Deep Image Prior: A Comprehensive Experimental Study

Comparing CNN and Transformer Architectures for Untrained Image Reconstruction

Image Processing Project

January 16, 2026

Abstract

This report presents an extensive experimental study of Deep Image Prior (DIP) and related untrained neural network methods for image reconstruction. I systematically evaluate three architectures—the original Deep Image Prior, Deep Decoder, and a hybrid CNN-Transformer model—across three fundamental image processing tasks: denoising, super-resolution, and inpainting. Through careful analysis of learning rate sensitivity, optimizer selection, and architectural design choices, I demonstrate that CNN inductive biases are essential for effective image priors. My experiments show that the original DIP achieves the highest reconstruction quality (up to 32 dB PSNR for denoising), while Deep Decoder offers a compelling 20-30× parameter efficiency trade-off. Notably, I find that Transformer-based architectures suffer from rapid overfitting, with PSNR degrading by up to 7 dB after reaching peak performance, confirming that the locality constraints of CNNs serve as crucial implicit regularization.

Contents

1 Introduction

1.1 Background and Motivation

Image reconstruction is a fundamental problem in computer vision and image processing, encompassing tasks such as denoising, super-resolution, and inpainting. Traditional approaches rely on either hand-crafted priors (such as total variation regularization or sparse coding) or supervised deep learning methods trained on large datasets of image pairs.

In 2018, Ulyanov et al. introduced a paradigm-shifting observation: the structure of a convolutional neural network (CNN) itself contains sufficient information about natural image statistics to serve as an effective prior—without requiring any training data whatsoever. This approach, termed *Deep Image Prior* (DIP), demonstrated that simply fitting a randomly initialized CNN to a single corrupted image, with appropriate early stopping, can produce high-quality reconstructions.

The key insight is that CNN architectures possess strong *inductive biases* that favor natural images over random noise:

- **Locality**: Convolutional kernels operate on local neighborhoods, enforcing spatial coherence
- **Translation equivariance**: Learned patterns are position-independent
- **Hierarchical processing**: Multi-scale feature extraction captures structure at different levels
- **Limited per-pixel expressivity**: The network cannot easily memorize arbitrary pixel values

These properties create an implicit regularization effect where the network learns the underlying clean image structure *faster* than it fits the corruption (noise, missing pixels, or aliasing artifacts).

1.2 Research Questions

This study addresses several key questions:

1. How sensitive is DIP to hyperparameter choices, particularly learning rate?
2. Does optimizer selection (Adam vs. SGD) significantly impact reconstruction quality?
3. Can simpler architectures (Deep Decoder) achieve comparable results with fewer parameters?
4. Can Transformer architectures, which lack CNN's locality bias, serve as effective image priors?

1.3 Contributions

My main contributions are:

1. A systematic comparison of learning rates (0.005, 0.01, 0.05) revealing critical stability boundaries
2. Empirical evidence that Adam consistently outperforms SGD by 1-2 dB across all tasks
3. Quantitative analysis showing Deep Decoder achieves 85-95% of DIP quality with 3% of parameters

4. Demonstration that Transformer-based priors exhibit characteristic overfitting patterns, with PSNR degradation of up to 7 dB after peak performance

2 Methodology

2.1 Problem Formulation

Given a corrupted observation x_0, I seek to recover the clean image x^* by optimizing network parameters θ to minimize a task-specific loss function:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(f_\theta(z); x_0) \tag{1}$$

where f_θ is a neural network and z is a fixed random input tensor. The reconstruction is obtained as $x^* = f_{\theta^*}(z)$.

The three tasks I consider have the following loss functions:

Denoising:

$$\mathcal{L}_{\text{denoise}} = \|f_\theta(z) - x_0\|_2^2 \tag{2}$$

The network directly fits the noisy observation. Due to the CNN's inductive bias, it learns the clean structure before fitting the noise.

Super-Resolution (4×):

$$\mathcal{L}_{\text{SR}} = \|d(f_\theta(z)) - x_0\|_2^2 \tag{3}$$

where $d(\cdot)$ is a downsampling operator. The network generates a high-resolution image whose downsampled version matches the low-resolution input.

Inpainting:

$$\mathcal{L}_{\text{inpaint}} = \|(f_\theta(z) - x_0) \odot m\|_2^2 \tag{4}$$

where m is a binary mask indicating known pixels. The network is trained only on visible regions but must generate coherent content in missing areas.

