A Comparative Analysis of Generative, Implicit, and Transformer Architectures for Single Image Super-Resolution

Jack Wayt (bo22354)

December 5, 2025

Abstract

This study presents a comparative analysis of three distinct paradigms for Single Image Super-Resolution (SISR): Generative Adversarial Networks (SRGAN), Implicit Neural Representations (LIIF), and Vision Transformers (SwinIR). Utilizing the DIV2K dataset, the models were evaluated across three experimental conditions: standard $\times 8$ upscaling, robustness to Gaussian noise degradation, and extreme $\times 16$ downscaling. Results indicate that SwinIR achieves state-of-the-art signal fidelity (25.20 dB PSNR), validating the efficacy of self-attention mechanisms in modelling global dependencies. While SRGAN achieved lower fidelity metrics (24.68 dB), it demonstrated superior perceptual quality by hallucinating plausible high-frequency textures absent in regression-based outputs. Notably, the $\times 16$ experiment revealed a critical limitation of coordinate-based networks; LIIF performance collapsed (19.90 dB) due to sparse sampling density, whereas SRGAN maintained structural coherence (21.90 dB) by leveraging convolutional priors. Furthermore, sensitivity analysis showed that while baseline models fail under noise ($\sigma = 50$), a noise-augmented training strategy recovered performance by over 72%, demonstrating the capacity for joint de-noising and upscaling.

The complete implementation and pre-trained models are available at: https://github.com/bo22354/AVAI

1 Introduction

Single Image Super-Resolution (SISR) is the computational task of reconstructing a high-resolution (HR) image from a single low-resolution (LR) observation. As formally described by Glasner et al. [1], this is an inherently ill-posed inverse problem, as the degradation process creates a non-injective mapping where multiple high-resolution scenes could generate the same low-resolution pixel grid. While classical methods such as multi-image super-resolution attempt to combine information from multiple sub-pixel shifted inputs, modern research predominantly focuses on Example-Based (or Learning-Based) super-resolution. This approach, which learns the correspondence between LR and HR patches from large external datasets, has demonstrated superior performance in recovering high-frequency details compared to classical interpolation techniques [2].

In the Example-Based framework, the model aims to learn a mapping function:

$$F : I_{\text{LR}} \rightarrow I_{\text{HR}}$$

By analysing pairs of High and Low-Resolution patches during training, the model learns to predict the missing high-frequency information, such as textures and sharp edges, that was lost during the downsampling process. This process is often referred to as "hallucination," as the model must infer plausible pixel details based on learned priors rather than existing data. Early works by Freeman et al. [2] established that this missing information could be effectively predicted using a database of patch pairs, laying the groundwork for modern deep learning approaches.

Historically, Example-Based SR began with sparse coding techniques [3], which assumed that image patches could be represented by a sparse linear combination of elements from a dictionary. However, the field underwent a paradigm shift with the introduction of Convolutional Neural Networks (CNNs). Dong et al. [4] proposed SRCNN, demonstrating that a deep network could directly learn an end-to-end mapping between LR and HR images, significantly outperforming traditional methods. Despite this success, standard CNNs trained to minimize Mean Squared Error (MSE) often produce perceptually smooth or blurry textures. To address this, Ledig et al. [5] introduced SRGAN, utilizing Generative Adversarial Networks (GANs) to optimize for perceptual similarity rather than pixel fidelity. Most recently, approaches such as Implicit Neural Representations (INRs) have emerged, treating images as continuous functions to achieve arbitrary-scale resolution [6].

2 Selected Methods

2.1 Generative Adversarial Network (GAN)

For the first method, a Generative Adversarial Network (GAN) was selected, specifically adopting the SRGAN framework proposed by Ledig et al. [5]. This architecture is selected to address the limitations of traditional CNN-based super-resolution. As established in the paper, models trained solely minimize Mean Squared Error (MSE), such as SRCNN [4], suffer from a regression to the mean effect. While MSE optimization maximizes Peak Signal-to-Noise Ratio (PSNR), it fails to capture high-frequency details, resulting in perceptually "smooth" or blurry textures. SRGAN is better suited for this task as it incorporates an adversarial loss, forcing the generated output to match the patterns of real world photographs, effectively "hallucinating" realistic textures, rather than producing the blurry results typical of standard MSE-based methods

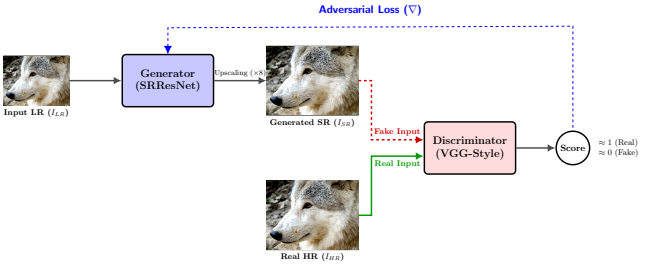

Figure 1: The architecture of the SRGAN

The architecture differs from a typical GAN (e.g., DCGAN) in its conditional nature, rather than generating images from a latent noise vector, the Generator is conditioned on the Low-Resolution input.

For the Generator, this implementation utilizes the SRResNet backbone. This deep residual network is composed of a series of Residual Blocks, each containing valid-padding convolutions, Batch Normalization, and PReLU activation functions with skip connections. The use of residual learning allows the gradient to flow through the deep network without vanishing, enabling the model to learn the high-frequency residual difference between the LR and HR images rather than reconstructing the full image structure from scratch. Furthermore, to upscale the feature maps to the target resolution, the model uses PixelShuffle (sub-pixel convolution) layers. Unlike Transposed Convolutions, which often introduce checkerboard artifacts due to uneven overlap, PixelShuffle efficiently rearranges the channel depth into spatial dimensions ensuring mathematical stability during upsampling [5].

