WEAVING THE CANVAS

AI INPAINTING MEETS SUPER-RESOLUTION

K. Aditya (AI22BTECH11013)
M. Ganesh (AI22BTECH11017)

Ch. Kushwanth (AI22BTECH11006)
T. Keshavardhan (AI22BTECH1109)

INTRODUCTION & MOTIVATION

Cultural Heritage at Risk

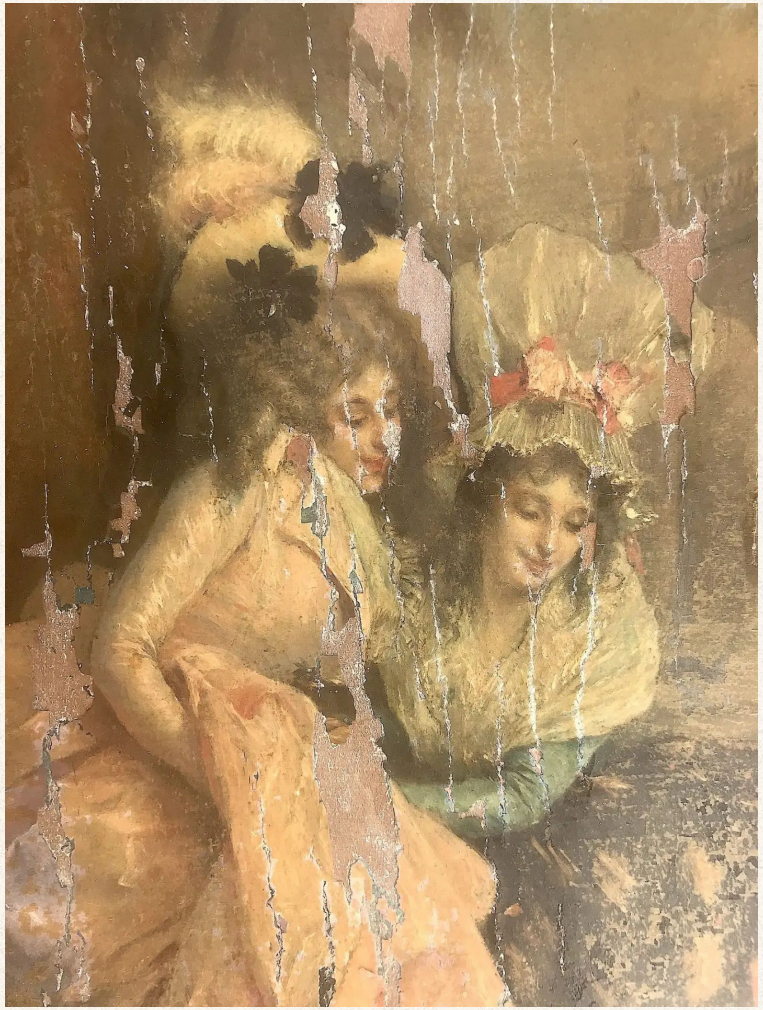
Many historic paintings suffer physical damage over time—scratches, holes, and fading that obscure the artist's original intent. Traditional restoration can be invasive, costly, and irreversible.

Gap in Digital Restoration

While inpainting algorithms can plausibly fill missing regions, they often lack fine detail. Separately, super-resolution models enhance texture but can't hallucinate missing content.

Unified Pipeline Needed

There's a clear need for a non-destructive, fully digital workflow that both reconstructs missing areas and enhances resolution, preserving authenticity while reinstating fine brushwork.



PROBLEM STATEMENT

We aim to develop an end-to-end pipeline for image super-resolution and image inpainting that can robustly handle diverse datasets of natural and artistic images.

"We may speak of a restoration of painting only when the restoration brings back the spirit of the creator."
— *Piero della Francesca*

PROBLEM STATEMENT

Image Inpainting with Fourier Convolutions (LaMa)

Learn a generator $g_\phi : [I_{\text{masked}}, M] \longmapsto \hat{I}$, restoring masked regions. The architecture leverages *Fast Fourier Convolutions (FFC),* which combine:

Spatial path: standard (3×3)convolutions

Global path: spectral transforms via real 2D FFT and 1×1 convolutions in frequency domain.

Total loss: $\mathcal{L}\text{total} = \kappa\,\mathcal{L}\text{adv} + \alpha\,\mathcal{L}_{\text{vgg}} + \beta\,\mathcal{L}_{\text{fm}} + \gamma\,\mathcal{L}_{\text{gp}},$

where ,

$(\mathcal{L}_{\text{adv}})$ is adversarial (PatchGAN) loss,

$(\mathcal{L}_{\text{vgg}})$ is perceptual loss from VGG19 feature maps

(L_{fm}) is discriminator feature-matching loss,

$(\mathcal{L}_{\text{gp}})$ is the R1 gradient-penalty: $\mathcal{L}_{\text{gp}} = \lambda\,\mathbb{E}\hat{x} \sim P_r\left[\|\nabla_{\hat{x}} D(\hat{x})\|_2^2\right],$

and typical hyperparameters are $\kappa = 10,\ \alpha = 30,\ \beta = 100,\ \gamma = 0.001$

Evaluation metrics: FID & Inception Score.

Assumptions:

- Masks and inputs are jointly available at training time.
- The generator and discriminator are trained jointly via alternating optimization using Adam (LR = 1e-4)
- No other regularization beyond the gradient penalty is applied

PROBLEM STATEMENT

Image Super-Resolution

Learn a mapping such that $\hat{I}$ approximates a high-resolution target $f_\theta : \underbrace{\tilde{I}}_{256 \times 256} \longmapsto \underbrace{\hat{I}}_{512 \times 512}$

We explored architecture of ***SwinIR*** (Transformer-based) (2 x) & (4 x) upsampling.