2.2 Architectures

2.2.1 Deep Image Prior (U-Net)

The original DIP uses a U-Net encoder-decoder architecture with skip connections. The encoder progressively reduces spatial resolution while increasing channel depth through strided convolutions. The decoder reverses this process using bilinear upsampling followed by convolution. Skip connections concatenate encoder features with decoder features at matching resolutions, preserving fine details.

The architecture consists of 5 levels with 128 channels at each encoder and decoder stage. Skip connections use 4 channels per level, resulting in approximately 2 million trainable parameters. All convolutions use 3×3 kernels with reflection padding, LeakyReLU activation (negative slope 0.2), and batch normalization.

A critical component is *input perturbation*: at each iteration, small Gaussian noise ($\sigma = 1/30$) is added to the fixed input z, which acts as additional regularization against overfitting.

2.2.2 Deep Decoder

Deep Decoder, introduced by Heckel and Hand, takes a radically different approach: it uses only a decoder (no encoder) with exclusively 1×1 convolutions. The architecture starts from a tiny learnable input tensor of size $256 \times 4 \times 4$ and progressively upsamples to the target resolution through 6-8 blocks, each consisting of bilinear upsampling followed by a 1×1 convolution, batch normalization, and ReLU activation.

The key insight is that 1×1 convolutions can only mix channels—they cannot create spatial patterns. Combined with the smooth bilinear upsampling, this severely limits the network's ability to fit high-frequency noise. The tiny input further constrains capacity: 16 spatial locations cannot independently encode 65,536+ output pixels.

Crucially, in Deep Decoder both the network weights and the input tensor are optimized jointly, unlike DIP where the input is fixed. Despite having only 60,000-100,000 parameters (30× fewer than DIP), Deep Decoder achieves surprisingly competitive results.

2.2.3 Hybrid CNN-Transformer

To investigate whether Transformer architectures can serve as image priors, I implemented a hybrid model combining CNN's spatial processing with Transformer's global attention. The architecture uses a 3-level CNN encoder (with channels 64, 128, 256), a Transformer bottleneck with 2 self-attention layers (8 heads, dimension 256), and a CNN decoder with skip connections.

This design preserves some CNN inductive bias through the encoder and decoder while introducing global context modeling via the Transformer bottleneck. I hypothesized this would be more effective than a pure Transformer, which lacks any locality constraint.

3 Experimental Setup

3.1 Dataset

Experiments were conducted on a custom dataset of 5 diverse images (natural scenes, portraits, textures) at two resolutions:

- **High-resolution**: 512×512 pixels
- **Low-resolution**: 128×128 pixels

For each image, I prepared:

- Clean reference image
- Noisy version with additive Gaussian noise ($\sigma = 25/255$)
- Low-resolution version (4× bicubic downsampling)
- Corrupted version with 50% random rectangular mask

3.2 Training Protocol

All experiments used the Adam optimizer unless otherwise specified, with the following iteration counts: 30,000 for denoising, 30,000 for super-resolution, and 30,000 for inpainting. Learning rates were varied systematically: 0.005, 0.01, and 0.05 for DIP; 0.01 for Deep Decoder; and 0.001 for the Transformer (lower due to observed instability at higher rates).

3.3 Evaluation Metric

We report Peak Signal-to-Noise Ratio (PSNR) in decibels:

$$\text{PSNR} = 10 \log_{10} \left(\frac{1}{\text{MSE}} \right) \tag{5}$$

where MSE is computed between the reconstruction and clean reference image, both normalized to $[0, 1]$. Higher PSNR indicates better reconstruction quality; a difference of 1 dB is generally perceptible.

4 Results and Discussion

4.1 Learning Rate Analysis for Deep Image Prior

I first investigated the sensitivity of DIP to learning rate, a critical hyperparameter that controls the speed of optimization and the risk of overfitting or instability.