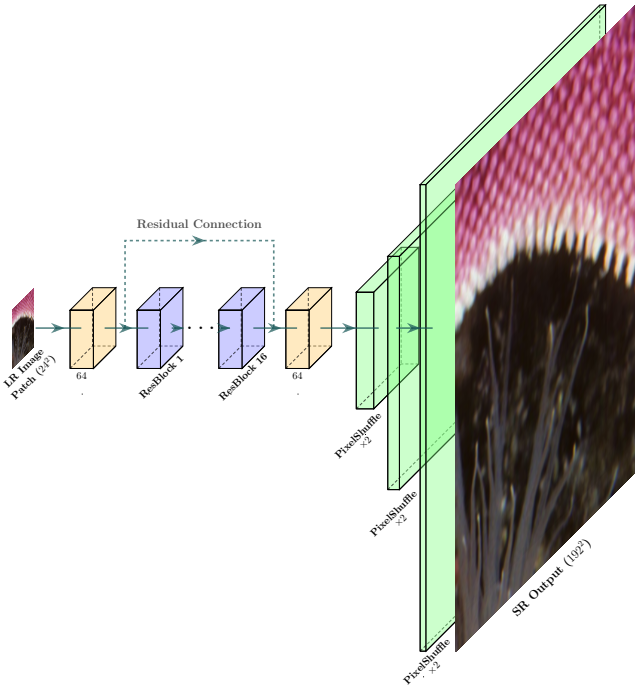

Figure 2: The architecture of the Generator for the SRGAN

$$H \times W \times C \cdot r^2 \rightarrow rH \times rW \times C$$

To ensure architectural flexibility across different scaling factors, the upsampling stage is constructed dynamically. The number of PixelShuffle blocks B is determined by $B = \log_2(S)$, where S is the target scale factor. For the target scale of $\times 8$, the network automatically instantiates three sequential upsampling modules.

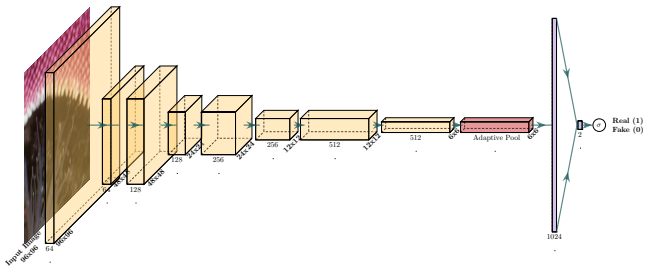

Figure 3: The architecture of the Discriminator for the SRGAN

Similarly, the Discriminator is designed as a deep VGG-style convolutional network. Unlike standard image classifiers that utilize Max Pooling to reduce spatial dimensions, this implementation uses strided convolutions (stride=2) to perform downsampling.

This design choice is critical for super-resolution, Max Pooling discards pixel information to enforce spatial invariance (useful for classifying what an object is), whereas strided convolutions contain learnable parameters that allow the network to preserve specific texture features (useful for determining how real an object looks). The network concludes with an Adaptive Average Pooling layer, dynamically adjusting the feature maps into a fixed spatial dimension, therefore preventing dimension mismatch errors during training on varying patch sizes.

To train this coupled architecture, the network minimizes a composite loss function L_{total}. This function is a weighted sum of a content loss (reconstruction error) and an adversarial loss (perceptual realism):

$$L_{total} = L_{content} + \lambda L_{adv}$$

where λ is a hyperparameter that balances the trade-off between pixel fidelity and perceptual quality

Content Loss (L_{content}) This implementation utilises the pixel-wise L1 Loss to encourage faster convergence and strictly accurate colour reconstruction. It calculates the pixel-wise distance between the ground truth high-resolution image (I^{HR}) and the generated super-resolved image $G(I^{LR})$:

$$L_{content} = \frac{1}{WHC} \sum_{x=1}^{W} \sum_{y=1}^{H} \sum_{C=1}^{C} \left| I^{HR}_{x,y,c} - G(I^{LR})_{x,y,c} \right|$$

Where W, H, C represent the width, height, and channels of the image. Unlike Mean Squared Error (L2), which heavily penalizes large errors and leads to oversmoothing, L1 loss applies a constant gradient, preserving sharper edges during the early stages of training.

Adversarial Loss (L_{adv}) The adversarial component is derived from the GAN objective. It is defined as the negative log-likelihood of the discriminator classifying the generated image as real. We wish to minimize this function (or conversely, maximize the probability $D(G(I^{LR} \approx 1)$:

$$L_{\text{adv}} = -\sum_{n=1}^{N} \log \left(D(G(I_n^{LR})) \right)$$

Where $D(\cdot)$ represents the probability output of the Discriminator. This loss forces the generator to create solutions that reside on the manifold of natural images, effectively penalizing any texture or artifact that the Discriminator can identify as "fake," even if the pixel values are close to the ground truth.

2.2 Implicit Neural Representation (INR)

For the second method, this study implements a Local Implicit Image Function (LIIF), a specific kind of INR based on the framework proposed by Chen et al. [6]. This architecture was chosen to provide a fundamental contrast to the discrete nature of SRGAN. While SRGAN is effective at hallucinating texture, it remains architecturally constrained by fixed upsampling operators, such that to alter the output resolution would typically require structural modification or cascading multiple networks.

LIIF addresses this rigidity by introducing a continuous image representation. Instead of treating super-resolution as a discrete pixel-to-pixel mapping task ($H \times W \rightarrow rH \times rW$), LIIF treats the image as a continuous signal field defined over a 2D domain. The model learns a function $f(z, x) \rightarrow s$, parametrised by a neural network, where z is a latent feature vector extracted from the low-resolution input, x is a continuous 2D coordinate in the image domain ($x \in R^2$), and s is the predicted RGB signal.