Loss function: mean absolute error (L1) $\mathcal{L}_{\ell_1}(\theta) = \mathbb{E}\big[\|\hat{I} - I^{\mathrm{HR}}\|_1\big].$

Evaluation metric: Peak Signal-to-Noise Ratio (PSNR)

$$\mathrm{PSNR} = 10 \log_{10}\Big(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\Big), \quad \mathrm{MSE} = \frac{1}{HW}\sum_{p=1}^{H}\sum_{q=1}^{W}(\hat{I}_{pq} - I^{\mathrm{HR}}_{pq})^2,$$

Where Max =225.

Assumptions:

- Perfectly aligned LR–HR pairs are available (via self-supervision using a pretrained Swin2SR model).
- Pixel intensities are normalized to [0,1] before feeding into networks.
- No additional noise degradation is applied beyond downsampling.

LITERATURE REVIEW

SwinIR: Image Restoration Using Swin Transformer (Liang et al., ICCV Workshops 2021)

Architecture Highlights:

- Shifted Window Attention: SwinIR builds on the Swin Transformer's pyramid design, partitioning feature maps into local windows where self-attention is computed. Windows are periodically shifted to enable cross-window connections, capturing both fine-grained details (within windows) and broader context (across windows).
- Residual-in-Residual Structure: Layers are organized into residual groups, each containing multiple Swin Transformer blocks plus a long skip connection—this facilitates stable training and information flow.
- Task-Agnostic Design: Though evaluated primarily on super-resolution, the same backbone—with minimal modification—handles denoising and JPEG artifact removal, demonstrating its versatility for image restoration tasks.

LITERATURE REVIEW

For swinIR super-resolution

Dataset	**Scale**	**SwinIR PSNR (dB}**	**Best CNN PSNR (dB}**	**ΔPSNR**
Set5	×2	38.42	38.00(RCAN)	+0.42
Urban100	×2	33.25	32.91 (RCAN)	+0.34
Manga109	×4	30.91	30.35 (RCAN)	+0.56

LITERATURE REVIEW

On *Set5* (×2 SR), SwinIR achieved 38.42 dB, surpassing the strongest CNN baseline (RCAN) by 0.42 dB. For Urban100, which contains many repetitive structures, it improved by 0.34 dB, underlining the benefit of windowed attention for architectural details. At 4x upscaling on Manga109, it gained 0.56 dB, showcasing strong texture recovery in line-art and cartoon styles.

Qualitative Insights:

SwinIR restorations exhibit crisper edges and fewer ringing artifacts compared to CNN methods. In dense-texture regions—e.g., foliage, brickwork—attention-based aggregation preserves pattern consistency without over-smoothing.

Relevance to Our Pipeline:

Enables high-fidelity enhancement of brush strokes and canvas texture after inpainting. Its task-agnostic backbone could, in future work, consolidate denoising alongside SR in a single model.

LITERATURE REVIEW

Resolution-Robust Large Mask Inpainting with Fourier Convolutions (Suvorov et al., CVPR 2022)

Fast Fourier Convolution (FFC) Module:

Dual-Branch Processing:
Splits feature channels into two pathways—one uses standard 3×3 convolutions (capturing local spatial cues), the other applies a learnable Fourier transform (capturing global, frequency-domain context).

Learnable Frequency Filters: Instead of fixed DFT kernels, FFC learns how to weight different frequency bands, tailoring global context aggregation to the inpainting task.

Multi-Scale Fusion: After separate processing, branches are fused via addition, ensuring that global scene structure and local texture jointly inform the prediction

LITERATURE REVIEW

On Places2 with masks covering up to 50% of the image, FFC-based inpainting lowered FID (Fréchet Inception Distance) from 12.4 to 9.8 and LPIPS (perceptual similarity) from 0.092 to 0.074. On FFHQ faces, FFC achieved FID 7.9 and LPIPS 0.062, outperforming LaMa by similar margins.

Qualitative Insights:

- Excels at filling very large holes (up to 60% of image area) with coherent global structure—e.g., reconstructing entire building facades or faces.
- Produces fewer colour bleeding artifacts at mask boundaries, thanks to the global frequency branch enforcing consistency across distant regions.

Relevance to Our Pipeline:

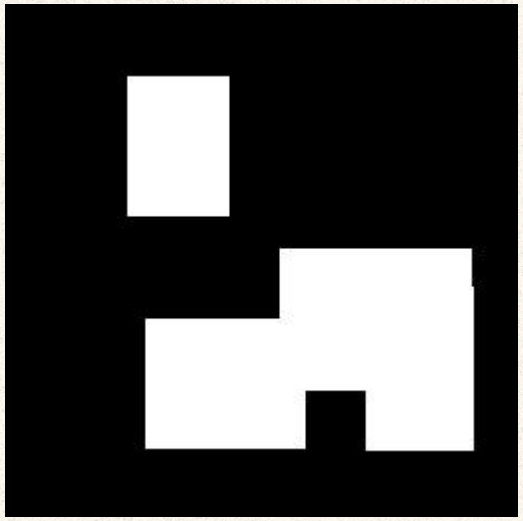
- Demonstrates that Fourier-based global reasoning can robustly handle even extreme damage masks—suggesting potential upgrades to our inpainting stage.
- The learned frequency filters may inform mask-aware weighting strategies in LaMa, further improving structural fidelity on intricate artworks.