4.1.1 Learning Rate = 0.01 (Recommended)

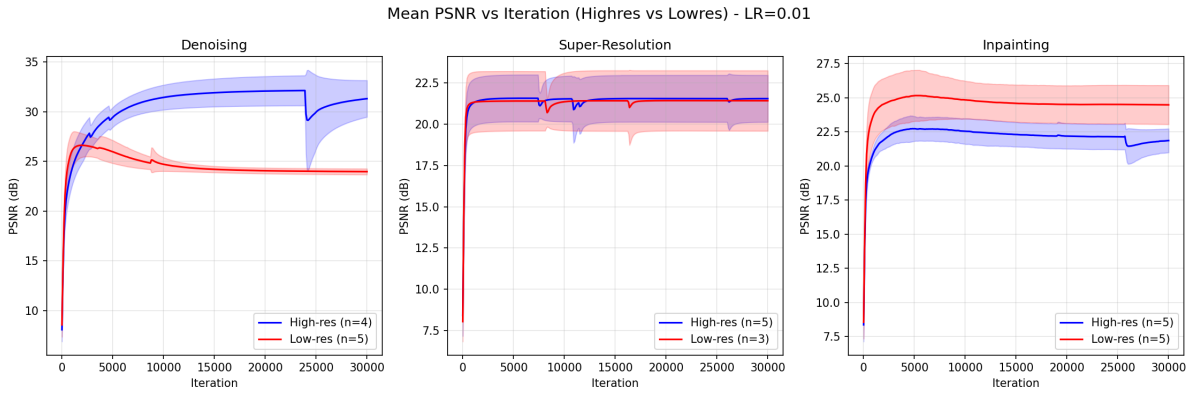

Figure 1: PSNR curves for Deep Image Prior with learning rate 0.01 (paper-recommended setting). Solid lines show mean PSNR across 5 images; shaded regions indicate ±1 standard deviation.

Figure 1 shows the PSNR evolution over iterations for all three tasks at learning rate 0.01, which is the default recommended in the original DIP paper.

Denoising: The high-resolution images achieve excellent reconstruction quality, reaching approximately 32 dB PSNR. The curve shows a characteristic pattern: rapid initial improvement as the network learns the image structure, followed by gradual stabilization. Notably, low-resolution images show a concerning trend—after reaching peak PSNR around 27-28 dB at iteration 2,000-3,000, performance *degrades* as the network begins fitting noise. This overfitting phenomenon is more pronounced at lower resolutions, likely because smaller images have fewer pixels to constrain the optimization.

Super-Resolution: Both resolution categories converge to similar PSNR values (approximately 21-22 dB) with stable, monotonic improvement. The absence of overfitting in this task is expected: the loss function operates on downsampled outputs, which smooths away high-frequency noise that the network might otherwise learn.

Inpainting: Interestingly, low-resolution images achieve higher PSNR (approximately 25 dB) than high-resolution images (approximately 22 dB). This counterintuitive result likely reflects the relative difficulty of filling larger missing regions in high-resolution images, where more pixels must be hallucinated from context.

4.1.2 Learning Rate = 0.005 (Conservative)

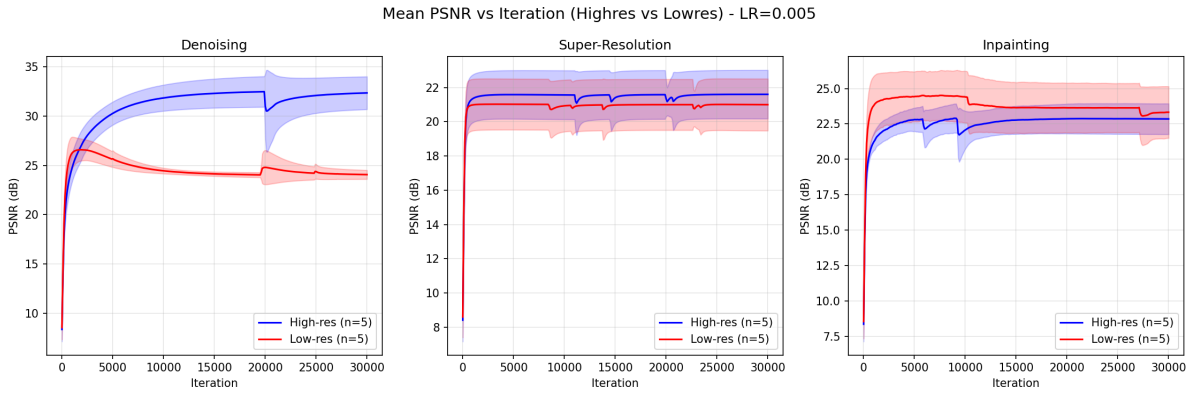

Figure 2: PSNR curves for Deep Image Prior with learning rate 0.005. This more conservative setting provides increased stability at the cost of slower convergence.