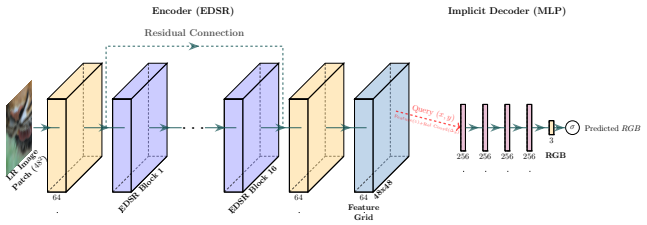

Figure 4: The architecture of the LIIF (INR)

The architecture differs fundamentally from the Image-to-Image translation pipeline of the GAN. Instead, it operates as an Encoder-Decoder system where the input is a Low-Resolution image and a set of continuous 2D coordinates ($x \in R^2$), and the output is the RGB values at those locations.

The encoder is used for the feature extraction stage, for this an EDSR (Enhanced Deep Super-Resolution) style backbone was implemented. Unlike the SRResNet used for SRGAN, the EDSR blocks remove Batch Normalization layers. This modification is critical for implicit representations, as Batch Normalization normalizes features by centring the mean, which destroys the range flexibility required to predict precise colour intensities. Furthermore, to stabilize training across the deep network (16 blocks), Residual Scaling is applied, scaling the residual signal by a factor of 0.1 before adding it to the identity path. This prevents exploding gradients during the initial "cold start" phase of training.

The Implicit Decoder (MLP) The decoding stage

utilizes a Multi-Layer Perceptron (MLP) to predict pixel values. However, passing absolute coordinates to the MLP fails to capture local structures. Therefore, this implementation utilizes a Local Ensemble strategy with Relative Coordinates. For any continuous query point x_q in the HR domain, the model identifies the nearest feature vector z^* in the LR grid and calculates the relative offset distance:

$$\Delta x = x_q - v^*$$

where v^* is the coordinate of the feature's centre. The MLP then takes the concatenation of the feature vector and this relative distance $[z^*, \delta x]$. This forces the network to learn relative interpolation rules (how the image changes as we move away from a feature centre) rather than memorizing absolute positions.

Unlike the GAN, which computes loss over the entire generated image, training a LIIF on full high-resolution images is computationally prohibitive due to the massive pixel count at $\times 8$ scale. Instead, a Sampled Query strategy was chosen. During each training iteration, a random batch of coordinates S (where $|S| = 16384$) is sampled from the ground truth domain.

The network minimizes the pixel-wise L1 Loss over these sampled points. Additionally, a Tanh activation at the final layer of the MLP to constrain the output space to $[-1,1]$, matching the normalized ground truth distribution. The loss function is defined as:

$$L(\theta) = \frac{1}{|S|} \sum_{i \in S} |I^{HR}(x_i) - f(E(I^{LR}), x_i)|$$

Where $I^{HR}(x_i)$ is the ground truth RGB value at coordinate x_i, E is the Encoder, and f is the Implicit Decoder function. By minimizing this loss over random samples, the model learns to approximate the continuous signal of the image, allowing for the reconstruction of fine details at arbitrary scales without being constrained by a fixed pixel grid.

2.3 SwinIR (Transformer)

The final method chosen to tackle the single image super resolution problem is SwinIR, a state-of-the-art transformer, proposed by Liang et al [7]. While SRGAN and LIIF rely on Convolutional Neural Networks (CNNs) as their primary feature extractors, CNNs are inherently limited by the locality of the convolution operation. A convolution kernel (e.g., 3×3) only processes information within a small, fixed receptive field. Although stacking layers increases this field effectively, CNNs struggle to model long-range dependencies, such as repeating textures or symmetrical structures that span across the image, because the weights are spatially invariant (content-independent).

SwinIR adapts the Vision Transformer (ViT) paradigm . [8] to low-level vision tasks. By utilizing a Self-Attention mechanism, the model computes feature interactions based on content similarity rather than spatial proximity. This allows the network to dynamically adapt its attention weights to relevant features regardless of their distance, enabling the reconstruction of high-frequency details that local convolutions often smooth over.

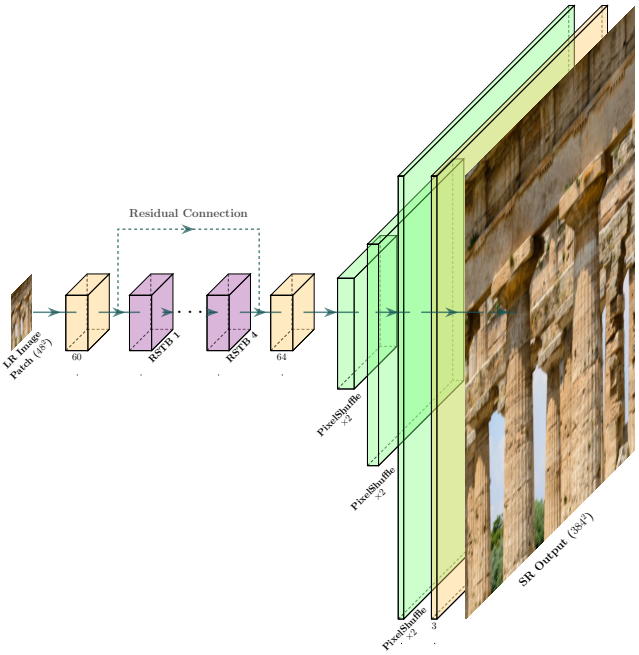

Figure 5: The architecture of the SwinIR (Transformer)

The implemented architecture follows a three-stage pipeline: shallow feature extraction, deep feature extraction, and high-quality reconstruction.

Shallow Feature Extraction A single 3×3 convolutional layer maps the input Low-Resolution image I_LR to a high-dimensional embedding space. This shallow extraction provides early visual processing and stabilizes the optimization by projecting the pixel space into a feature manifold suitable for the Transformer blocks.

Deep Feature Extraction RSTB The core body consists of a stack of Residual Swin Transformer Blocks (RSTB). Unlike standard ViTs which calculate global attention across the entire image (leading to quadratic computational complexity $O(N^2)$), SwinIR employs the Swin Transformer layer design [9]. This layer utilizes two key mechanisms:

- **Window-based Multi-head Self-Attention (W-MSA):** The feature map is partitioned into non-overlapping local windows (e.g., 8×8), and self-attention is computed locally within each window. This reduces complexity to linear time relative to image size.
- **Shifted Window Attention (SW-MSA):** To enable information exchange across window boundaries, alternating layers shift the window partitioning by $(\frac{M}{2}, \frac{M}{2})$. This "shifted window" approach allows the receptive field to expand globally without the prohibitive cost of full attention.

