/BSD100>

“eugenesiow/Urban100 · Datasets at Hugging Face.” [Online]. Available: <https://huggingface.co/datasets/eugenesiow/Urban100>

U. Sara, M. Akter, and M. S. Uddin, “Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, Mar. 2019, number: 3 Publisher: Scientific Research Publishing. [Online]. Available: <https://www.scirp.org/journal/paperinformation?paperid=90911>

Contributions

- **Nisheet Das:**
 - Model training implementation and experiment setup
 - Set5 and Set14 benchmarking
 - Inference generation code
- **Vihan P. Tyagi:**
 - Model training implementation and experiment setup
 - Manga109 benchmarking
 - Dataset pre-processing code.
- **Vishakha Pathak:**
 - Integration and preprocessing of the DF2K training corpus
 - Urban100 benchmarking
 - PSNR/SSIM evaluation and visualization of results
- **Rishabh Devgon:**
 - Hyperparameter tuning and ablation study coordination
 - BSD100 benchmarking
 - Custom validation script

GitHub Repository

Code available at: <https://github.com/nisheetdas1/ESRT>