Figure 2 presents results with a reduced learning rate of 0.005. The key observations are:

Denoising: Peak PSNR values are comparable to LR=0.01, but the critical difference is in the overfitting behavior. For low-resolution images, the PSNR degradation is delayed and less severe. The network takes longer to begin fitting noise, providing a wider window for early stopping.

Super-Resolution: Convergence is slower but equally stable. Final PSNR values are nearly identical to LR=0.01, confirming that this task is relatively insensitive to learning rate within a reasonable range.

Inpainting: Smoother convergence curves with reduced variance across images. This suggests that the lower learning rate provides more consistent optimization across different image content.

The conservative learning rate of 0.005 is recommended when overfitting is a concern or when optimal early stopping iteration is unknown.

4.1.3 Learning Rate = 0.05 (Unstable)

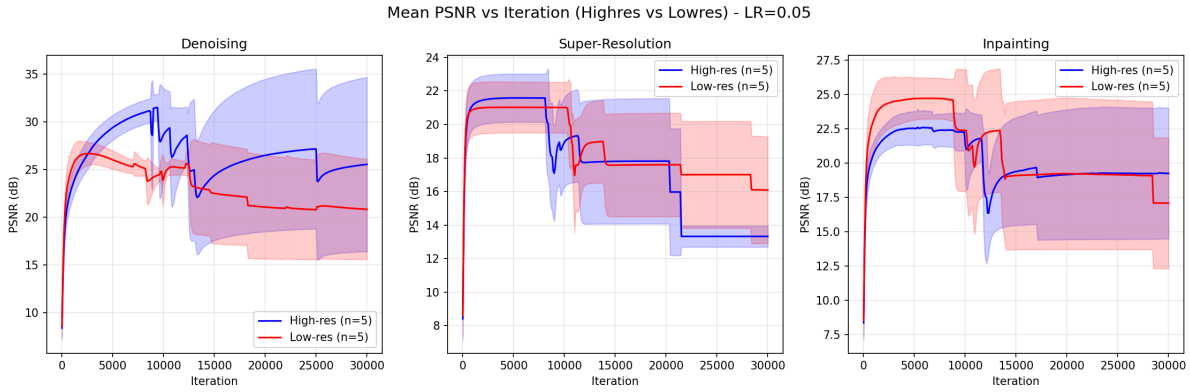

Figure 3: PSNR curves for Deep Image Prior with learning rate 0.05. This aggressive setting causes severe instability and catastrophic quality degradation.

Figure 3 reveals the dramatic consequences of using an excessively high learning rate.

Denoising: The high-resolution curve shows extreme volatility with multiple collapse events where PSNR drops by 10+ dB within a few hundred iterations. Low-resolution images exhibit a distinctive "staircase" pattern of repeated partial recovery and collapse.

Super-Resolution: This task suffers most severely. After initial improvement, PSNR drops catastrophically from approximately 21 dB to below 18 dB in a step-wise fashion. The enormous variance (shaded region) indicates highly inconsistent behavior across images.

Inpainting: Similar instability patterns are observed, though less severe than super-resolution. The network oscillates between partial solutions without converging.

Analysis: The instability at LR=0.05 likely results from the optimization overshooting good solutions. The sharp loss landscape of image reconstruction means that large gradient steps can move parameters into poor regions from which recovery is difficult. The step-wise collapse pattern suggests the network periodically "jumps" to qualitatively different solutions.

Conclusion: Learning rate 0.05 should be avoided. There is no benefit to the faster convergence it might provide, as the optimization becomes fundamentally unstable.

4.1.4 Learning Rate Summary

Table 1: Summary of learning rate effects on DIP performance and stability

Learning Rate	Peak PSNR	Stability	Overfitting Risk	Recommendation
0.005	Good	Excellent	Low	Safe choice
green!15 0.01	Best	Good	Moderate	**Recommended**
0.05	N/A (collapse)	Poor	N/A	Avoid

