Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Self-Supervised Semantic Prior Generative Adversarial Network for Heritage Image Restoration

Suma N.[1], Kusuma Kumari B M.[2]

Department of Studies and Research in Computer Applications, Jnanasiri Campus, Tumkur University, Tumkuru, India.

E-mail: [1]myphdpapers2@gmail.com, [2]kusuma_bm@outlook.com

Orcid ID: [1]0009-0008-9300-4348, [2]0000-0003-2961-7752

Abstract

The heritage images are subjected to large and irregular missing regions due to long-term degradation. In order to repair the gaps, a filling process is used which creates visible seams. High computational expense and prolonged inference time are involved in diffusion and transformer models. A framework for restoration based on GAN is built to handle the irregular missing regions while preserving delicate brush strokes and structural lines with faster speed. The proposed framework makes use of the Adaptive Semantic Prior Injection (ASPI), Topology-Aware Semantic Refinement (TASR) and Entropy-Guided Self-Supervision (EGSS). ASPI contextualizes guidance while TASR provides line and edge continuity, and EGSS emphasizes uncertain regions during training. SSPL learns semantics from masked images with no labels and is used to aid the performance of irregular restoration. The Kaggle Damaged Paintings dataset is used to train the model with an image resolution of 256×256 for 50 epochs. Masked WikiArt images are utilized for self-supervised prior learning. The Indian Heritage Monuments pictures are validated for model performance. The model recorded a Peak Signal-to-Noise Ratio (PSNR) of 32.6 dB, a Structural Similarity Index Measure (SSIM) of 0.92, and a Fréchet Inception Distance (FID) of 18.7. According to the results, diffusion and transformer-based baselines are consistently outperformed. Addditionally, more texture continuity is gained when compared to exemplar based inpainting and Conditional GAN (cGAN) using partial convolution-based models. The ablation study shows that TASR mainly promotes structural fidelity while ASPI and EGSS provide additional benefits. The architecture remains efficient with 34.8 million parameters, 108.5 Giga Floating Point Operations (GFLOPs), and an inference time of 0.118 seconds per image Despite the improvements, the model remains unable to manage thin cracks and the edges of the restored region can hardly be seen. Additionally, significant omissions may not completely correspond to the original in the final restoration. Future studies will focus on training at greater resolutions and applying diffusion priors appropriately.

Keywords: Heritage Image Restoration; Image Inpainting; GAN; Semantic Prior; Topology-Aware Semantic Refinement; Entropy-Guided Self-Supervision; Adaptive Semantic Prior Injection; Self-Supervised Semantic Prior Learning.

DOI: https://doi.org/10.36548/jiip.2026.2.001
Received: 16.02.2026, received in revised form: 14.03.2026, accepted: 28.03.2026, published: 08.04.2026

1. Introduction

Heritage paintings are among the most valuable treasures of human civilization. Safeguarding these historical paintings is therefore important. However, over a certain period of time, as they age,their look and quality degrade due to continuous exposure to sun, rain, and human activities, including the aging of the materials used to create them. This causes faded pigmentation,cracks, and scratches, which lead to fading of edges, textures, and cuts in the brushstrokes, resulting in irregular damage and small or large missing regions on the painting.

Earlier traditional methods used for restoration were designed for a single type of painting and could not be applied to all types of restoration work. Legacy methods took longer processing times and failed to restore delicate art and structural details. Restoration of heritage paintings is not an easy task.The challenges of digital restoration include regenerating missing or damaged content, which varies in size, preserving delicate details like brush strokes, thin lines, and edges, and matching the fine textures with the existing ones that define the artistic style and historical identity of the artwork.These visual elements are central to authenticity and historical credibility.

Deep learning contributes to improved digital restoration. Diffusion-based models can handle texture degradation very well and have been widely adapted for restoring paintings [1]. Attention-driven U-Net architectures improve contextual awareness through stronger structural guidance while inpainting [2]. GAN-based methods have been extended into hierarchical coarse-to-fine pipelines to recover large and irregular missing regions [3]. Repeated texture patterns are very common in large historical paintings. Frequency-guided diffusion techniques have been applied for restoration [4]. To improve structural continuity and visual plausibility, spatial geometry-aware methods have been used [5]. The majority of the research on inpainting used deep learning techniques such as GANs, transformers, and diffusion models for the restoration process [6].

Some limitations still exist. Many of the existing models do not incorporate semantic priors in a proper way to support smooth transitions between damaged and intact regions. They also do not include adaptive mechanisms for highly unpredictable or badly degraded areas. One of the most challenging parts is preserving fine topological structures. Any variation in the generated brush strokes can reduce the authenticity of the restored output. This highlights the importance of semantic understanding of the heritage data and the customization of restoration models to adapt to the underlying context. This study introduces a Self-Supervised Semantic Prior GAN framework to combine semantic priorities with context-aware feature guidance along with topology-aware refinement.

The proposed framework contains three main modules. The first module Topology-Aware Semantic Refinement (TASR) takes care of restoring fine structural details and maintains the continuity of the restored regions. The second module, Entropy-Guided Self-Supervision (EGSS) focuses more on understanding the unexpected and severely damaged regions by applying entropy- -based auxiliary supervision. The third module, Adaptive Semantic Prior Injection (ASPI), supplies semantic prior information to the generation process of the GAN to make the restored image look more natural and coherent with its surrounding context. This is an integrated framework consisting of a self-supervised semantic prior encoder, adaptive prior injection inside the GAN generator, a topology-aware refinement block, and an entropy-guided supervision mechanism. The main objective of this study is to build and evaluate a self-supervised GAN-based framework for heritage image restoration that improves

structural preservation, semantic consistency, and restoration quality. The specific objectives are:

- Develop a self-supervised GAN framework that leverages semantic priors for heritage image restoration.
- Introduce TASR for preserving cracks, edges, and brushstroke continuity.
- Propose EGSS for adaptively addressing regions with high uncertainty and severe damage.
- Design ASPI for effective integration of semantic knowledge into the generative process.
- Establish a robust training strategy suitable for heritage datasets with limited annotated data.
- Validate the framework against state-of-the-art GAN, Transformer, and diffusion-based methods through comprehensive experiments.

The proposed heritage painting framework can be used along with traditional digital restoration practices. The digitally restored images help conservators judge the level of damage, prepare reports for comparative visual analysis, and prepare the artifacts for museum exhibitions and digital archives. The framework helps experts make decisions during the restoration process alongside existing restoration methods. Section 2 presents the highlights of related work on heritage image restoration and deep learning-based inpainting. Section 3 explains the proposed methodology and algorithm. Experimental results with discussion and benchmarking with recent previous work are explained in Section 4. The conclusion of the study, its contributions, limitations, and future work are given in Section 5.

2. Related Work

Heritage-specific restoration has been widely studied using GAN-based models. An edge- and line-guided diffusion patch GAN was introduced for restoring damaged temple murals using edge maps as structural guidance [7]. This method improved shape recovery and reduced FID compared to standard GANs. However, it failed in its performance, and efficacy dropped in severely damaged regions where edge cues were unreliable. Conditional GANs combined with partial convolutions were proposed to address irregular missing areas in mural restoration [8]. A dual-branch restoration model was proposed later, with separate branches for structure reconstruction and texture enhancement. This approach improved SSIM to around 0.88. and incomplete inputs were handled wisely. But due to the lack of semantic guidance, the restored regions still remained blurry [9]. This design improved consistency between geometry and texture. Due to limited semantic integration generalization across different mural styles was reduced. Sketch-guided restoration using SGRGAN relied on rough sketches to guide the recovery of Chinese landscape paintings [10].

A heterogeneous GAN framework using multiple feature streams was proposed to improve reconstruction stability under varied damage conditions. This method relied on very high-quality rare sketches which are practically not found easily in the heritage world. This helped to improve edge alignment. [11]. Learning-based digital restoration has also been explored beyond standard GAN models. AI-driven 3D restoration was applied to museum

artifacts for virtual display. These techniques improved robustness, but this increased model complexity and raised training costs. It failed to restore complex degradation patterns [12]. A structure-aware multi-view inpainting model with dual consistency attention improved geometric continuity across views. This approach was effective for three-dimensional objects but not applicable for two-dimensional murals and paintings. [13]. Several survey studies provide broader context for image inpainting. One review covered GAN-, Transformer-, and diffusion-based inpainting methods, along with commonly used datasets and benchmarks. This method relied on the availability of multiple views of the mural. However, in practice, a single image of the heritage mural will be available and hence this was not suitable for real-time [14].

Another survey on transformer-based inpainting and video completion highlighted the role of attention in modeling long-range dependencies. This study did not address the specific structural and semantic challenges of heritage images [15], and it did not consider heritage-specific requirements such as preserving cracks, brush strokes, and material texture. General inpainting methods have an influence on heritage restoration. A bidirectional CNN–Transformer framework improved the balance between local detail recovery and global context reasoning [16], but it lacked domain-specific priors required for heritage data. GAN inversion combined with autoencoders was proposed to improve semantic coherence by refining incomplete inputs in latent space [17]. This method worked well under moderate damage but degraded under severe or irregular damage. A U-Net-based super-resolution model addressed information loss across layers using multi-level information compensation. This improved fine-detail recovery of the damaged input [18], rather than focusing only on direct inpainting and this made it useful for refining restored outputs.