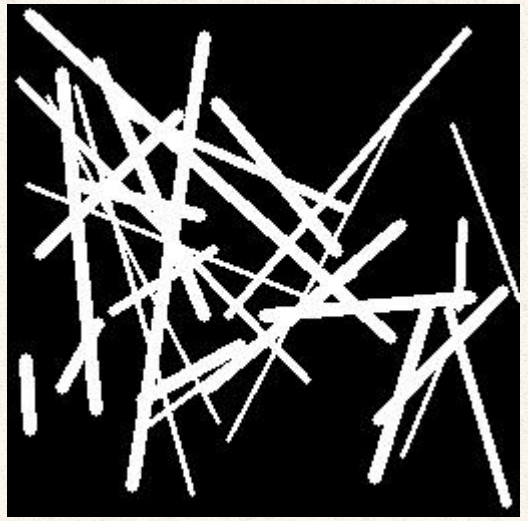
LITERATURE REVIEW

Lama ffc inpainting results

Dataset	Metric	LaMa	FFC(our)
Places2	FID ↓	12.4	9.8
Places2	LPIPS ↓	0.092	0.074
FFHQ	FID ↓	10.1	7.9
FFHQ	LPIPS ↓	0.085	0.654

IMPLEMENTATION





IMPLEMENTATION

LaMa – Image Inpainting with Fourier Convolutions

- Combines Fast Fourier Convolutions (FFC) with a deep encoder-decoder
- Focuses on global coherence and sharp local details

Core Architecture Components

Network Backbone

- **DownscaleBlock** (×3)
- **SpectralTransformBlock**
- **FFC (Fast Fourier Convolution)** block
- **InpaintingNet** (9 blocks)
- **UpscaleBlock** (×3)

Hierarchical Processing

Transitions image through low and high-resolution spaces for better semantic understanding and restoration.

IMPLEMENTATION

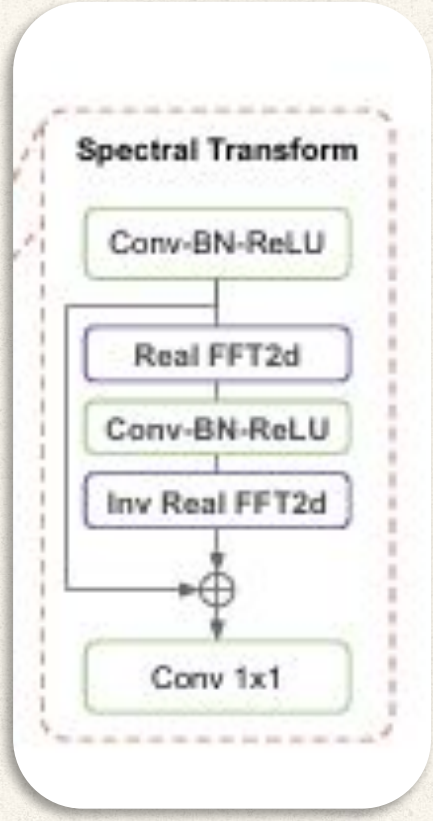
Spectral Transform Block

Goal: Capture Both Local and Global Patterns

- Operates in both **spatial** and **frequency** domains
- Uses **FFT** to extract frequency-based features
- Reconstructs image via **Inverse FFT** + **Residual connection**

Key Stages

- 1×1 Conv → BN + ReLU (pre-FFT)
- Real 2D FFT → Split Real & Imaginary
- 1×1 Conv on frequency parts
- Recombine + Inverse FFT
- Add back residual: $x_{\text{out}} = \mathcal{F}^{-1}(X_{\text{fft}}) + x_{\text{in}}$



IMPLEMENTATION

Fast Fourier Convolutions (FFC)

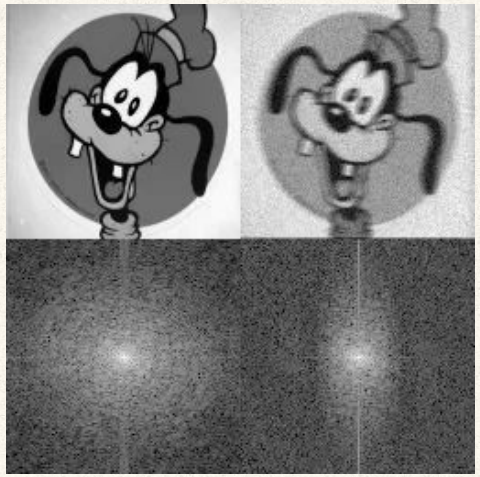
Dual-Pathway Design

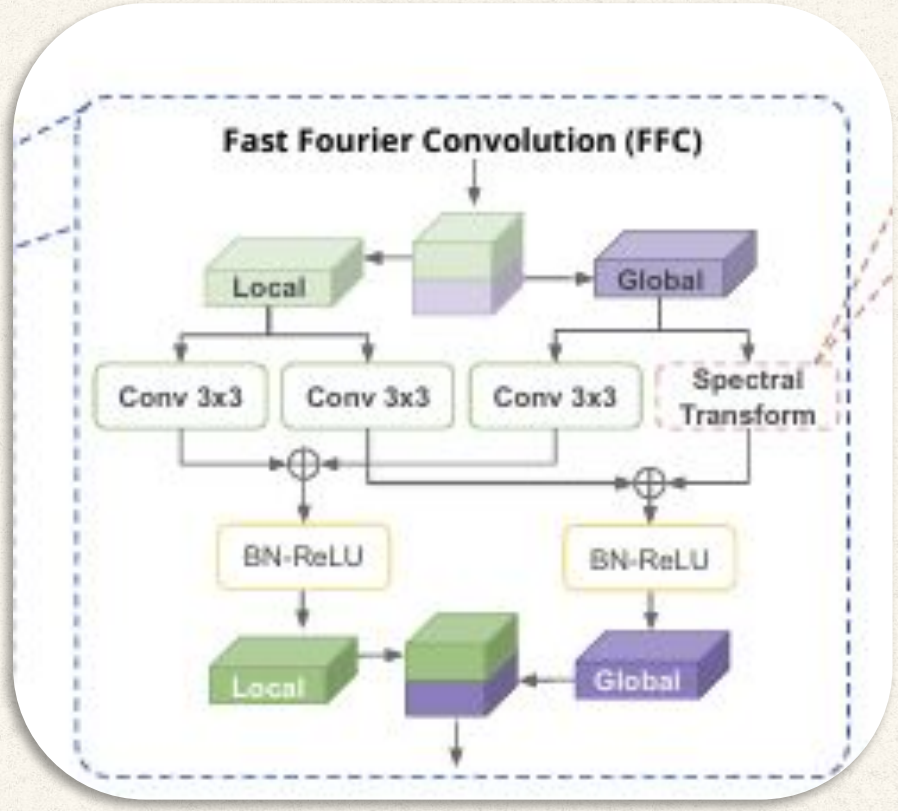
- **Local Path**: Standard spatial convs for fine details
- **Global Path**: Uses SpectralTransform for context