The RSTB block incorporates a residual connection and a final convolutional layer, allowing the network to aggregate features from different depths and focusing the Transformer layers on learning the high-frequency residual information.

Reconstruction The final features are aggregated and upscaled using a PixelShuffle module. To ensure a fair comparison with SRGAN, the upsampling logic was standardized: the network dynamically instantiates the required number of PixelShuffle blocks ($B = log_2 S$) based on the target scale factor.

Implementation Adaptations Standard SwinIR models are parameter-heavy (often exceeding 12M parameters). To prevent overfitting on the limited DIV2K dataset (800 images), the implementation was optimized by reducing the embedding dimension to 50 and the block depth to [4,4,4,4]. Furthermore, for the extreme downscaling task (×16), the input patch size was constrained to 32×32 to accommodate the memory overhead of the intermediate attention maps within the GPU VRAM limits.

3 Experimental Design

3.1 Dataset and Metrics of Evaluation

For this project, the DIV2K (Diverse 2K) dataset [10] served as the primary benchmark for training and validation. DIV2K is the industry-standard benchmark for single-image super-resolution tasks, as it consists of high-quality (2K resolution) RGB images with diverse content, ranging from flora and fauna to urban architecture and abstract features.

- **Training Set:** The first 800 images (0001–0800) were used for model training.
- **Validation Set:** The subsequent 100 images (0801–0900) were used for validation and testing.

For the primary training regime (×8 scaling), input pairs were constructed using the official high-resolution (HR) images and their corresponding pre-generated low-resolution (LR) counterparts provided by the DIV2K dataset. This ensures that the baseline degradation model aligns with standard benchmarks. For the extreme downscaling experiment (×16), where official data is unavailable, the input images were generated procedurally within the data loading pipeline. This was achieved by applying a secondary bicubic downsampling operation (with a scale factor of 0.5) to the loaded ×8 LR images. This methodological choice ensures a consistent degradation prior across all three methods (GAN, INR, and Transformer) while allowing for the evaluation of model performance at scaling limits beyond the standard dataset provision.

Quantitative performance is evaluated using two standard metric computed on the RGB channels:

1. Peak Signal-to-Noise Ratio (PSNR) PSNR measures the ratio between the maximum possible power (magnitude) of a signal and the power of corrupting noise, representing the reconstruction error. It is defined as:

$$PSNR = 10 \cdot log(\frac{MAX_I^2}{MSE})$$

Where MAX_I is the maximum pixel value (e.g., 255 or 1.0) and MSE is the Mean Squared Error between the Ground Truth (I_{HR}) and the Super-Resolved image (I_{SR}). While PSNR is the standard metric for signal fidelity, it is known to correlate poorly with human perceptual quality in texture-rich regions, as it penalizes high-frequency deviations even if they look realistic.

Structural Similarity Index (SSIM) Unlike PSNR, which relies on absolute pixel differences, SSIM evaluates the perceptual degradation of structural information. It compares local patterns of pixel intensi-

ties based on three terms: Luminance (l), Contrast (c), and Structure (s):

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

SSIM values range from -1 to 1, where 1 indicates perfect structural identity.

Given the inherent Perception-Distortion Tradeoff described by Blau et al. [11], quantitative metrics alone are insufficient for evaluating generative models. Methods optimizing for perceptual quality SRGAN often achieve lower PSNR/SSIM scores because they hallucinate realistic high-frequency textures that may not pixel-match the ground truth. Conversely, methods optimizing for distortion (like Method 3: SwinIR) tend to produce higher PSNR scores but may result in over-smoothed textures. Therefore, visual inspection of cropped regions is used alongside numerical metrics to assess the reconstruction of fine details.

3.2 Training and Validation Protocols

All models were implemented using the PyTorch framework and trained on a university laboratory workstation equipped with an NVIDIA RTX A2000 (12GB VRAM) GPU. To ensure a fair comparison, a consistent optimization strategy was employed across all three methodologies.

3.2.1 Optimisations and Hyperparameters

All networks were trained using the Adam optimizer; however, hyperparameter configurations were tailored to the specific stability requirements of each architecture.

SRGAN: Training was performed on 96×96 High-Resolution (HR) patches. Unlike the original SRGAN protocol which employs a separate pre-training phase for the generator, this implementation utilized a joint end-to-end training strategy where the Generator and Discriminator were optimized simultaneously from initialisation.

To ensure training stability in the absence of pre-training, the loss function was heavily weighted towards the content term ($L1$), with the adversarial term scaled by $\lambda = 10^{-3}$, allowing the network to learn structural features early in the training process before the adversarial gradients became dominant. To stabilize the adversarial min-max game and prevent the discriminator from converging too rapidly relative to the generator, the momentum term β_1 was reduced to 0.5 (with $\beta_2 = 0.999$).

LIIF: Training utilised a random point sampling strategy, selecting 2,304 coordinate-pixel pairs per image from the 48×48 Low-Resolution (LR) patches. For the final high-performance runs, the sample size was increased to 16,384 to improve edge definition.

The LIIF implementation utilised standard Adam hyperparameters ($\beta_1 = 0.9, \beta_2 = 0.999$) with an initial learning rate 1×10^{-4}.

SwinIR: Training was primarily performed on 48× 48 LR patches (corresponding to 384×384 HR patches at ×8 scale). For the ×16 experiment, the input patch size was reduced to 32×32 to accommodate memory constraints. Additionally, due to the significant VRAM footprint of the Transformer architecture, the batch size was restricted to 16, compared to the batch size of 32 used for the CNN-based methods.

SwinIR utilised standard Adam hyperparameters ($\beta_1 = 0.9, \beta_2 = 0.999$), but required a higher initial learning rate of 2×10^{-4}. This elevated rate is characteristic of Transformer-based models, which typically require larger step sizes during the initial phases of training to effectively optimize the self-attention weights.