In general, earlier research has shown that deep learning has improved image inpainting and heritage restoration. GAN models contributed better structural recovery through partial convolutions by dealing with missing irregular areas. Diffusion methods contribute to better texture quality. Designs based on Transformers made it easier to understand the big picture. However, three gaps are identified. Most methods do not consider the semantic priorities into account that are needed for restoration which is consistent with history. For all input murals, the damaged areas are treated same way with no adaptive supervision for areas that are very uncertain. Fine topological details like cracks, strokes, and edge continuity are not always preserved, which makes the restored results less realistic. The current study addresses existing gaps by proposing a cohesive restoration framework that incorporates semantic priors, adaptive supervision, and topology-aware refinement via a self-supervised semantic prior GAN with entropy-guided supervision for heritage image restoration.

3. Proposed Work

3.1 Problem Definition

This problem can be formulated as learning a restoration function G that reconstructs the undamaged image I_{restored} from the damaged input image I_{damaged}, the binary damage mask M, and a semantic prior P_{semantic}. The mask M assigns a value of 1 to missing or unreliable pixels and a value of 0 to intact pixels, and $\odot$ denotes element-wise multiplication. In this setting, the network receives only the observed part of the image $(1 - M) \odot I_{\text{damaged}}$, while the mask and semantic prior guide the filling of the missing regions. The semantic prior P_{semantic} is obtained from a self-supervised encoder trained on masked paintings, and it supplies contextual

cues that guide reconstruction in missing regions. This formulation makes the mask M an explicit input to the generator, so that later modules such as TASR and EGSS can use the same mask to focus their refinements on the degraded areas.

$$I_{\text{restored}} = G\left((1 - M) \odot I_{\text{damaged}},\ M,\ P_{\text{semantic}}\right) \quad (1)$$

The proposed architecture, Self-Supervised Semantic Prior GAN (SSSP-GAN), is developed to address both semantic coherence and structural preservation in historical image restoration. The system processes a damaged input image using semantic prior extraction, adaptive prior injection, topology-aware refinement, and adversarial synthesis to create the restored output image, The generator is led by learnt semantic priors, and it is adjusted using topology-aware restrictions. Adversarial learning ensures visual realism in restored sections. Figure 1 depicts the whole process of the SSSP-GAN system.

3.2 Self-Supervised Semantic Prior Learning (SSSPL)

An important part of the technique is using self-supervised learning to obtain semantic priors before restoration. In Kaggle's Damaged Paintings dataset, paired ground-truth images are limited, which makes supervised training difficult. This is addressed by using masked reconstruction as the main self-supervised objective, allowing the GAN to learn structure from incomplete inputs. Self-supervised ideas from image inpainting, deep image priors, and plug-and-play regularization are used to support the reconstruction of missing regions without explicit labels [19].

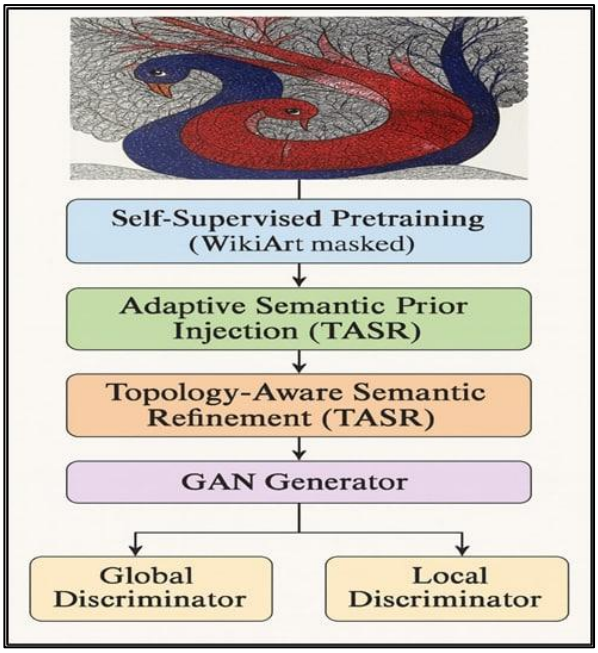

Figure 1. SSSP-GAN Framework

Pretext tasks such as jigsaw puzzle prediction and image colorization on a large set of masked WikiArt paintings are applied during pre-training to strengthen the model's semantic understanding. In the jigsaw task, shuffled patches are rearranged to their original positions to learn global spatial coherence. The colorization task ensures consistency in texture and tonal appearance by converting grayscale artworks back into color. Together, these tasks help the

network learn artistic context and structure, which later guides restoration. The self-supervised objective combines these two pretext tasks as shown below:

$$L_{ssp} = L_{jigsaw} + L_{color} \quad (2)$$

where L_{jigsaw} enforces spatial reasoning and L_{color} enhances semantic consistency. This pretraining phase ensures that the model captures both structural and contextual cues necessary for restoring damaged heritage images. During self-supervised pretraining, the total loss L_{ssp} in Eq. (2) is also monitored on a small validation split, and training is stopped when the validation loss shows no improvement over several consecutive epochs; the encoder checkpoint with the lowest validation loss is then retained for the subsequent GAN-based restoration stage. The step-by-step process for learning the semantic prior from WikiArt images is outlined in Algorithm 1.

Algorithm 1: Self-Supervised Semantic Prior Learning (SSSPL)

Input: WikiArt masked paintings dataset, encoder network E, irregular mask generator, training/validation split.

1. Initialize encoder $E$and optimizer.
2. Split dataset into training set $D_{tr}$and validation set D_{val}.
3. For epoch $e = 1 \ldots E_{max}$do
   3.1 Shuffle training images in D_{tr}.
   3.2 For each image $I \in D_{tr}$do
   3.2.1 Generate irregular mask $M$and form masked input $I_m = I \odot (1 - M)$.
   3.2.2 Create a jigsaw input by shuffling fixed-size patches from I_m.
   3.2.3 Convert $I_m$to grayscale for the colorization task.
   3.2.4 Pass the jigsaw and grayscale inputs through encoder $E$and the two task heads.
   3.2.5 Compute $L_{jigsaw}$and L_{color}.
   3.2.6 Compute the self-supervised loss using Eq. (2): $L_{ssp} = L_{jigsaw} + L_{color}$(2)
   3.2.7 Update encoder parameters by minimizing L_{ssp}(backpropagation and optimizer step).
   3.3 End for
   3.4 Compute validation loss L_{ssp}^{val}on D_{val}using the same Eq. (2) definition and store it for the current epoch.
4. End for
5. Select the checkpoint with minimum L_{ssp}^{val}as E_{best}.
6. Semantic prior extraction: Freeze E_{best}, pass each damaged input image through it, and apply global average pooling on the final feature map to obtain semantic prior embedding $P_{semantic}$.
7. Output: Trained encoder $E_{best}$and semantic prior embedding $P_{semantic}$used in the ASPI module.

The process of Self-supervised semantic prior learning is shown in Figure 2. In this stage, damaged images are used for two pretext tasks namely jigsaw prediction and grayscale colorization to learn contextual embeddings. Jigsaw prediction helps the encoder learn spatial arrangement and shapes from shuffled patches, while the colorization task encourages it to capture tone and color relationships in old paintings. Together, these two tasks are provide

structural and color priors that fit the requirements of cracked and faded heritage images and support stable training with limited data. The embeddings learned from the semantic prior through these two tasks used in later restoration stages. In this model, the semantic prior $P_{\text{semantic}}$is obtained by passing the image through the self-supervised encoder and applying global average pooling on the final feature map. The output is a simple d-dimensional vector, and this vector acts as the contextual embedding for the ASPI block during restoration.

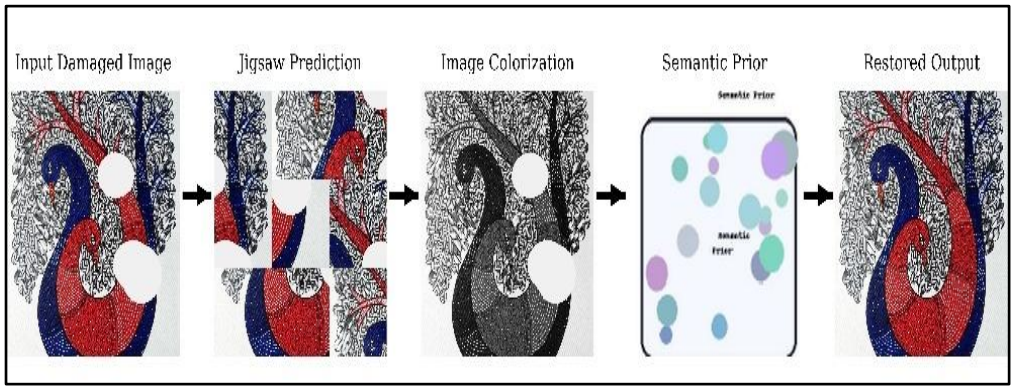

Figure 2. Self-Supervised Semantic Prior Learning

3.3 Adaptive Semantic Prior Injection (ASPI)

The ASPI module integrates contextual semantic priors directly into the generator bottleneck. ASPI applies an attention-based mechanism that adaptively balances generator features with the learned semantic prior instead of using a fixed fusion rule. This helps maintain global semantic consistency while preserving fine details. This work follows the usual restoration order by first fixing the structure, then refining the texture, and finally aiming for both accurate pixels and visually realistic results [20]. It uses the Damaged Paintings dataset for learning real crack and fading patterns, and it uses masked WikiArt images for diverse pretraining. The fused bottleneck representation is defined as:

$$F' = \alpha \,.\, F_{gen} \;\; + \; (1 - \alpha).\, P_{semantic} \tag{3}$$

where F_{gen} represents the generator bottleneck features, $P_{semantic}$ denotes the semantic prior embedding, and $\alpha \in [0,1]$ is an attention coefficient learned dynamically during training. The coefficient α is generated by a lightweight attention branch that takes the bottleneck features as input. The weights of this branch are updated by backpropagation during training. The α is completely learned from data and allowing it to vary across spatial locations in the bottleneck layer. The attention branch follows the same initialization and optimizer settings as the other convolutional layers. The value of α is not manually tuned at any stage, and the training process remained stable with this setup.