Fusion Process

1. Local: Conv → ReLU → Conv
2. Global: FFT → Transform → iFFT
3. Combine both paths for rich representation

Boosts performance on structured image tasks like inpainting & deblurring



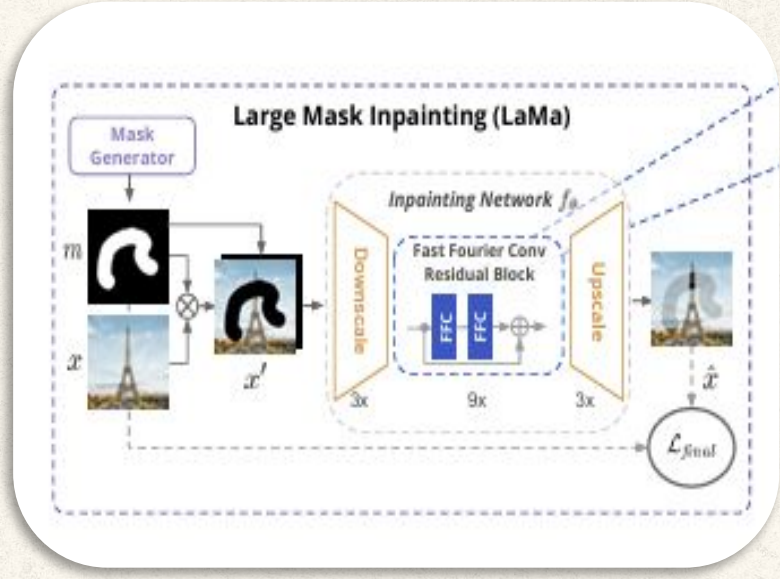


IMPLEMENTATION

Inpainting Network:

Reconstruct semantically and visually consistent content within masked regions.

- **Inputs**: Masked image + mask → 4 channels
- **Downsample**: 3× reduction for efficient processing
- **Transform**: 9 stacked **FFC residual blocks**
- **Upsample**: 3× restoration to original size
- **Output**: Final 3-channel completed image



Why It Work Benefits of LaMa Architecture

- FFC / FFTCaptures - **global structure**, low-freq info
- Down/Up Blocks Preserve - **hierarchical features**
- Spectral Transform Hybrid - **spatial + frequency** understanding
- Residual Connections - Retain original detail + efficient learning

IMPLEMENTATION

The Challenge of Image Inpainting

Why Is Inpainting Difficult?

- Inpainting is inherently ambiguous many plausible solutions for a masked region
- Only one ground truth, but multiple valid completions
- Pixel-based losses (L1, L2) fail to evaluate global and semantic correctness

LaMa's Approach

Combine multiple specialized loss functions to guide training

Total Loss Function $\mathcal{L}\text{total} = \kappa\,\mathcal{L}\text{adv} + \alpha\,\mathcal{L}_{\text{vgg}} + \beta\,\mathcal{L}_{\text{fm}} + \gamma\,\mathcal{L}_{\text{gp}},$

IMPLEMENTATION

Adversarial Loss ($\mathcal{L}_{\text{adv}}$) and Patch Discriminator

Adversarial Loss (AdvLoss)

- Uses PatchGAN to distinguish real vs fake patches
- Generator Loss :Encourages realism in generated areas
- Discriminator Loss: Learns to detect fakes
- Encourages high-frequency realism, especially in textures and edges

Patch Discriminator (PatchGAN)

- Operates on image patches, not full image
- Architecture: Convolutional layers + LeakyReLU + BatchNorm
- Outputs probability map for local realism
- Intermediate layers used for feature matching loss

IMPLEMENTATION

Perceptual & Feature Matching Losses

- **VGG-Based Perceptual Loss** $(\mathcal{L}_{\text{vgg}})$
  - Extracts features from pretrained VGG19
  - Computes ℓ2 distance between real vs fake feature maps
  - Captures high-level structural and semantic similarities
- **Feature Matching Loss** (L_{fm})
  - Uses PatchGAN's internal features
  - Matches intermediate feature maps of real vs generated images
  - Stabilizes GAN training & retains fine structure

Gradient Penalty $(\mathcal{L}_{\text{gp}})$

R1 Gradient Penalty

- Regularizes discriminator by keeping gradients close to 1
- Applied only on real images: : $\mathcal{L}\text{gp} = \lambda\, \mathbb{E}\hat{x} \sim P_r \left[\|\nabla_{\hat{x}} D(\hat{x})\|_2^2\right],$
- Prevents overfitting & improves training stability

Final Aggregated Loss

$$\mathcal{L}\text{total} = \kappa\, \mathcal{L}\text{adv} + \alpha\, \mathcal{L}_{\text{vgg}} + \beta\, \mathcal{L}_{\text{fm}} + \gamma\, \mathcal{L}_{\text{gp}},$$

IMPLEMENTATION

Loss Component Overview

- **Adversarial Loss** Forces realism at patch level
- **VGG Perceptual Loss** Preserves global semantics
- **Feature Matching Loss** Enhances structural consistency
- **Gradient Penalty** Stabilizes training & prevents overfitting

Epoch	L_adv	L_hrfpl	L_discpl
5	2.0520	2.1677	0.9677
15	2.0519	1.3646	0.7252
24	2.0521	1.1568	0.7255
34	2.0518	1.0629	0.6542
44	2.0518	0.9843	0.6412
54	2.0517	0.8858	0.6420
75	2.0515	0.8319	0.6227