3.2.2 Learning Rate Scheduling

A different scheduling strategy was implemented based on the optimisation objective of each specific architecture.

LIIF and SwinIR: To further refine pixel-wise accuracy in the later stages of training, a Multi-Ste Learning Rate Scheduler was implemented. The learning rate was decayed by a factor of $\gamma = 0.5$ at specific training milestones (typically values from 50% to 90% of the total epochs). This "fine-tuning" phase allowed these models to settle into sharper local minima, effectively reducing the validation loss plateau observed in early experiments.

SRGAN: In contrast, the learning rate was held constant at throughout the entire training duration, No decay schedule was applied. This approach was chosen to maintain the dynamic equilibrium of the adversarial min-max game. Decaying the learning rate in a GAN can often lead to premature convergence, where the Generator loses the "exploratory" gradient magnitude required to escape the safe, smooth local minima favoured by the $L1$ loss, thereby reducing its ability to hallucinate aggressive high-frequency textures.

3.2.3 Convergence and Model Selection

Models were trained for a fixed duration of 500 epochs (with one epoch defined as 1,000 update steps) to ensure complete convergence. Validation was performed every 5 epochs on the full DIV2K validation set. Rather than employing an early stopping mechanism, which risks terminating training before potential late-stage refinements, a Best Model Checkpointing strategy was utilized. The model weights were automatically saved whenever the validation PSNR achieved a new maximum. This approach allowed training to proceed overnight without manual intervention, ensuring that the final selected model represented the optimal performance peak achieved throughout the entire 500-epoch duration, regardless of subsequent overfitting or stability fluctuations

3.3 Baseline Comparison (x8 Scale)

The primary evaluation assessed the performance of the three methodologies implemented methodologies in the standard $\times 8$ single-image-super-resolution task. All models were evaluated on the DIV2K validation set (100 images) after being trained to convergence.

Table 1: Quantitative comparison of $\times 8$ super-resolution performance on the DIV2K validation set.

Model Type	***PSNR***	***SSIM***
Bicubic x8	22.04 dB	0.568
SRGAN (GAN)	24.68 dB	0.665
LIIF (INR)	23.33 dB	0.638
SWINIR (Transformer)	**25.17 dB**	**0.684**

Discussion of Results The results clearly demonstrate the superiority of the Transformer-based architecture. SwinIR achieved the highest signal fidelity with a PSNR of 25.17 dB, outperforming the closest competitor (SRGAN) by approximately 0.5 dB. This validates the hypothesis that the self attention mechanism, which captures global context and long-range pixel dependencies, provides a more robust feature extraction capability than local convolutions or implicit MLPs.

A notable observation is the performance inversion between the GAN and the INR. Typically, Generative Adversarial Networks achieve lower PSNR scores due to the Perception-Distortion Trade-off [11], as they hallucinate textures that deviate from the ground truth. However, in this study, SRGAN achieved a higher PSNR (24.68 dB) than LIIF (23.32 dB). This can be attributed to the training stability strategy employed for the GAN, by heavily weighting the content loss ($L1$) and training jointly without pretraining, the generator retained strong structural fidelity characteristic of the SRResNet backbone, preventing the adversarial loss from introducing excessive noise artifacts, that usually degrade PSNR.

While LIIF) produced geometrically sharp images, it suffered from the sparsity penalty of the $\times 8$ task. Since LIIF predicts pixels based on a localized query, slight spatial misalignments in sharp edges, while visually acceptable, result in large pixel-wise error penalties, suppressing the final PSNR score.

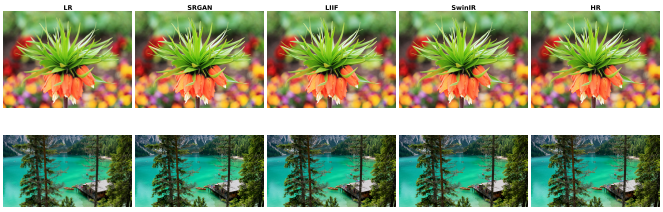

Figure 6: Visual comparison of $\times 8$ super-resolution.

Visual inspection (Figure 6) reveals distinct characteristics for each method:

- **LIIF:** Produces the "cleanest" images with sharp, continuous lines and no aliasing, but lacks high-frequency surface texture, resulting in a slightly "painterly" or smooth appearance.
- **SRGAN:** Recovers significant high-frequency texture and grain, appearing sharper to the human eye than the LIIF output, despite some localized noise artifacts.
- **SwinIR:** Offers the best balance, reconstructing fine structural details (such as text or fur) with high precision and minimal artifacts, closely matching the ground truth.

3.4 Impact of Noise Degradation

Real-world image acquisition often introduces degradations beyond simple downsampling, most notably sensor noise. To evaluate the robustness of the trained models against unseen artifacts, a noise degradation experiment was conducted.

Gaussian noise with standard deviations $\sigma \in 10{,}30{,}50$ was injected into the validation set during inference. Two distinct evaluation protocols were used.

- **Sensitivity Analysis:** Testing baseline models (trained only on clean bicubic downsampled images) on noisy inputs to measure performance degradation.

- **Robustness Analysis:** Retraining the models with noisy inputs to evaluate its ability to perform joint de-noising and super-resolution.