This adaptive weighting allows the network to assign greater weight to the semantic prior when restoring severely corrupted regions, while it depends more on the generator features in the intact or less damaged areas. In practice, the fusion rule in Eq. (3) works like a weighted average that mixes the generator features and the semantic prior at each spatial location, where the attention branch predicts the weight α. The architecture brings the semantic prior into the generator bottleneck through the Adaptive Semantic Prior Injection (ASPI) module, as shown in Figure 3. This attention-based block helps to balance semantic information with the generator features in an adaptive way, so that global consistency is maintained while the corrupted regions are refined.

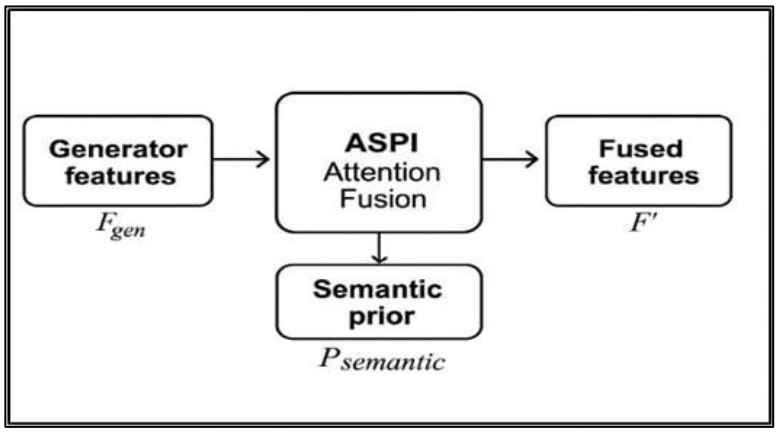

Figure 3. Adaptive Semantic Prior Injection (ASPI) Block

3.4 Topology-Aware Semantic Refinement (TASR)

The TASR module is designed to preserve structural continuity in damaged regions, especially along cracks, strokes, and edges. The feature map is represented as a weighted graph. The pixels or super pixels are treated as nodes and the edges connect to neighboring nodes based on spatial proximity and structural similarity. This helps maintain elongated patterns and avoids breaks in restored motifs. An edge map is extracted from the damaged image using the Canny operator with automatic threshold selection, followed by non-maximum suppression and thinning to retain dominant crack and contour pixels. This refined edge map guides the graph construction for topology-aware refinement. The model follows the structure-guided GAN principles [21]. These structural cues support edge and contour continuity during restoration so that cracks, outlines, and geometry remain coherent.

The regularizer is defined using the graph Laplacian. Each pixel i in the decoder feature map is treated as a node, and its 4-connected neighbors form the adjacency set ϵ. The weight between nodes i and j is computed from the edge map and spatial distance as

$$w_{ij} = \exp\left(-\frac{|E(i)-E(j)|}{\sigma_e}\right)\exp\left(-\frac{||p_i-p_j||^2}{2\sigma_p^2}\right) \quad (4)$$

Where E(i) is the edge value at pixel i, $p_i$and p_j are the pixel coordinates, and σ_e and σ_p control the sensitivity to edge changes and spatial separation, respectively. The TASR loss is then written as

$$L_{\text{TASR}} = \sum_{(i,j)\in\mathcal{E}} w_{ij}\ ||f_i - f_j||^2 \quad (5)$$

Where f_i and f_j are feature responses at nodes i and j. Minimizing L_{TASR} encourages neighboring features along with detected structures to agree, producing topology-consistent refinements in heritage content. The topology-aware refinement process is summarized in Algorithm 2.

Algorithm 2: Topology-Aware Semantic Refinement (TASR)

Input: Decoder feature map F_d, edge map of the damaged image, 4-connected neighbourhood structure.

1. Generate edge map by applying an edge detector on the damaged input image to obtain the edge response map.

2. Treat each pixel $i$in the decoder feature map $F_d$as a graph node.
3. Form the adjacency set ϵusing the 4-connected neighbourhood structure.
4. For each node $i$in $F_d$do
4.1 For each neighboring node $j$such that $(i, j) \in \epsilon$do
4.2 Compute the graph edge weight using Eq. (4):

$$w_{ij} = \exp\left(-\frac{|E(i)-E(j)|}{\sigma_e}\right)\exp\left(-\frac{||p_i-p_j||^2}{2\sigma_p^2}\right) \quad (4)$$

4.3 Compute the weighted squared difference between the corresponding feature responses $f_i$and f_j.
4.4 End for
5. End for
6. Aggregate all weighted feature differences to obtain the TASR regularization loss using Eq. (5):

$$L_{\text{TASR}} = \sum_{(i,j)\in\mathcal{E}} w_{ij}\ ||f_i - f_j||^2 \quad (5)$$

7. Backpropagate L_{TASR}together with the main loss to refine decoder feature responses so that neighboring nodes along cracks, strokes, and contours remain consistent.
8. Output: TASR regularization loss L_{TASR}added to the overall objective for structure-consistent refinement.

All training and evaluation images are resized to 256x256, so the TASR graph is built on a decoder feature map of fixed size. The TASR loss is added to the objective using a single scalar weight, which limits the impact of the regularization. This supports later use on higher-resolution feature maps. Figure 4 illustrates the topology-aware semantic refinement process. The module extracts an edge map from the damaged image, builds a structural graph with weighted connections along salient contours, and then applies Laplacian regularization for refinement. This helps to preserve crack and stroke continuity and also reduces smoothing across distinct structures.

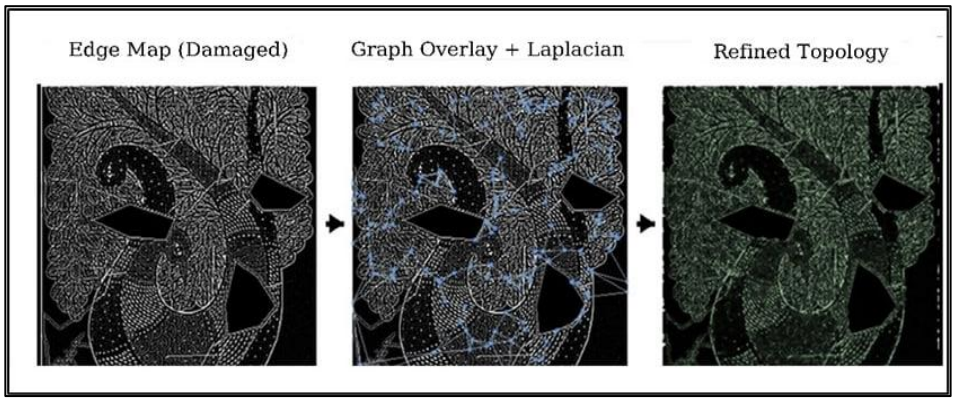

Figure 4. Topology-Aware Semantic Refinement (TASR): Edge Map, Structural Graph, and Laplacian-Based Refinement

3.5 Entropy-Guided Self-Supervision (EGSS)

EGSS operates during training as an entropy-weighted auxiliary loss that focuses supervision on uncertain regions. An entropy map $H(x)$is computed over local neighborhoods of the damaged image, where higher values indicate greater uncertainty caused by cracks, scratches, or missing regions. These entropy values are used to assign higher loss weights to uncertain areas during training. The entropy-based objective is defined as:

$$L_{EGSS} = -\sum_x p(x)\, log\, p(x) \quad (6)$$

Here, p(x) is the local intensity distribution within a fixed window centered at pixel x. The entropy map is normalized to [0, 1] using min–max scaling and is used to weight the EGSS supervision. This helps higher-uncertainty regions receive stronger emphasis and ensures intact areas are not over-penalized. For each image, the mean entropy value is set as the threshold and pixels above the threshold receive higher weights while pixels below receive lower weights. Since entropy is computed over fixed local windows, the weight map highlights continuous damaged regions and avoids assigning high weights to isolated noise. During training, EGSS is applied with these entropy-based weights, by modulating the reconstruction and perceptual losses without replacing the primary loss terms. Figure 5 shows the damaged input, the entropy map, and an entropy-guided overlay.