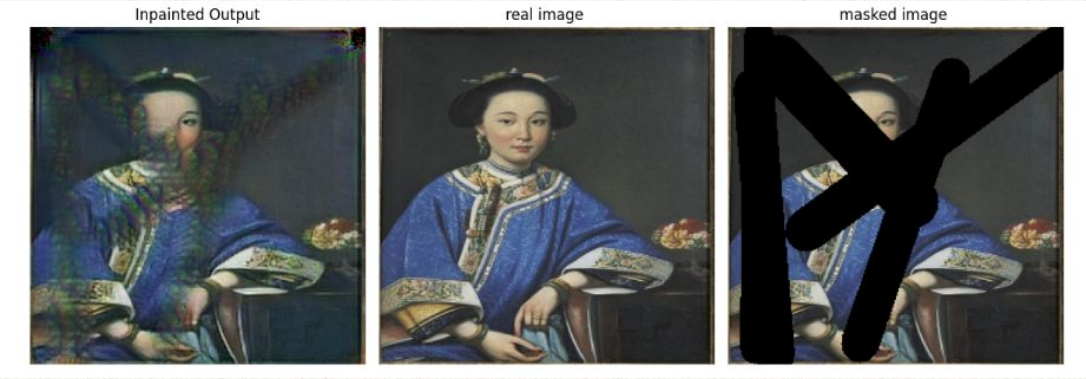
Training Philosophy

By targeting local textures, global semantics, and regularization together, LaMa achieves visually coherent, realistic, and stable inpainting results.

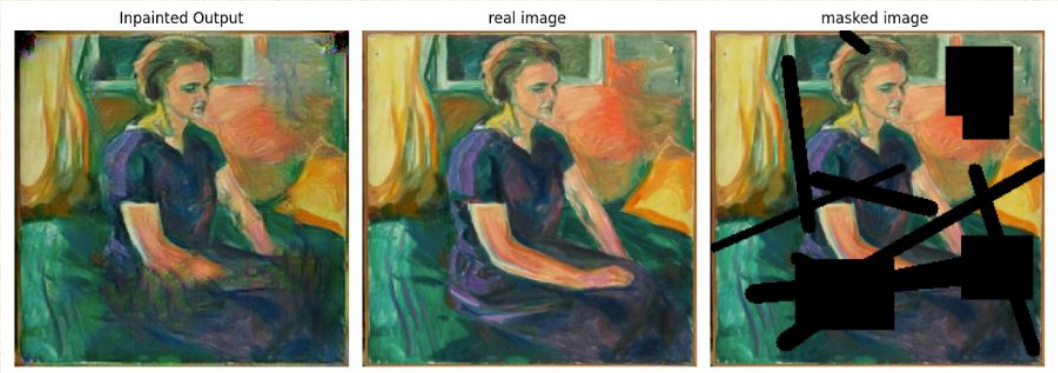
Results

Large Box

Narrow



Large Wide



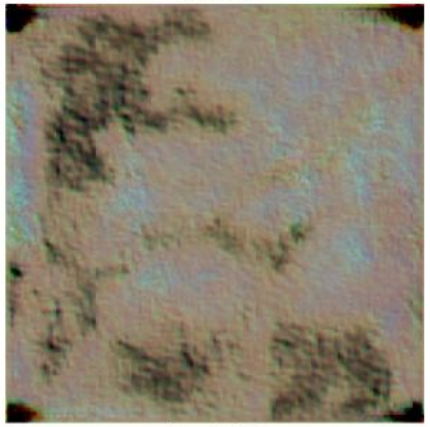
Deepfill

Results

Original

Masked

Epoch 0



Epoch 5

Epoch 15

Epoch 24

Epoch 34

Epoch 44

Epoch 54

Epoch 75

Results

Mask Type	Scope	Key Behaviors	Artifacts
Large Scribbles (Deep-fill blobs)	Global	Guesses broad textures (e.g., foliage)	Blurry fills, repetitive “ripple” patterns, color/texture shift
Mid-sized Blobs (Semantic regions)	Semi-global	Recovers coarse shapes (e.g., torso, arm)	Soft edges, color banding, fine detail loss
Thin Scratches (Narrow masks)	Local	Near-perfect line/tone interpolation	Seamless results — no visible artifacts
Wide Rectangular Hole	Global	Hallucinates rough forms (e.g., branches, fountain)	Warped geometry, color speckles, blurred cross-hatching

Inpainting Type	FID ↓	Inception Score ↑
Large Wide (Only)	132.45	2.55 ± 0.37
All Combined	100.28	2.82 ± 0.55
DeepFill (Only)	126.96	2.39 ± 0.26
Narrow (Only)	**83.10**	**2.96 ± 0.40**

Implementation

SwinIR Architecture

SwinIR consists of 3 stages:

Shallow Feature Extraction

A single convolutional layer maps the low-quality (LQ) image to an initial feature space:

$$F_0 = H_{SF}(I_{LQ})$$

Where:

H_{SF}: 3 Conv layer

F_0: Initial extracted features

Deep Feature Extraction

Multiple Residual Swin Transformer Blocks (RSTBs), followed by a convolutional layer:

$$F_{DF} = H_{DF}(F_0) = H_{CONV}(H_{RSTB_K} \circ \cdots \circ H_{RSTB_1}(F_0))$$

Where:

K: Number of RSTBs

H_{DF}: Deep feature extraction module

$\circ$: Function composition

Image Reconstruction

Combines initial and deep features for final high-quality output using PixelShuffle (for upsampling):

$$I_{HQ} = H_{REC}(F_0 + F_{DF})$$

In residual learning form:

$$\hat{I}_{HQ} = H_{SwinIR}(I_{LQ}) + I_{LQ}$$

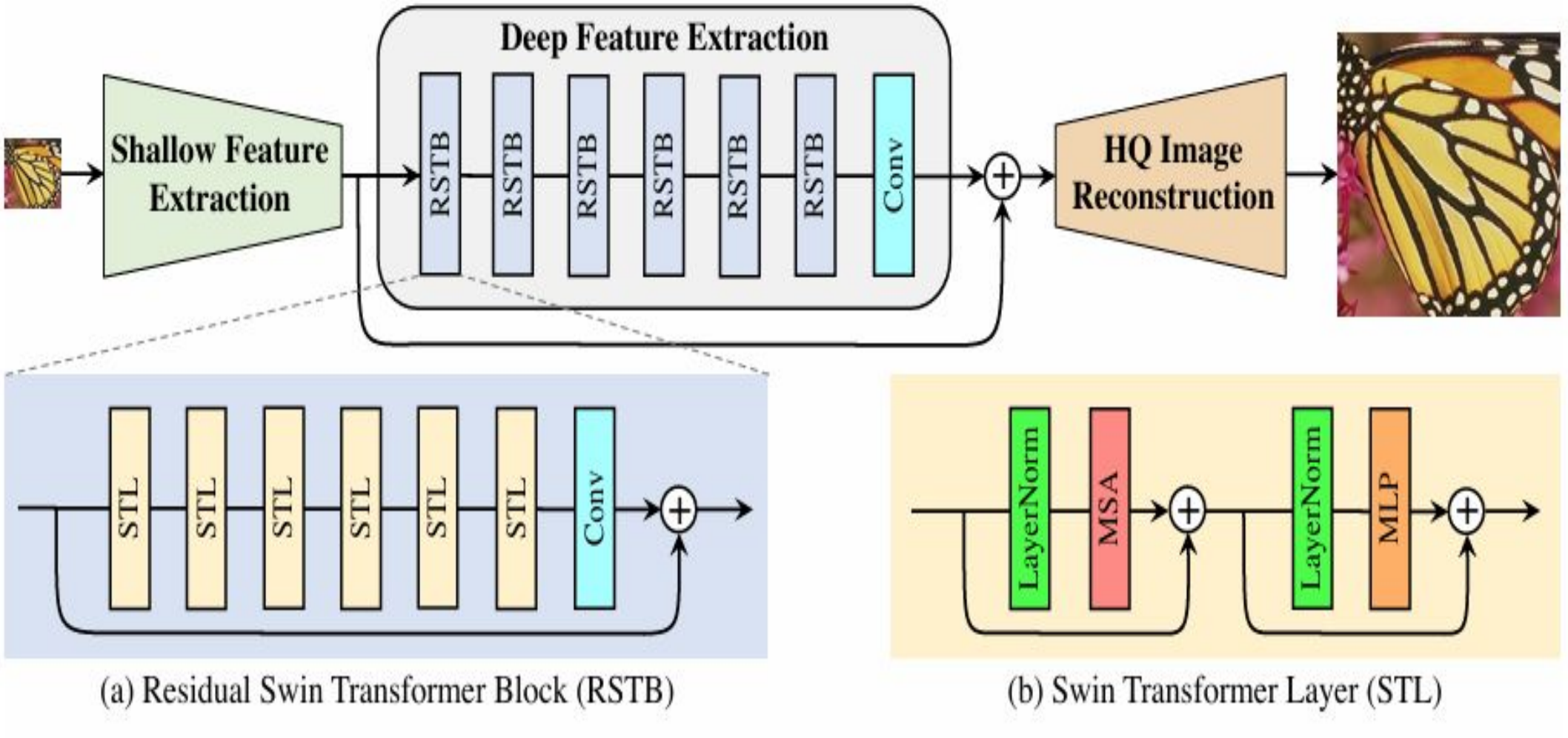


Figure 2: The architecture of the proposed SwinIR for image restoration.

Implementation

Residual Swin Transformer Block (RSTB)

Each RSTB contains L Swin Transformer Layers (STLs) and a residual convolutional connection.

Input

$$F_{i,0} \quad \text{(input to the } i\text{-th RSTB)}$$

Step 1 – Swin Transformer Layers

$$F_{i,j} = H_{\mathrm{STL}_{i,j}}(F_{i,j-1}) \quad \text{for } j = 1, \ldots, L$$

- $H_{\mathrm{STL}_{i,j}}$: the j-th Swin Transformer Layer in block i
- $F_{i,j-1}$: the output of the previous STL

Step 2 – Convolution + Residual Connection

$$F_{i,\mathrm{out}} = H_{\mathrm{Conv}_i}(F_{i,L}) + F_{i,0}$$

- H_{Conv_i}: convolution in block i
- $F_{i,L}$: final STL output
- $F_{i,0}$: original input to block
- $F_{i,\mathrm{out}}$: output of RSTB

Swin Transformer Layer (Part 1: Local Attention)

The Swin Transformer Layer introduces localized attention and windowing mechanisms to enhance efficiency and scalability.

Input Reshaping:

- Input shape: $H \times W \times C$
- Partitioned into $M \times M$ windows
- Number of windows: $\frac{HW}{M^2}$
- Reshape each window to $M^2 \times C$

Local Self-Attention:

- For a window $X \in \mathbb{R}^{M^2 \times C}$:

$$Q = XP_Q, \quad K = XP_K, \quad V = XP_V$$

- P_Q, P_K, P_V: shared projection matrices
- $Q, K, V \in \mathbb{R}^{M^2 \times d}$
- Attention:

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V$$

- B: learnable relative positional encoding

Multi-Head Attention (MSA):

- Attention is computed in h parallel heads
- Results are concatenated for final output

Implementation

Swin Transformer Layer (Part 2: MLP, Residuals, Shifted Windows)

Feed-Forward Network (MLP):

- Composed of two linear layers with GELU activation in between.