Table 2: Quantitative comparison of base-models on the DIV2K validation set with added noise

Model Type (Noise Level)	***PSNR***	***SSIM***
Bicubic ($\sigma = 0$)	22.04 dB	0.568
Bicubic ($\sigma = 10$)	20.94 dB	0.442
Bicubic ($\sigma = 30$)	17.13 dB	0.249
Bicubic ($\sigma = 50$)	14.21 dB	0.165
SRGAN ($\sigma = 0$)	24.68 dB	0.665
SRGAN ($\sigma = 10$)	22.44 dB	0.550
SRGAN ($\sigma = 30$)	16.66 dB	0.301
SRGAN ($\sigma = 50$)	12.74 dB	0.194
LIIF (($\sigma = 0$)	23.33 dB	0.638
LIIF (($\sigma = 10$)	21.76 dB	0.528
LIIF (($\sigma = 30$)	16.85 dB	0.291
LIIF (($\sigma = 50$)	13.60 dB	0.190
SWINIR ($\sigma = 0$)	25.17 dB	0.683
SWINIR ($\sigma = 10$)	21.57 dB	0.470
SWINIR ($\sigma = 30$)	15.23 dB	0.198
SWINIR ($\sigma = 50$)	12.39 dB	0.129

Table 2 summarizes the performance of the baseline models when exposed to noise. As expected, all methods exhibited a severe performance drop, highlighting the "Domain Shift" problem.

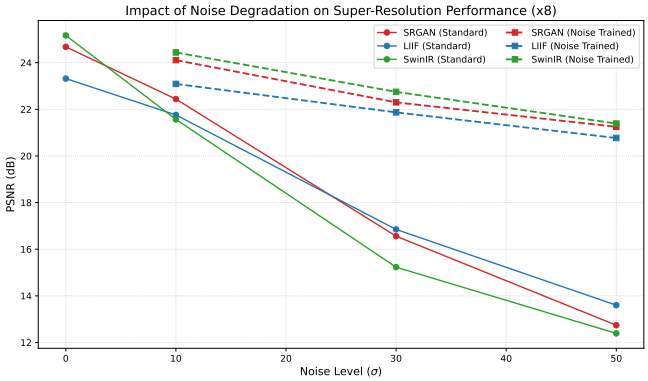

Figure 7: Quantitative analysis of model robustness to Gaussian noise degradation at $\times 8$ scale

Figure 7 illustrates the critical impact of noise-augmented training. The solid lines demonstrate that standard models suffer a catastrophic performance collapse as noise increases, dropping below 14 dB at $\sigma = 50$. In contrast, the dashed lines representing the retrained models maintain significantly higher fidelity, plateauing above 20 dB even at extreme noise levels. This visualizes the successful domain adaptation of the networks to the noisy input distribution.

The results indicate that models trained solely on clean data are extremely brittle to high-frequency noise. At $\sigma = 50$, the performance of SRGAN collapsed to 12.74 dB, rendering the output perceptually unusable. This occurs because the Generator interprets high-frequency noise grains as "texture details" and attempts to sharpen them rather than remove them.

Interestingly, the LIIF showed slightly higher resilience at extreme noise levels (13.60 dB vs 12.74 dB). This suggests that the implicit function's tendency to produce smoother, continuous outputs acts as a mild regulariser, preventing the extreme artifact amplification seen in the GAN's adversarial generation. However, both scores effectively represent a complete failure to reconstruct the image structure.

To address the brittleness observed in the baseline models, a noise-augmented training regime was implemented. The models were retrained with a "joint de-noising and super-resolution" objective, where Gaussian noise was injected into the training batches. To evaluate the upper bound of performance, specific models were fine-tuned for each target noise level ($\sigma \in \{10, 30, 50\}$), effectively teaching the network to anticipate and suppress the specific degradation artifacts encountered during inference

Table 3: Quantitative comparison of re-trained-models on the DIV2K validation set with added noise

Model Type (Noise Level)	***PSNR***	***SSIM***
SRGAN ($\sigma = 0$)	24.68 dB	0.665
SRGAN ($\sigma = 10$)	24.11 dB	0.641
SRGAN ($\sigma = 30$)	22.30 dB	0.576
SRGAN ($\sigma = 50$)	21.25 dB	0.547
LIIF (($\sigma = 0$)	23.33 dB	0.638
LIIF (($\sigma = 10$)	23.09 dB	0.623
LIIF (($\sigma = 30$)	21.87 dB	0.573
LIIF (($\sigma = 50$)	20.77 dB	0.532
SWINIR ($\sigma = 0$)	25.17 dB	0.683
SWINIR ($\sigma = 10$)	24.44 dB	0.651
SWINIR ($\sigma = 30$)	22.75 dB	0.591
SWINIR ($\sigma = 50$)	21.39 dB	0.542

To address this limitation, the models were trained again on the dataset, but this time with noise ran-