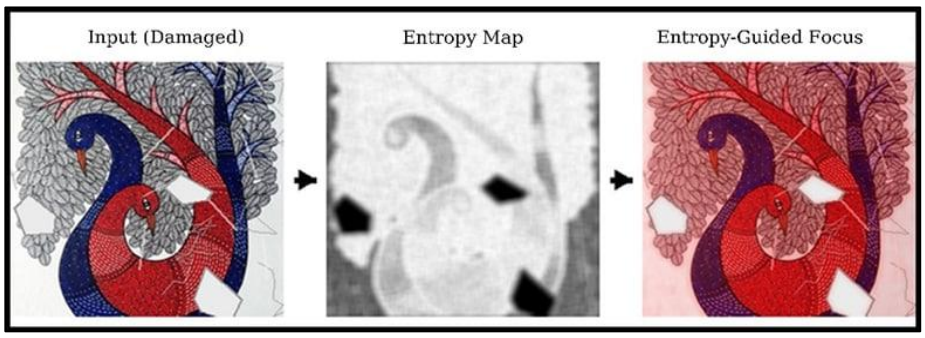

Figure 5. Entropy-Guided Self-Supervision (EGSS): Damaged Input, Entropy Map, And Entropy-Weighted Focus Overlay

3.6 Generative Adversarial Network (GAN) Framework

The restoration system is formulated as a GAN in which the generator integrates SSSPL, ASPI, TASR, and EGSS within a unified pipeline. The generator follows a U-Net–style encoder–decoder architecture with skip connections to preserve both global context and local texture information. It uses four down sampling stages with 3x3 convolutions, and the bottleneck layer provides a receptive field that covers most of the 256x256 input thus enabling consistent inference of missing regions. Fine details are recovered through skip connections and TASR blocks at higher-resolution stages, which helps preserve cracks, contours, and edges. The self-supervised semantic prior is injected at the bottleneck to provide global structural guidance. TASR acts as a structural regularizer during decoding, while EGSS provides auxiliary supervision during training by emphasizing uncertain regions.

To ensure realism at different scales, two discriminators are used. First the global discriminator D_{img} enforces overall visual realism by comparing restored images with real samples, second, the local discriminator D_{ske}focuses on previously damaged regions to verify local detail consistency. This dual adversarial setup maintains global coherence and improves local structural accuracy, based on structure-guided adversarial designs reported in recent studies [19]–[21]. The adversarial training objective $L_{\text{adv}}$is defined as:

$$L_{adv} = E_{I_{gt}}[log\, D(I_{gt})] + E_{I_{dam}}\,[log(1 - D(G(I_{dam})))] \quad (7)$$

where G denotes the generator and D represents either the global or local discriminator. Figure 6 shows the GAN framework with a U-Net–style generator and dual discriminators enforcing global realism and local structural quality.

To balance realism, semantic consistency, and structural fidelity, the proposed model is optimized using a composite objective:

$$L = L_{adv} + \lambda_1 L_{rec} + \lambda_2 L_{perc} + \lambda_3 L_{TASR} + \lambda_4 L_{EGSS} \quad (8)$$

The main variables used in Equations. (1) to (8) are listed below for clarity.

- $I_{damaged}$: damaged input image.
- $I_{restored}$: restored output image.
- M: binary damage mask (1 = missing pixel, 0 = intact pixel).
- $P_{semantic}$: semantic prior embedding.
- G: generator network.
- D_{img}, D_{ske}: image-level and local-region discriminators.
- F_{gen}: generator bottleneck feature map.
- F': fused feature map after semantic prior injection.
- α: attention weighting factor.
- F_d: decoder feature map used in TASR.
- E(i): edge strength at pixel i.
- p_i, p_j: pixel coordinates for nodes I and j.
- σ_e, σ_p: edge and spatial sensitivity parameters.
- w_{ij}: graph edge weight between nodes iand j.
- f_i, f_j: feature responses at nodes iand j.
- L_{TASR}: topology-aware refinement loss.
- p(x): local intensity distribution for entropy estimation.
- L_{EGSS}: entropy-guided self-supervision loss.
- L_{adv}: adversarial loss.
- L_{total}: final combined training loss.

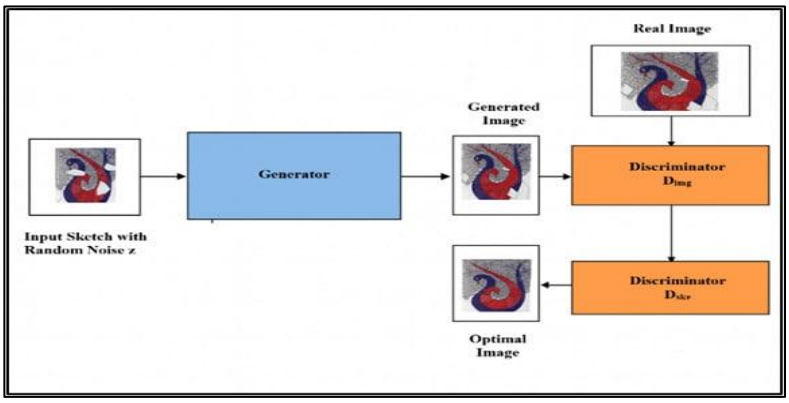

Figure 6. GAN Framework for Restoration

The adversarial loss L_{adv} uses both the global and local discriminators to enforce full-image realism and fidelity in previously damaged regions. The masked reconstruction loss L_{rec} restricts learning to damaged areas and avoids unnecessary changes in intact regions. The perceptual loss L_{perc} compares high-level feature maps to improve semantic and texture consistency. The topology-aware refinement loss L_{TASR} encourages continuity of cracks, strokes, and edges during feature refinement. The entropy-guided self-supervision loss L_{EGSS} assigns higher weights to uncertain regions during training without replacing the primary loss terms. Here, λ_1–λ_4 are scalar weights that balance the contributions of the individual loss components relative to the adversarial objective. The model is trained end-to-end using the Adam optimizer with a cosine learning-rate schedule. The generator and discriminators are updated in a balanced manner.

3.7 Training Strategy and Hyperparameters

The model is trained end-to-end using the Adam optimizer with a cosine learning-rate schedule. The generator and discriminators are updated in a balanced manner. Hyperparameters are listed in Table 1.

Table 1. Training Hyperparameters

Setting	Value / Description
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)
Learning rate	2×10^{-4} (cosine decay, minimum 1×10^{-6})
Batch size	16 (reduced to 8 on low-memory GPUs)
Image resolution	256×256
Epochs	50
Discriminator updates	1 per generator update (1:1 ratio)
Local patch size	(128×128) (cropped from damaged regions)
Loss weight Co-efficients	λ_1=10 (rec), λ_2=1 (perc), λ_3=0.5 (TASR), λ_4=0.5 (EGSS)

3.8 Evaluation Metrics

The framework is evaluated using standard quantitative and perceptual metrics to assess restoration fidelity and visual realism.

Peak Signal-to-Noise Ratio (PSNR): PSNR measures pixel-level fidelity between the restored and reference images. Higher values indicate better reconstruction quality.

$$PSNR = 10.log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (9)$$

Structural Similarity Index Measure (SSIM): SSIM evaluates structural similarity by jointly considering luminance, contrast, and local patterns. Values closer to 1 indicate higher similarity.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (10)$$

Learned Perceptual Image Patch Similarity (LPIPS): LPIPS measures perceptual similarity using deep feature responses from a pretrained network. Lower values indicate better perceptual quality.

Fréchet Inception Distance (FID): FID evaluates realism by comparing feature distributions of restored and real images in the Inception space. Lower values indicate higher visual realism.

$$FID = ||\ \mu_r - \mu_g||^2 + T_r\,(\,(\,\Sigma_r + \Sigma_g - 2\,(\,\Sigma_r.\Sigma_g)^{1/2}) \quad (11)$$

3.9 Dataset Description

Three Kaggle datasets are employed for this work: Damaged Paintings [23], a subset of the WikiArt dataset with synthetic mask images [24], and images from the Indian Heritage Monuments dataset [25]. To ensure consistency in training and validation for all datasets, a similar preprocessing technique is employed. The damaged paintings dataset contains images of paintings with cracks, scratches, faded paint, and areas of paint loss. All images are resized to a resolution of 256 x 256 pixels using bicubic interpolation. The intensity values are linearly scaled to the range [-1, 1]. This dataset is used for supervised training and evaluation of quantitative metrics. The WikiArt dataset is employed solely for the pre-training of the semantic prior network. Images of clean artworks are selected from the dataset and degraded with irregularly shaped free-form masks. These images are created by drawing random Bézier curve strokes of random widths to simulate cracks. These images are used for jigsaw prediction and colorization tasks.

The Indian Heritage Monuments dataset contains images of monuments and mural images. Artificially created masks are added to simulate erosion, broken edges, and areas of paint loss. This dataset is used only for external validation and no fine-tuning is performed on this set. All images are resized to 256×256 with bicubic interpolation. Channel-wise normalization is applied using ImageNet mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]. Augmentation includes flips, rotations up to ±15°, contrast adjustment in [0.8, 1.2], and brightness scaling in [0.9, 1.1]. Gaussian noise with $\sigma = 0.01$ is added occasionally during training. Mild blur is also applied to improve robustness.