Normalization and Residuals:

- LayerNorm applied before both MSA and MLP
- Residual connections are added post each module:

$$X = \text{MSA}(\text{LN}(X)) + X$$

$$X = \text{MLP}(\text{LN}(X)) + X$$

Shifted Window Mechanism:

- To allow cross-window interaction, alternate layers shift window positions
- Shift by $(\frac{M}{2}, \frac{M}{2})$ pixels before window partition
- Enables global connectivity while retaining efficiency

Intuition Behind RSTB + STL

Combines locality (Conv) + non-local modeling (Transformer)

Shifted windows improve receptive field without excessive computation

RSTB enables hierarchical feature aggregation with skip connections

Pixel Loss for Super-Resolution (SR)

For classical and lightweight image SR, SwinIR is trained using L1 loss, which measures absolute pixel-wise error:

$$L_{\text{pixel}} = \|\hat{I}_{\text{HQ}} - I_{\text{HQ}}\|_1$$

Where:

$$\hat{I}_{\text{HQ}} = H_{\text{SwinIR}}(I_{\text{LQ}})$$

$\hat{I}_{\text{HQ}}$: Predicted high-quality image

I_{HQ} : Ground-truth high-quality image

Reason: L1 loss encourages sparsity and is more robust to outliers than L2, while also avoiding excessive blurring common with L2.

Evaluation Metric: PSNR (Peak Signal-to-Noise Ratio)

While training uses L1/Charbonnier loss, model performance is evaluated using PSNR:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right)$$

Where:

MAX: Maximum possible pixel value (e.g., 255)

MSE: Mean Squared Error between predicted and target images

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{I}_i - I_i)^2$$

Interpretation:
Higher PSNR $\Rightarrow$ Better image quality (closer to ground truth)

Swin-Ir Trainning

SwinIR Super-Resolution Pipeline

Data Preparation: LR-HR pairs were created by using a prebuilt SwinIR, whose weights were used to create hr image.

Network Architecture

- **Shallow Embedding:** 1×1 conv → 96-d features
- **Residual Swin Transformer Body:**
  - 2× RSTBs, each with 6 Swin Transformer layers (depths=[6,6], heads=[6,6], window=8)
  - 3×3 conv + residual skip at end of each RSTB
- **Upsampling Head:**
  - 3×3 conv expands channels ×4 → PixelShuffle(2) → 256→512
  - Final 3×3 conv → RGB output
- **Training Config**

Loss: L1 (edge-sharpness, robustness)

Optimizer: Adam, LR = 2×10^{-4}, batch size = 8

Validation: PSNR on 10 samples each epoch, checkpoint best

From Prototype to Final Model

Aspect	Prototype Variant	Final Model
Depths / Heads	[4,4], heads=[4,4], window=4	[6,6], heads=[6,6], window=8
Optimizer / Scheduler	AdamW + OneCycleLR (warmup→2e-4, cosine)	Adam (fixed LR=2e-4)
Loss Functions	Charbonnier → MSE	L1 only
Outcome	Fast early PSNR gain, late instability	Stable training; best PSNR at epoch 38
Notes	Strong start, converged early but unstable	Sharp edges, consistent PSNR improvements

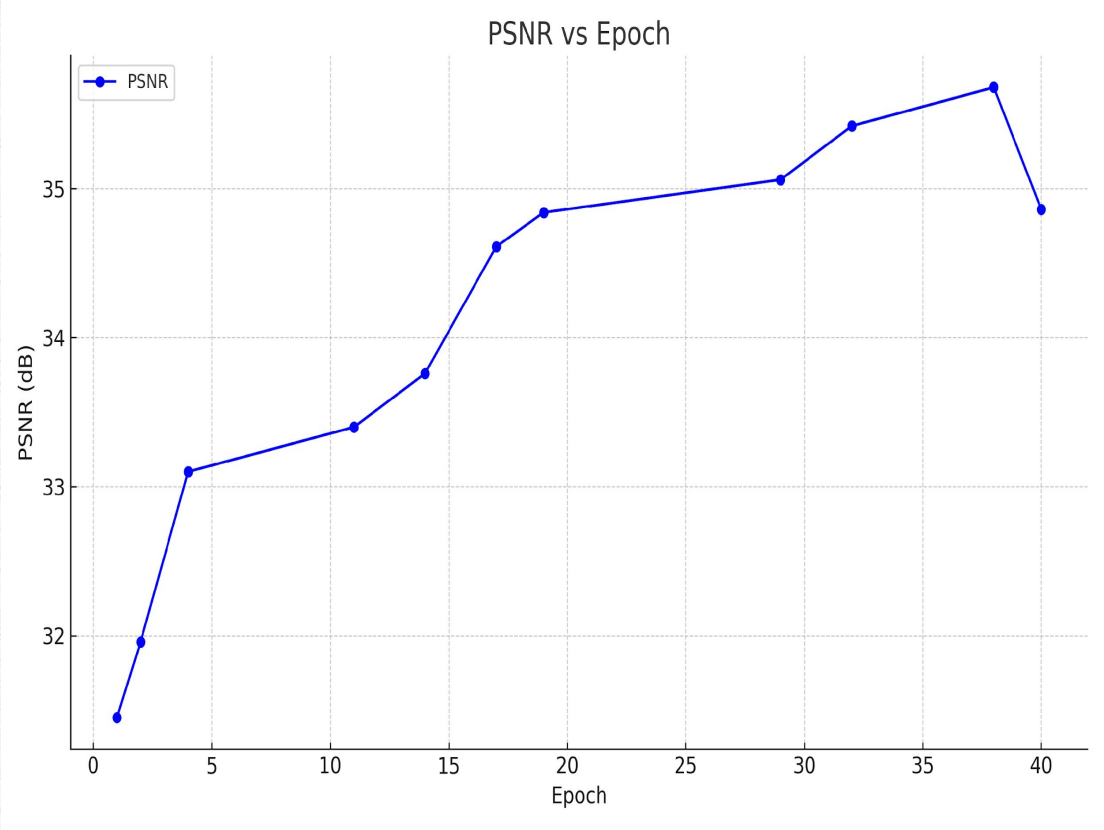
Ablation Study

- OneCycleLR boosted initial updates but added complexity
- L1 loss correlated best with PSNR and produced the sharpest textures
- Simplification removed late-epoch oscillations

Training Results & Qualitative Comparisons

Performance Over Epochs & Sample Outputs

Epoch	Avg. L1	PSNR (dB)
1	0.0273	31.45
4	0.0148	33.40
11	0.0128	34.61
29	0.0130	35.06
38	0.0124	35.68 ★
40	0.0134	34.86