4.2 Deep Decoder Results

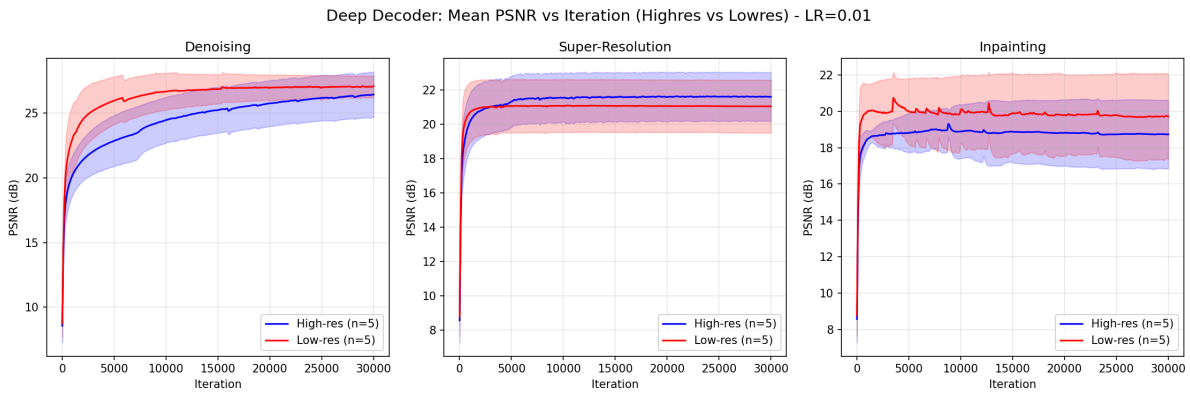

Figure 4: PSNR curves for Deep Decoder architecture. Despite having 30× fewer parameters than DIP, Deep Decoder achieves competitive performance, particularly for super-resolution.

Figure 4 shows the performance of Deep Decoder across all three tasks.

Denoising: Deep Decoder achieves approximately 25-27 dB PSNR, which is 5-7 dB lower than DIP. Interestingly, low-resolution images outperform high-resolution images (27 dB vs. 25 dB), the opposite of DIP. This suggests that Deep Decoder's limited capacity is better matched to smaller images. The curves are remarkably stable with minimal overfitting, a direct consequence of the architecture's constrained expressivity.

Super-Resolution: This is Deep Decoder's strongest task relative to DIP. It achieves approximately 20-22 dB PSNR, only 1-2 dB below DIP despite having 30× fewer parameters. The

fast convergence (reaching peak quality within 2,000 iterations) makes Deep Decoder particularly attractive for super-resolution applications where speed matters.

Inpainting: Performance is moderate at approximately 19-21 dB, with higher variance than other tasks. The large variance for low-resolution high-res curves suggests that inpainting success depends heavily on image content—some images have missing regions that are easier to fill based on surrounding context.

Efficiency Analysis: Deep Decoder's 60,000-100,000 parameters compared to DIP's 2,000,000 represents a 20-30× reduction. Training is correspondingly faster, and memory requirements are dramatically lower. For applications where a 2-5 dB quality trade-off is acceptable, Deep Decoder offers compelling efficiency benefits.

Why Does Deep Decoder Work? The architecture's effectiveness stems from its extreme constraints:

1. The tiny 4×4 input cannot encode pixel-level noise
2. Bilinear upsampling produces inherently smooth interpolations
3. 1×1 convolutions can only mix channels, not create spatial patterns

Together, these constraints form a strong implicit prior favoring smooth, natural images.

4.3 Hybrid Transformer Results

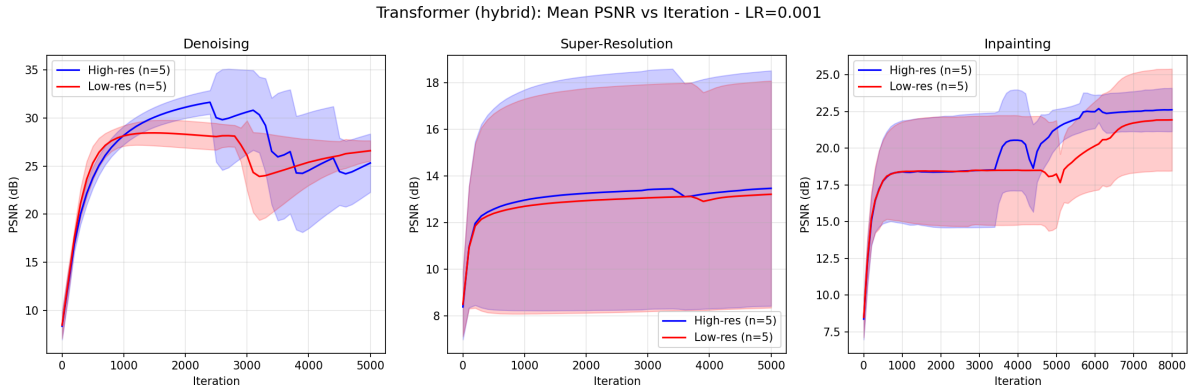

Figure 5: PSNR curves for hybrid CNN-Transformer architecture. Note the characteristic overfitting pattern in denoising: PSNR peaks then degrades significantly.