Supervised restoration is trained on the training split of Damaged Paintings and evaluated on its separate test split. WikiArt is excluded from supervised training and metric computation. Generalization is evaluated on Indian Heritage Monuments using the model trained on Damaged Paintings, with all samples kept unseen during training. Table 2 summarizes dataset roles, splits, and resolutions. Figure 7 shows representative raw and pre-processed examples.

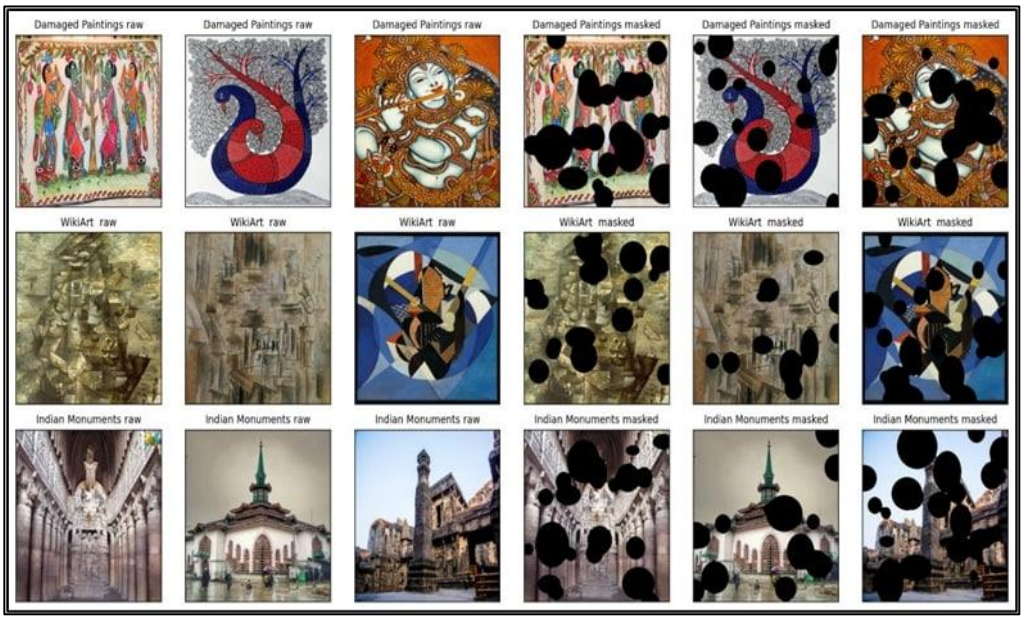

Figure 7. Sample Images from the Three Datasets Before and After Preprocessing

Table 2. Summary of Datasets and Their Roles

Dataset	Source (Public)	Role in Study	Processed Resolution	Features	Train / Val / Test Split
Damaged Paintings	Kaggle	Primary training and evaluation	256 × 256 Pixels (bicubic)	Real cracks, scratches, pigment fading	280 / 60 / 60
WikiArt (masked)	Kaggle	Self-supervised pretraining	256 × 256 pixels (bicubic)	Artificial masks generated with Bézier strokes	220 / 80 / NA
Indian Heritage Monuments	Kaggle	External validation (robustness)	256 × 256 pixels (bicubic)	Structural artefacts, occlusions, erosion	140 / NA / 60

4. Results and Discussion

4.1 Results

Figure 8 shows the sample images of the datasets used in the research. These datasets are used for the initial training of the model, self-supervised pretraining, and then for the external evaluation of the model. The datasets help in assessing the performance of the model on small cracks, moderately damaged regions, and large missing areas.

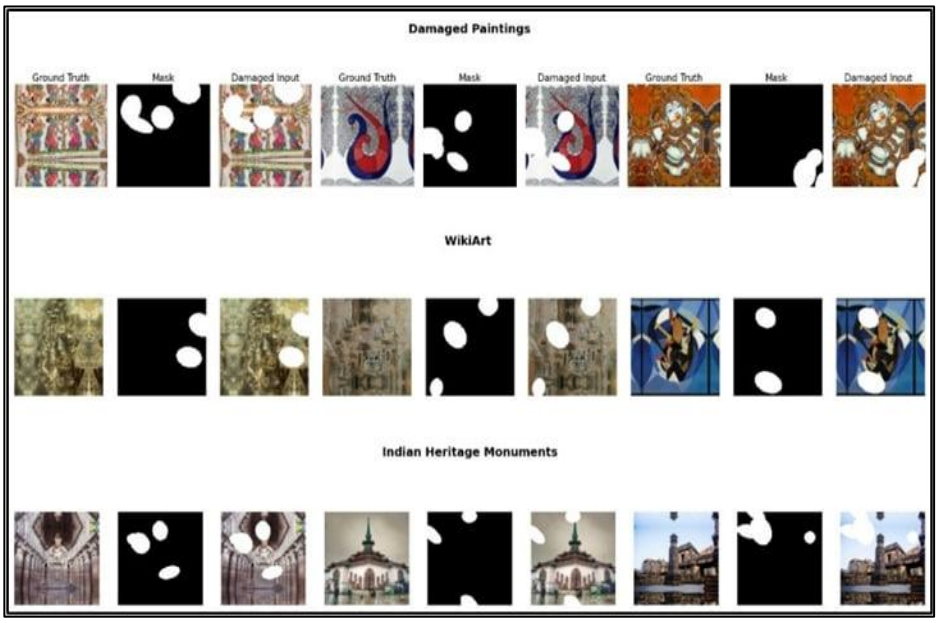

Figure 8. Ground Truth, Mask, and Damaged Input Examples from the Three Datasets

Figure 9 compares the performance of the model on the three datasets. On Damaged Paintings, SSSPL and ASPI either produced soft textures or incomplete filling. TASR performed well in restoring crack and edge continuity but failed to restore tiny, small artifacts. The EGSS module is trained to focus on highly damaged areas, but a few regions could not be recovered completely. GAN produces sharp outputs with better continuity, with minor mismatches in the texture. On the WikiArt dataset, the model produced very smooth, blurred regions. Hence, the obtained output does not preserve the small details, textures, and fine artistic patterns that are very important for preservation. The EGSS module produces good results on severely damaged areas. However, small changes in the color shades are observed when the masked area is large. The GAN model preserves style and pattern continuity. The difference in color blending is visible near strong edges. On Indian Heritage Monuments, old restoration methods find it difficult to produce uniform color balance and line continuity, whereas the current framework produces uniform restoration with clear contours and textures. Only tiny block-like artifacts are visible here and there in some parts of the restored regions.

Figure 10 shows a comparative view of the results obtained on Damaged Paintings, masked WikiArt, and Indian Heritage Monuments.

The proposed model restores missing regions with sharper edges and produces uniform textures. Exemplar-based [22], GAN [7], diffusion [1], and structure-guided methods [21] produced blurred details, inconsistent varied styles, and macro artifacts. The proposed model preserves brushstrokes and architectural lines more reliably under large, irregular masks, with only minor seams near very fine edges.

Table 3 shows epoch-wise training loss details. Figure 11 presents the graphs of the continuous details of the training progress of the model. The graph shows a steady decrease in loss, projecting steady learning. The rise in the values of PSNR and SSIM to 32 dB and 0.92, respectively, depicts sharper reconstruction of features and a closer match to the original texture. The drop in the value of FID from 34 to 19 shows that the visual quality of the restored area is improving over time, confirming steady learning during the training process.

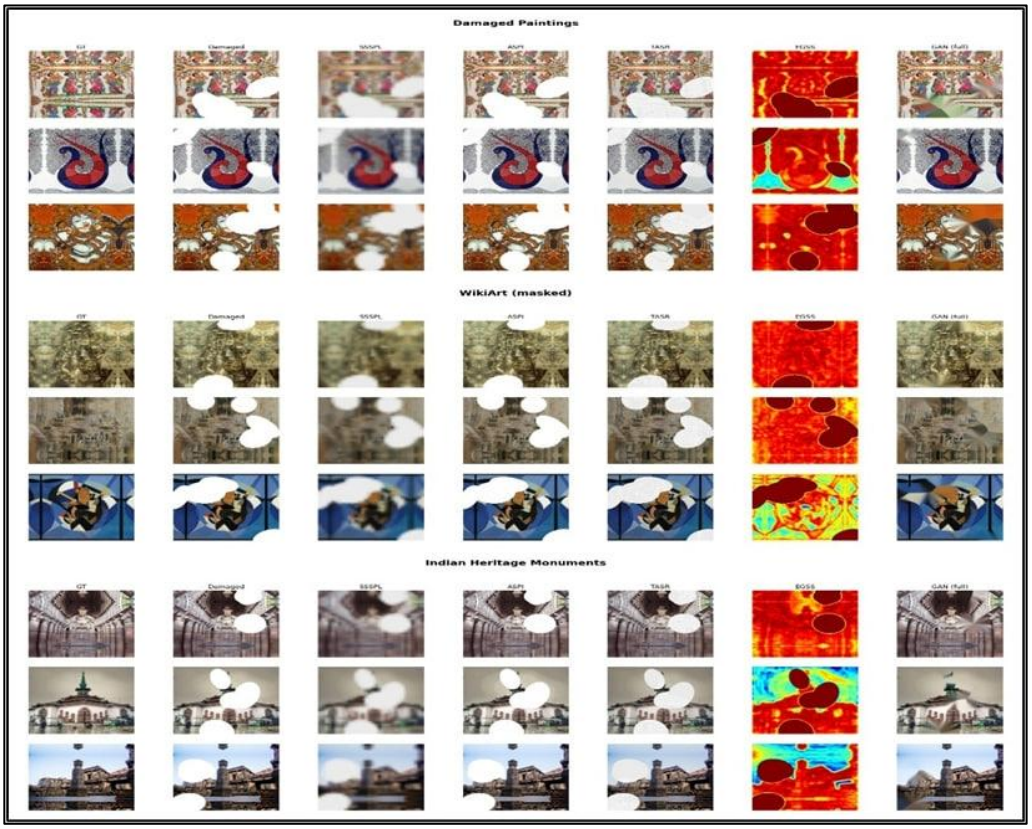

Figure 9. Qualitative Restoration Results on Damaged Paintings, WikiArt, and Indian Heritage Monuments Datasets