Results of Swin-IR

Low-resolution image

High resolution image

Results of Total Pipeline

Lr image

Hr image

Sr image

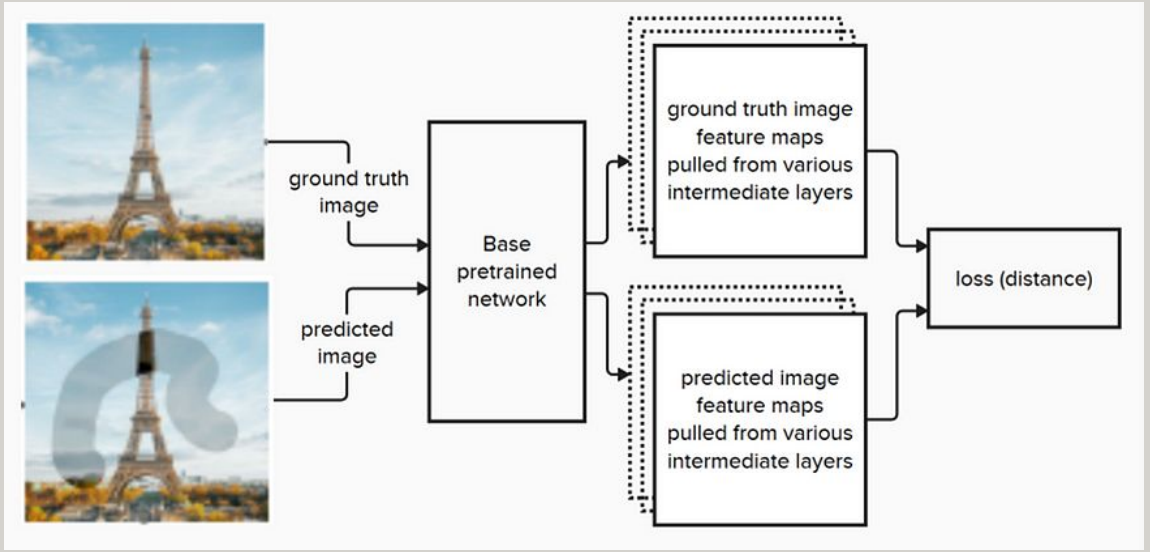
Learnings

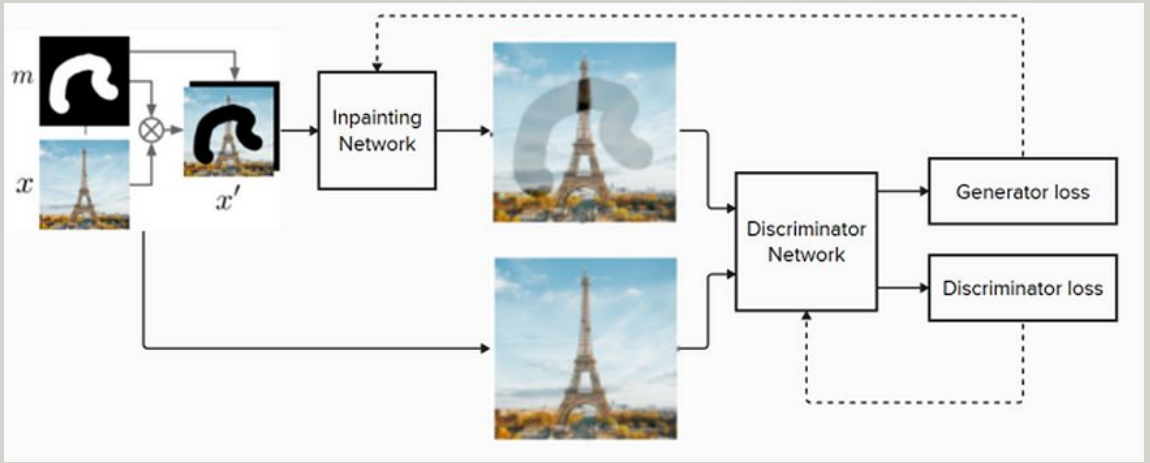
- **Deepened Theoretical Concepts Through Practice**
- Observed model convergence firsthand (PSNR from 23.40 dB to 29.04 dB) and balanced multiple loss terms (adversarial, perceptual, feature-matching) for stable training
- Managed data diversity (mask types, image resolutions) to improve generalization beyond textbook examples
- **Robust End-to-End Pipeline Development**
- Designed a three-stage workflow—data merging, analysis, resizing—and handled edge cases for images
- Modularized complex architectures into reusable components, enhancing maintainability and enabling rapid experimentation
- **Enhanced Collaboration & Project Management**
- Led a 4-member team with clear task division,regular syncs to meet deadlines
- Prioritized stable baselines over unproven ideas, balancing innovation with deliverable commitments
- **Professional Communication & Reporting**
- Crafted concise, structured documentation with tables/figures for loss trends and evaluation metrics
- Presented findings clearly, translating complex results for diverse audiences
- **Resource & Future-Readiness Strategies**
- Optimized architectures and batch sizes to fit GPU constraints, gaining hardware-aware design insights
- Outlined next steps—learning-rate schedules, expanded validation, to drive continual improvement

Contributions

- **M Ganesh (*AI22BTECH11017*)** – Lama inpainting model
- **K Aditya(*AI22BTECH11013*)** - Lama inpainting model
- **Ch Kushwanth(*AI22BTECH11006*)**- swinIr model
- **T Keshavardhan (*AI22BTECH11029*)**-swinIr model

"The object of restoration is to repair the work of art without falsifying it."— *Cesare Brandi*


ground truth image
predicted image
Base pretrained network
ground truth image feature maps pulled from various intermediate layers
predicted image feature maps pulled from various intermediate layers
loss (distance)


m
x
x'
Inpainting Network
Discriminator Network
Generator loss
Discriminator loss