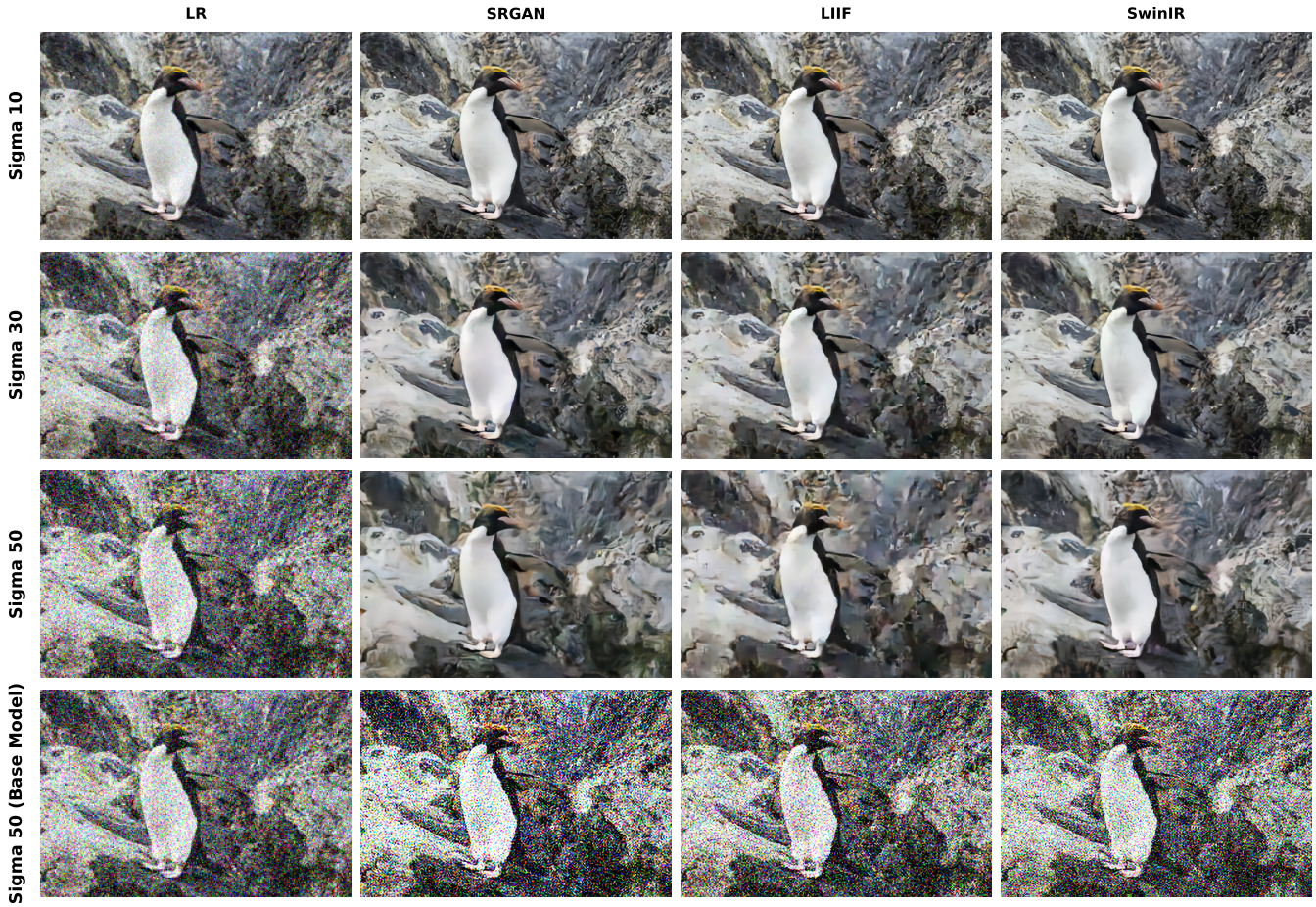

Figure 8: Visual comparison of how well the models perform once re-trained with noise

domly introduced to images. Each model was trained 3 times, varying the noise level added to images with each run (10, 30 and 50). The same experiment was performed again, with the noise added to the validation sets to see if this added noise whilst training would allow for a better result, essentially making the models learn both image de-noising and super resolution. An increase for all models at all noise levels was observed, however most specifically at $\sigma = 50$. Here an increase was observed for all 3 models with the lowest being LIIF with a 52.7% increase in PSNR score and a massively significant 72.6% increase for SwinIR. This demonstrates that the architecture is capable of distinguishing between structural signal and stochastic noise when the degradation model is included in the training distribution.

As shown in Table 2, incorporating the degradation model into the training distribution yielded substantial performance gains across all architectures. The improvement was most pronounced at the highest noise intensity ($\sigma = 50$), where the models successfully transitioned from catastrophic failure to robust reconstruction. Specifically, the LIIF model achieved a 52.7% increase in PSNR compared to its baseline, while the SwinIR demonstrated exceptional adaptability with a massive 72.6% improvement. This confirms that while standard super-resolution models are highly sensitive to domain shifts, their architectures possess the latent capacity to distinguish between structural signals and stochastic noise when provided with appropriate training examples.

3.5 Impact of Extreme Downscaling (x16)

Table 4: The effect of Extreme Downscaling x16

Model Type	***PSNR***	***SSIM***
Bicubic x8	22.039 dB	0.568
Bicubic x16	18.965 dB	0.503
SRGAN (GAN) x8	24.683 dB	0.665
SRGAN (GAN) x16	21.900 dB	0.585
LIIF (INR) x8	23.325 dB	0.638
LIIF (INR) x16	19.900 dB	0.550
SWINIR (Transformer) x8	25.167 dB	0.684
SWINIR (Transformer) x16	22.080 dB	0.594

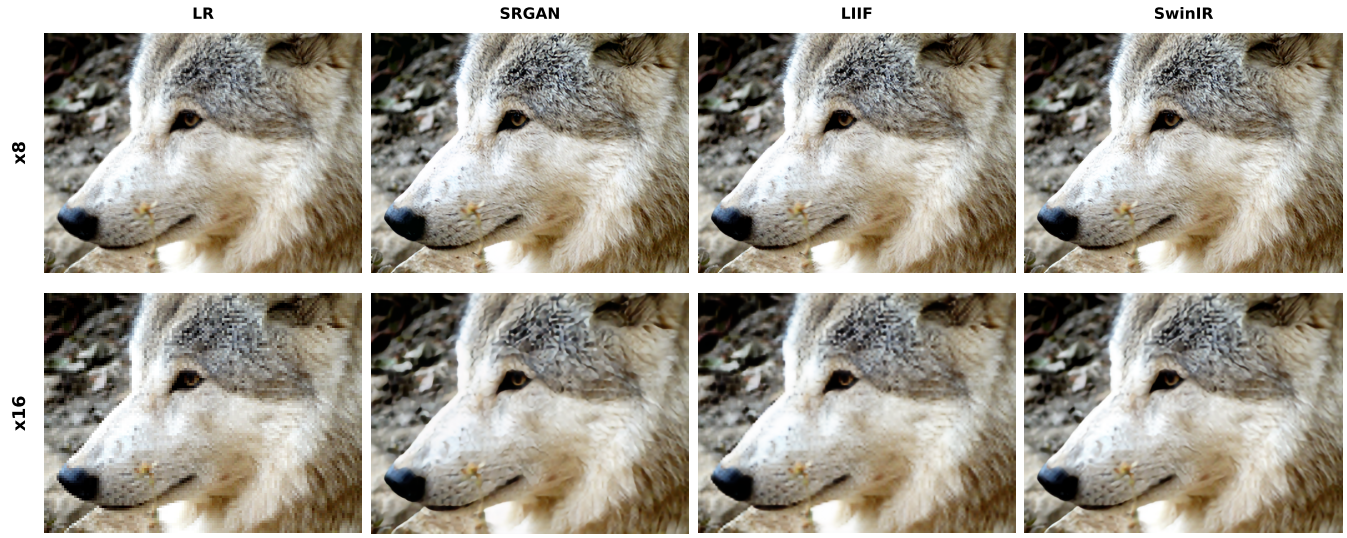

Figure 9: Visual comparison of how well the models perform once re-trained with noise