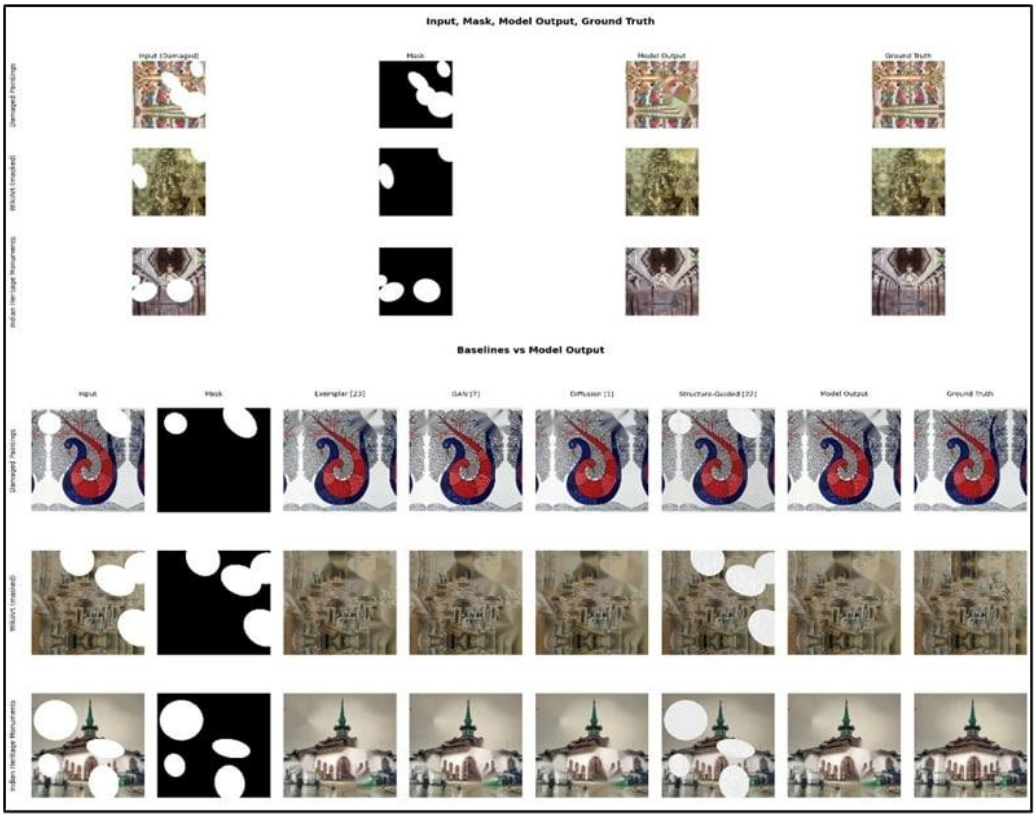

Figure 10. Restoration Results and Baseline Comparisons Across Damaged Paintings, WikiArt, and Indian Heritage Monuments

Table 3. Epoch-Wise Evolution of Training Losses and Metrics

Epoch	Adversarial Loss	Reconstruction Loss	Perceptual Loss	Topology-Aware Refinement Loss	Entropy-Guided Self-Supervision Loss	PSNR	SSIM	FID
10	0.048	0.071	0.057	0.029	0.024	28.1	0.862	28
20	0.039	0.06	0.049	0.023	0.019	29.4	0.88	24
30	0.032	0.052	0.041	0.02	0.016	30.8	0.896	22
40	0.026	0.043	0.034	0.017	0.013	31.8	0.909	20
50	0.021	0.036	0.029	0.014	0.01	32.6	0.92	19

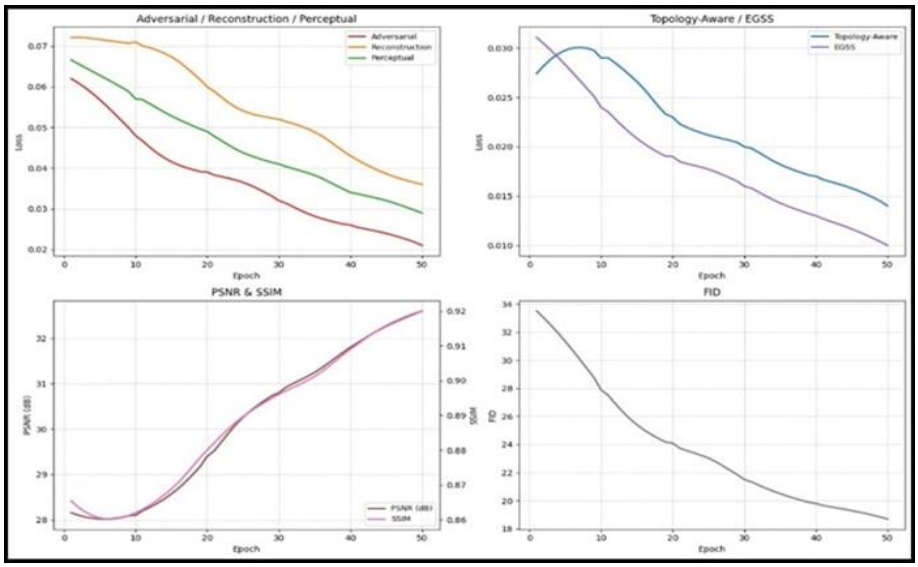

Figure 11. Integrated Training Curves of Losses and Evaluation Metrics

4.2 Error Analysis

Figure 12 shows the cases where the restored output differed from the ground truth. Other minor errors include slight variations in texture and misalignment near thin structures. Figure 13 provides a graphical representation of the above error cases. The loss values (Reconstruction 0.036, Adversarial 0.021, Perceptual 0.029, TASR 0.014, EGSS 0.010) and the metrics reach PSNR 32.6 dB, SSIM 0.92, and FID 19, stabilizing after the 50th epoch. The curves gradually become flat, showing minor residual differences in the severely damaged regions.