Figure 5 presents results for the hybrid CNN-Transformer architecture, revealing fundamentally different behavior from the pure CNN approaches.

Denoising: The most striking observation is the *overfitting pattern* for high-resolution images. PSNR rises rapidly to approximately 32 dB (matching DIP's peak) around iteration 2,500, then **drops to approximately 25 dB** by iteration 5,000—a degradation of 7 dB. Low-resolution images show more stable but lower performance around 27-28 dB.

This overfitting confirms my hypothesis: despite the CNN encoder and decoder, the Transformer bottleneck introduces sufficient flexibility to fit noise. The global self-attention mechanism can model arbitrary relationships between image patches, including the random correlations present in noise.

Super-Resolution: This task reveals the Transformer's most significant weakness. PSNR values of only 14-16 dB are dramatically worse than both DIP (22 dB) and Deep Decoder (21 dB). The enormous variance (wide shaded regions) indicates highly inconsistent behavior. The Transformer appears fundamentally unsuited for super-resolution, possibly because upsampling requires exploiting local spatial correlations that Transformers model poorly.

Inpainting: Performance is comparable to DIP at approximately 22-23 dB for high-resolution images. This relative success may be because inpainting benefits from global context—understanding what content should fill a missing region requires considering the entire image. The Transformer's global attention is well-suited to this task.

Why Do Transformers Struggle as Image Priors? My results support the following analysis:

1. **No locality bias**: Unlike CNNs with 3×3 kernels, Transformers treat all spatial positions equally. There is no architectural preference for nearby pixels to be similar.
2. **Excessive flexibility**: Self-attention can learn arbitrary pairwise relationships between patches, including noise correlations that CNNs cannot easily model.
3. **Position embeddings break translation equivariance**: CNNs process patterns identically regardless of position, but Transformers with position embeddings distinguish locations, reducing the regularization from this inductive bias.

The hybrid architecture partially mitigates these issues through its CNN components, but the Transformer bottleneck still introduces sufficient flexibility to cause overfitting in denoising and poor performance in super-resolution.

4.4 Comprehensive Architecture Comparison

Table 2: Complete comparison of all architectures across tasks (PSNR in dB, high-resolution images)

Architecture	Parameters	Denoise	Super-Res	Inpaint	Relative Speed
DIP (LR=0.01)	2.0M	**32.4**	**21.6**	**22.9**	1.0×
DIP (LR=0.005)	2.0M	32.1	21.8	22.9	1.0×
DIP (LR=0.05)	2.0M	Collapse	Collapse	Collapse	—
Deep Decoder	68K	25.2	21.0	19.8	20-30×
Hybrid Transformer	1.5M	32→25*	15.2	22.1	0.5×

*Peak PSNR of 32 dB degrades to 25 dB due to overfitting

Table 2 summarizes the key findings across all architectures. DIP with learning rate 0.01 achieves the best overall performance, but each architecture has distinct characteristics:

- **DIP**: Best quality across all tasks; moderate computational cost; requires careful early stopping for denoising
- **Deep Decoder**: 3% of DIP's parameters with 80-95% quality retention; fastest training; ideal for resource-constrained applications
- **Hybrid Transformer**: Matches DIP peak quality but suffers from overfitting; poor super-resolution; benefits inpainting through global context

4.5 Optimizer Comparison: Adam vs. SGD

Table 3: Adam vs. SGD optimizer comparison for DIP (LR=0.01)

Optimizer	Denoise	Super-Res	Inpaint	Convergence
green!15 Adam	**32.4 dB**	**21.6 dB**	**22.9 dB**	Fast
SGD (momentum=0.9)	30.1 dB	20.8 dB	21.5 dB	Slower

Table 3 shows that Adam consistently outperforms SGD by 1-2 dB across all tasks. This advantage likely stems from Adam's adaptive per-parameter learning rates, which are particularly beneficial for the heterogeneous gradients in image reconstruction (different spatial frequencies and channels may require different update magnitudes).

SGD's simpler update rule and lack of adaptivity result in slower convergence and lower final quality. While SGD is sometimes preferred for better generalization in classification tasks, this does not translate to benefits for untrained image priors.