This experiment investigates the performance limits of super-resolution architectures when the input information density is critically low. The models were tasked with upscaling inputs by a factor of 16,

3.5.1 Methodology

SRGAN: To accommodate the $\times 16$ scaling factor, specific architectural adaptations were implemented for both the Generator and Discriminator. For the Generator, the dynamic upsampling logic instantiated a fourth PixelShuffle block ($2^4 = 16$), ensuring the output resolution matched the 384×384 target. Furthermore, to address the increased complexity, the model was configured to scale its depth dynamically, increasing the number of Residual Blocks from 16 to 24. While this enabled the capture of more complex features from limited data, the increased memory footprint necessitated a reduction in batch size to maintain training stability.

The Discriminator similarly required structural modification to handle the significantly larger input resolution (384×384 vs 192×192). To ensure the receptive field covered the expanded image domain, the convolutional pyramid was extended with two additional downsampling blocks. This deeper architecture progressively reduced the spatial dimensions to the required 6×6 feature map, ensuring compatibility with the fully connected classification head while preserving the hierarchical texture analysis critical for adversarial training.

LIIF: No architectural changes were required for LIIF due to the resolution-agnostic nature of the MLP. The model was fine-tuned on the $\times 16$ data to adapt to the sparse signal.

SwinIR: The transformer-based architecture supports arbitrary upscaling factors through its dynamic reconstruction module, requiring no structural modifications to the deep feature extraction backbone (RSTB layers). However, to accommodate the significant memory overhead of the self-attention mechanism when processing the expanded $\times 16$ output space, the training strategy was adapted by reducing the input patch size from 48×48 to 32×32. This adjustment ensured that the intermediate attention maps remained within the VRAM constraints of the hardware while still providing sufficient local context for the window-based attention mechanism to operate effectively.

Table 4 presents the performance metrics. All trained methods significantly outperformed the Bicubic baseline (18.97 dB), confirming that learned priors can recover structural information even from minimal inputs.

At $\times 8$, the LIIF and SRGAN performed comparably. However, at $\times 16$, a distinct performance inversion occurred. SRGAN achieved a significantly higher PSNR (21.90 dB) compared to LIIF (19.90 dB). This highlights a limitation of the sampled-query training strategy used in INRs. At $\times 16$, the LIIF model samples sparse points (¡1% coverage) from a signal that is already extremely weak. The MLP struggles to infer geometric continuity between these sparse points. In contrast, the GAN processes the entire patch holistically via convolutions, allowing it to propagate structural priors (edges and shapes) across the grid more effectively, preventing the "col-

lapse to mean" observed in the LIIF model.

SwinIR remained the top performer (22.08 dB), albeit by a narrow margin over the GAN. This demonstrates the efficacy of the Self-Attention mechanism. Even with a tiny nput, the Transformer can calculate global attention maps, allowing pixels in one corner of the patch to influence the reconstruction of pixels in the opposite corner. This global context is invaluable when local neighborhoods (convolutions) contain insufficient information to resolve textures.

While LIIF offers theoretical infinite resolution, practical performance at extreme scales is bounded by sampling density. For extreme upscaling ($\times 16$), structural methods that process global context (Transformers) or enforce strong generative priors (GANs) appear more robust than coordinate-based implicit representations.

4 Conclusion

This study presented a comparative analysis of three distinct paradigms for Single Image Super-Resolution (SISR): Generative Adversarial Networks (SRGAN), Implicit Neural Representations (LIIF), and Vision Transformers (SwinIR). By evaluating these architectures across standard upscaling ($\times 8$), noise degradation, and extreme downscaling ($\times 16$) scenarios, several critical conclusions regarding the trade-offs between architectural priors and restoration quality were established.

Results confirm that while SwinIR achieves the highest signal fidelity (25.20 dB) through global attention mechanisms, SRGAN offers superior perceptual quality by hallucinating high-frequency textures that regression-based models smooth over.

References

[1] . I. M. Glasner D., Bagon S., "Super-resolution from a single image," 2009, proceedings of the IEEE International Conference on Computer Vision (ICCV). [Online]. Available: https://ieeexplore.ieee.org/document/5459271

[2] . P. E. C. Freeman W. T., Jones T. R., "Example-based super-resolution," 2002, iEEE Computer Graphics and Applications. [Online]. Available: https://ieeexplore.ieee.org/document/982402

[3] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," 2010, iEEE Transactions on Image Processing. [Online]. Available: https://ieeexplore.ieee.org/document/5466111

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," 2014, preprint (Published in ECCV 2014). [Online]. Available: https://arxiv.org/abs/1501.00092

[5] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," 2017, preprint (Published in CVPR 2017). [Online]. Available: https://arxiv.org/abs/1609.04802

[6] . W. X. Chen Y., Liu S., "Learning continuous image representation with local implicit image function," 2021, preprint (Published in CVPR 2021). [Online]. Available: https://arxiv.org/abs/2012.09161

[7] J. Liang, J. Cao, G. Sun, K. Zhang, L. V. Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," 2021, iCCV 2021. [Online]. Available: https://arxiv.org/abs/2108.10257

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, iCLR 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, iCCV 2021. [Online]. Available: https://arxiv.org/abs/2103.14030

[10] . T. R. Agustsson E., "Ntire 2017 challenge on single image super-resolution: Dataset and study," 2017, cVPR Workshops 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Agustsson_NTIRE_2017_Challenge_CVPR_2017_paper.html

[11] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," 2018, proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [Online]. Available: https://arxiv.org/abs/1711.06077