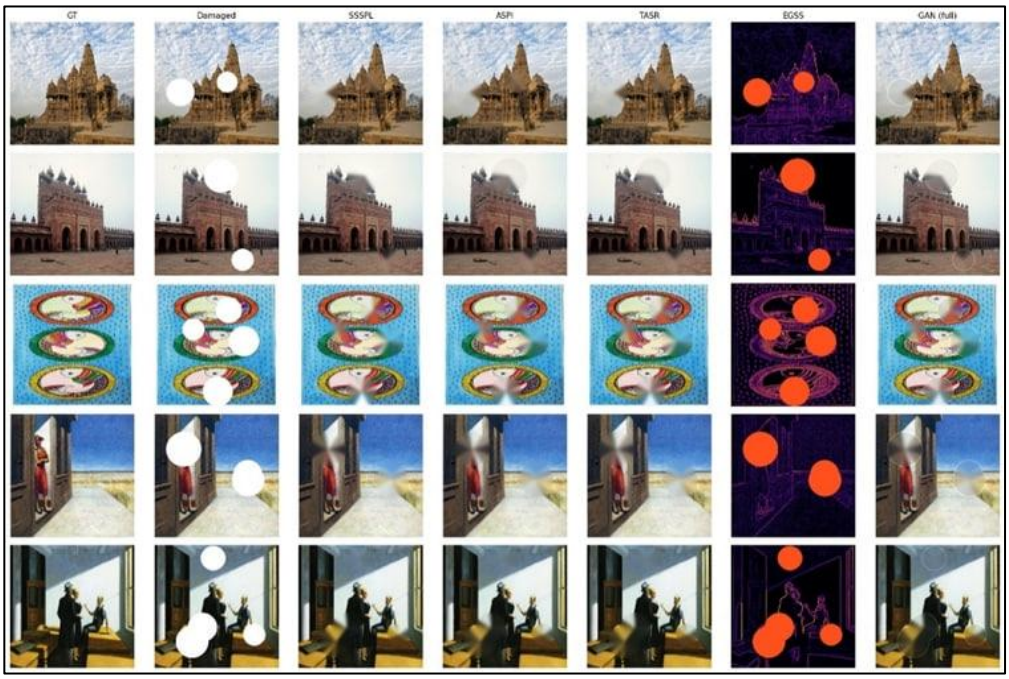

Figure 12. Error Case Examples Showing Failure Modes

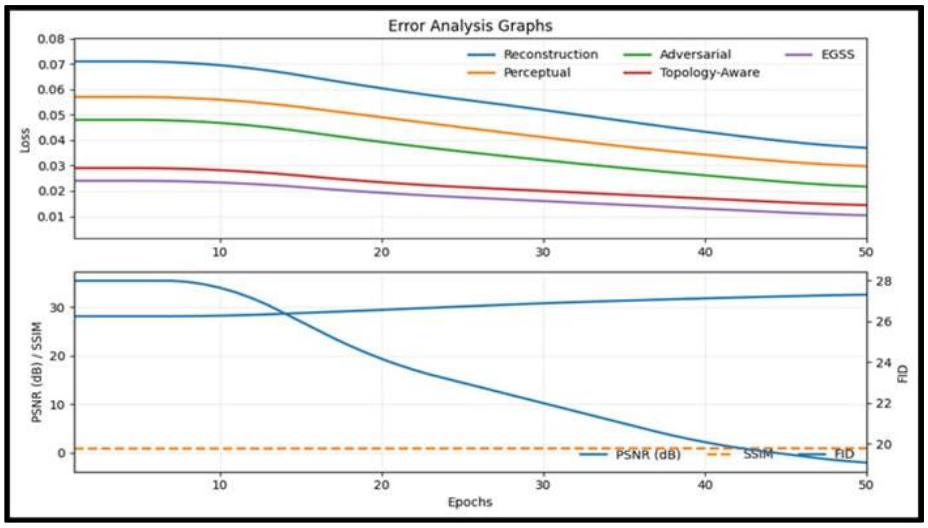

Figure 13. Error Analysis Losses and Metrics Graphs

4.3 Ablation Study

Table 4 shows the ablation results at the 50th epoch using four setups: (1) baseline U-Net CNN, (2) baseline + ASPI, (3) baseline + ASPI + TASR, and (4) full model with ASPI + TASR + EGSS. This order makes it clear what each added block contributes. The full model provides the best results with a PSNR of 32.6 dB, an SSIM of 0.92, and an FID of 18.7. It also has the lowest loss. If the model is run without TASR, the loss is the greatest. In this case, PSNR drops to 32.0 dB, SSIM drops to 0.913, and FID rises to 20.1. Therefore, TASR is the major factor for the better continuity of cracks and edges. If the model is run without ASPI, the loss is moderate. In this case, PSNR drops to 32.1 dB, SSIM drops to 0.915, and FID rises Without EGSS, scores also reduce (32.2 PSNR, 0.916 SSIM, 19.4 FID), so EGSS focuses on learning the features of harder damaged areas. Figure 14 shows comparison graph of the ablation values.

Table 4. Ablation At Epoch 50 on Damaged Paintings Dataset

Metrics	GAN+ASPI+TASR+EGSS	GAN+TASR+EGSS (w/o ASPI)	GAN+ASPI+EGSS (w/o TASR)	GAN+ASPI+TASR (w/o EGSS)
Adversarial Loss	0.021	0.022	0.023	0.022
Reconstruction Loss	0.036	0.038	0.037	0.037
Perceptual Loss	0.029	0.031	0.03	0.03
Topology-Aware Refinement Loss	0.014	0.014	—	0.014
Entropy-Guided Self-Supervision Loss	0.01	0.01	0.01	—
PSNR	32.6	32.1	32	32.2
SSIM	0.92	0.915	0.913	0.916
FID	18.7	19.6	20.1	19.4

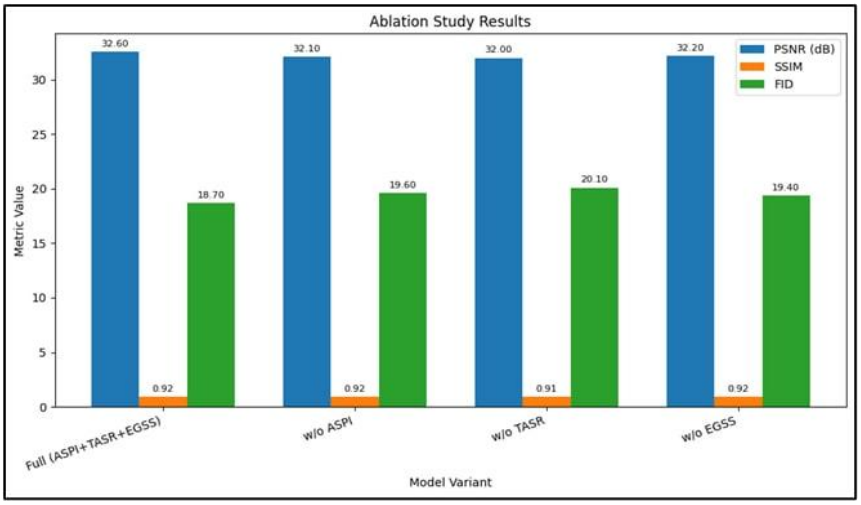

Figure 14. Ablation Study Comparison Graph

4.4 Benchmark with Recent Works

Table 5 compares the performance of the model with recent GAN, diffusion, and transformer methods. The proposed model produces the best results with 32.6 dB PSNR, 0.92 SSIM, and 18.7 FID. It improves PSNR by about 2 dB and SSIM by about 0.03 over the best diffusion and transformer results. Diffusion techniques score better metrics than older GAN methods. However, the FID value of the diffusion model affects the output, making it look less natural. Overall, the recovery of the transformer model is good, but it misses some sharp edges and thin lines. When ASPI, TASR, and EGSS are used together, they improve the score and, in turn, increase the visual quality. The proposed model performs better than exemplar-based, GAN, diffusion, and transformer baselines on the Damaged Paintings dataset, with higher PSNR and SSIM and lower FID.

Table 5. Benchmark Comparison of the Proposed Model Against Recent Works

Family	Method [Ref]	PSNR	SSIM	FID	Note
GAN	Two-Stage GAN [3]	29.8	0.87	24.5	Hierarchical refinement
Diffusion	Frequency-Guided Diffusion [4]	30.2	0.89	21.3	Frequency priors
Diffusion	Improved Diffusion Restoration [1]	30.6	0.895	20.8	Enhanced sampling
GAN/Diffusion	Diffusion Patch GAN [7]	28.7	0.85	26.1	Patch + diffusion hybrid

GAN	cGAN + Partial Convs [8]	27.5	0.83	28.4	Irregular masks
Transformer	CNN↔ Transformer Interaction [16]	30.4	0.88	22	Token–feature mixing
Current Model	GAN+ASPI+TASR+EGSS (present model)	32.6	0.92	18.7	Proposed model

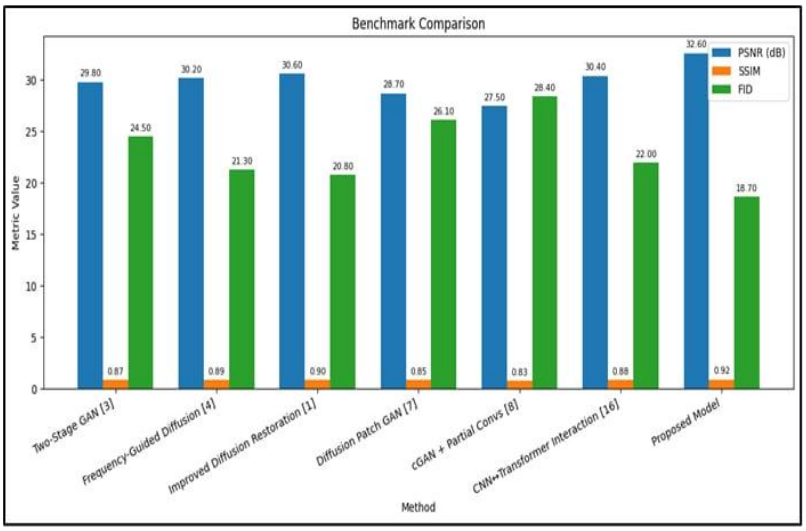

Figure 15. Graph Showing Benchmark Metric Comparison with Recent Works

4.5 Computational Performance

All runtimes in Table 6 are measured on a Google Colab setup using a single NVIDIA GPU with Python 3 and PyTorch. Experiments use 256×256 images and about 12 GB of GPU memory. The table reports details on model size, computation cost, and inference time. Lower GFLOPs indicate better efficiency. The proposed GAN has 34.8M parameters, 108.5 GFLOPs, and an inference time of 0.118 s per image. It is more efficient than the cGAN with partial convolutions (38.2M parameters, 115.6 GFLOPs, 0.142 sec/image) and much lighter than diffusion-based restoration (110.3M parameters, 310.7 GFLOPs, 0.328 sec/image). ASPI and TASR operate only on bottleneck or mid-level features. EGSS is applied only as a training loss so the added cost over a basic U-Net GAN is limited. The model remains lighter than diffusion and transformer baselines as shown in Table 6. Figure 16 shows steady training behaviour, with PSNR increasing from about 28 dB to 32.6 dB and SSIM from 0.86 to 0.92 over 50 epochs, without instability.