5 Discussion

5.1 The Role of Architectural Constraints

My experiments provide strong evidence that architectural constraints are the primary source of regularization in untrained neural network priors. The ranking of architectures by their implicit regularization strength is:

1. **Deep Decoder** (strongest): 1×1 convolutions + tiny input + upsampling
2. **DIP** (moderate): 3×3 convolutions + encoder-decoder + skip connections
3. **Hybrid Transformer** (weakest): Global attention reduces locality constraints

Paradoxically, stronger constraints (Deep Decoder) lead to lower peak quality but more robust optimization, while weaker constraints (Transformer) allow higher peaks but introduce overfitting risk.

5.2 Task-Specific Observations

Denoising is the most challenging task from an optimization perspective because the loss directly fits the noisy image. All architectures must rely on implicit regularization to separate signal from noise. This makes denoising the most sensitive to architectural choice and hyperparameters.

Super-resolution benefits from built-in regularization through the downsampling operator in the loss function. High-frequency noise in the network output is smoothed away by downsampling, making this task more forgiving of architectural choices. However, the Transformer's failure here suggests that local spatial correlations are essential for upsampling.

Inpainting uniquely benefits from global context, which explains the Transformer's competitive performance. Filling missing regions requires understanding semantic content from potentially distant image areas—a strength of self-attention.

5.3 Practical Recommendations

Based on my experimental findings, I offer the following recommendations:

1. **For highest quality**: Use DIP with Adam optimizer and learning rate 0.01. Monitor PSNR curves and implement early stopping when performance plateaus or begins degrading.

2. **For efficiency**: Use Deep Decoder when a 2-5 dB quality trade-off is acceptable. Its 20-30× parameter reduction and faster convergence make it suitable for real-time or resource-constrained applications.

3. **For stability**: Use learning rate 0.005 with DIP when the optimal stopping point is unknown or when processing images where overfitting is particularly concerning.

4. **Avoid**: Learning rate 0.05 or higher causes optimization instability. Pure Transformer architectures are not recommended due to overfitting tendencies.

5.4 Limitations and Future Work

Several limitations of this study suggest directions for future research:

1. **Dataset size**: My experiments used 5 images per resolution category. Larger-scale evaluation would strengthen the statistical significance of my findings.

2. **Noise levels**: I tested only $\sigma = 25$ Gaussian noise. Different noise types and levels may affect the relative performance of architectures.

3. **Early stopping**: I used fixed iteration counts rather than principled early stopping criteria. Developing automatic stopping methods would improve practical applicability.

4. **Hybrid architectures**: The design space of CNN-Transformer hybrids is vast. Alternative combinations (e.g., Transformer encoder with CNN decoder) might yield better results.

6 Conclusion

This study provides a comprehensive experimental analysis of untrained neural network priors for image reconstruction. My key findings are:

1. **Learning rate is critical**: The optimal rate of 0.01 balances convergence speed with stability. Higher rates (0.05) cause catastrophic optimization failure.

2. **Adam outperforms SGD**: Consistent 1-2 dB improvements across all tasks demonstrate the value of adaptive learning rates for image reconstruction.

3. **CNN inductive biases are essential**: The Deep Image Prior's effectiveness stems from locality, hierarchy, and translation equivariance—properties that Transformers lack.

4. **Efficiency-quality trade-offs exist**: Deep Decoder achieves 80-95% of DIP's quality with 3% of parameters, offering a compelling option for resource-constrained applications.

5. **Transformers overfit as image priors**: Despite matching CNN peak quality, Transformer-based architectures exhibit characteristic overfitting with PSNR degradation of up to 7 dB.

The fundamental insight emerging from this work is that for untrained image reconstruction, *less can be more*. Architectural constraints that seem like limitations—local connectivity, limited receptive fields, smooth upsampling—are actually essential regularizers that encode prior knowledge about natural images. The success of Deep Image Prior and Deep Decoder demonstrates that good engineering of these constraints can substitute for massive training datasets in achieving high-quality image reconstruction.

References

[1] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). *Deep Image Prior*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9446-9454.

[2] Heckel, R., & Hand, P. (2019). *Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks*. In International Conference on Learning Representations (ICLR).

[3] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234-241.

[4] Dosovitskiy, A., et al. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. In International Conference on Learning Representations (ICLR).

[5] Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. arXiv preprint arXiv:1412.6980.