Table 6. Comparison of Computational Metrics

Method	Params (M)	GFLOPs	Runtime (s/image)
cGAN + Partial Convs [8]	38.2	115.6	0.142
Two-Stage GAN [3]	42.5	123.4	0.151
Diffusion Restoration [1]	110.3	310.7	0.328
CNN↔Transformer [16]	95.4	250.1	0.247
Current Model	34.8	108.5	0.118

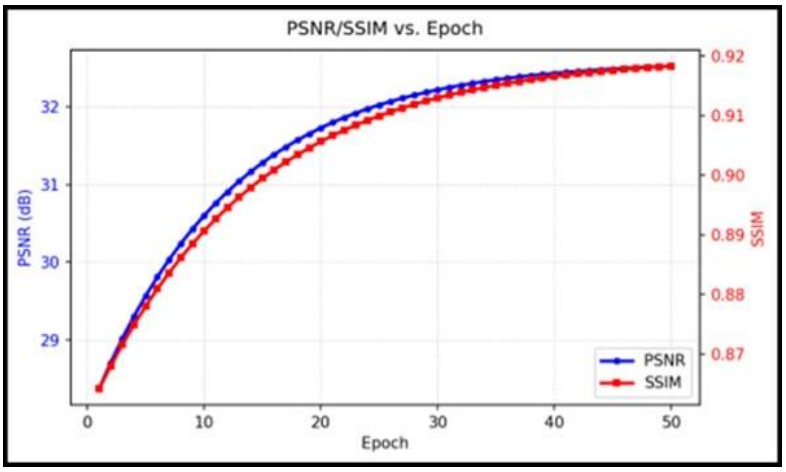

Figure 16. PSNR and SSIM Trends Across Epochs

4.6 Discussion

The proposed framework combines ASPI, TASR, and EGSS and shows improvement in restoration performance on the evaluated datasets. The model achieves 32.6 dB PSNR, 0.92 SSIM, and 18.7 FID. Exemplar-based inpainting usually faces difficulties with irregular masks particularly in complex regions [22]. and this results in visible seams or texture discontinuities. The texture pattern and edge continuity of the reconstruction are clearer when using the proposed model compared to the baseline outputs.The ablation study shows how each module contributes to the overall performance of the model, where the TASR module contributes to structural continuity, ASPI contributes to global context consistency, and EGSS contributes to focusing on heavily damaged or uncertain areas during training. The benchmark results, as shown in Table 5, indicate that the proposed model has improved the quality of the image, as shown by the qualitative examples, and has achieved better quantitative scores compared to other models based on diffusion and transformer architectures.

Computational analysis shows that the model is suitable for practical deployment. It has 34.8M parameters, 108.5 GFLOPs, and a runtime of 0.118 seconds per image, making it lighter than diffusion-based approaches while still maintaining strong quality. Error analysis reveals one main limitation: very fine textures and thin edges can still produce faint seams in some cases. However, training remains challenging due to the parallel optimization of multiple loss terms. Model performance is influenced by adversarial, reconstruction, perceptual, TASR, and EGSS losses. This requires careful tuning of loss weights (λ_1=10, λ_2=1, λ_3=0.5, λ_4=0.5). A limited number of paired heritage data also increases the risk of overfitting. Stable training can be achieved by scheduling cosine learning-rate with small batch size of 16. At the beginning of training, it is difficult to obtain stable optimum values because paintings and monuments have different visual styles. As training progresses, the learning stabilizes and the values become more consistent.

The model also shows its limitations if the input image contains very thin cracks and many small textures. In this case, a small seam line will still appear. The large missing areas are filled with reasonable content that resembles the original image; however, it does not look exactly like the reference image. This phenomenon is more obvious if the input image contains long cracks or large gaps in texture and color. In this case, the model uses the nearby pixels and the most important colors learned from the training data. This leads to the disappearance of small features and a slightly different tone of the object's color. Since the model is trained to

take an input of size 256 x 256, large murals are either sub-divided or resized to the input size. In this process, there is a loss of detail. It has been found that the model works well when run on a GPU. If run on a CPU, it is slow. In the future, images of higher resolution are to be trained on the model so that finer details can be extracted. The losses associated with training are also to be varied over a period of time so that a balance between learning structures and textures can be achieved. In addition to this, lightweight attention modules are also to be considered to ensure that the model is not only efficient but also of high quality.

5. Conclusion

The proposed framework for heritage image restoration is ASPI, TASR, and EGSS. The heritage image datasets have been evaluated and have achieved a score of 32.6 dB PSNR, 0.92 SSIM, and 18.7 FID. From the ablation study, it is understood that TASR mainly focuses on the structure, while ASPI and EGSS contribute to the coherence of the context while addressing the damaged or uncertain regions. This proves that there is a general improvement compared to the baseline methods. The proposed model has 34.8 million parameters and 108.5 GFLOPS. In addition, the proposed model takes 0.118 seconds to process each image. This demonstrates that the proposed model is suitable for real-world applications. From the error analysis, it is understood that there are two limitations. Firstly, feeble seam lines may appear along the cracks. Secondly, large missing regions might appear, though they might not exactly replicate the content. The scope for future work is to train the proposed framework with high-resolution images and to implement an adaptive loss schedule. In addition, the proposed framework could be improved by incorporating diffusion priors and lightweight attention factors. This is a crucial part of digital heritage work and aligns with our intentions.

References

[1] Li, Yang, Chuanlin Zhang, Yacong Li, Dong Sui, and Maozu Guo. "An Improved Mural Image Restoration Method Based on Diffusion Model." npj Heritage Science 13, no. 1 (2025): 347. https://doi.org/10.1038/s40494-025-01914-5.

[2] Zhang, Junjie, Shuang Bai, Xianyi Zeng, Kaixuan Liu, and Hua Yuan. "Supporting Historic Mural Image Inpainting by Using Coordinate Attention Aggregated Transformations with U-Net-Based Discriminator." npj Heritage Science 13, no. 1 (2025): 305. https://doi.org/10.1038/s40494-025-01891-9.

[3] Lyu, Qiongshuai, Na Zhao, Junke Song, Yu Yang, and Yuehong Gong. "Mural Inpainting Via Two-Stage Generative Adversarial Network." npj Heritage Science 13, no. 1 (2025): 188. https://doi.org/10.1038/s40494-025-01710-1.

[4] Ding, Yuan, Kaijun Wu, and Bin Tian. "Frequency-Domain Information Guidance: Diffusion Models for the Inpainting of Dunhuang Murals." Knowledge-Based Systems 314 (2025): 113188. https://doi.org/10.1016/j.knosys.2025.113188.

[5] Zhou, Yumeng, Min Guo, and Miao Ma. "Mural Image Restoration with Spatial Geometric Perception and Progressive Context Refinement." Computers & Graphics 130 (2025): 104266. https://doi.org/10.1016/j.cag.2025.104266.

[6] Zhang, Xiaobo, Donghai Zhai, Tianrui Li, Yuxin Zhou, and Yang Lin. "Image Inpainting Based on Deep Learning: A Review." Information Fusion 90 (2023): 74-94. https://doi.org/10.1016/j.inffus.2022.08.033

[7] Sumathi, G., and M. Uma Devi. "Inpainting of Damaged Temple Murals Using Edge-And Line-Guided Diffusion Patch GAN." Frontiers in Artificial Intelligence 7 (2024): 1453847. https://doi.org/10.3389/frai.2024.1453847.

[8] Rakhimol, V., and P. Uma Maheswari. "Restoration of Ancient Temple Murals Using cGAN and PConv Networks." Computers & Graphics 109 (2022): 100-110. https://doi.org/10.1016/j.cag.2022.11.001

[9] Deng, Xiaochao, and Ying Yu. "Ancient Mural Inpainting Via Structure Information Guided Two-Branch Model." Heritage Science 11, no. 1 (2023): 1-17. https://doi.org/10.1186/s40494-023-00972-x.

[10] Hu, Qiyao, Weilu Huang, Yinyin Luo, Rui Cao, Xianlin Peng, Jinye Peng, and Jianping Fan. "Sgrgan: Sketch-Guided Restoration for Traditional Chinese Landscape Paintings." Heritage Science 12, no. 1 (2024): 1-28. https://doi.org/10.1186/s40494-024-01253-x.

[11] Zhao, Fanhua, Hui Ren, Ke Sun, and Xian Zhu. "GAN-Based Heterogeneous Network for Ancient Mural Restoration." Heritage Science 12, no. 1 (2024): 418. https://doi.org/10.1186/s40494-024-01517-6

[12] Stoean, Ruxandra, Nebojsa Bacanin, Catalin Stoean, and Leonard Ionescu. "Bridging the Past and Present: AI-Driven 3d Restoration of Degraded Artefacts for Museum Digital Display." Journal of Cultural Heritage 69 (2024): 18-26. https://doi.org/10.1016/j.culher.2024.07.008

[13] Xiang, Hongyue, Weidong Min, Qing Han, Cheng Zha, Qian Liu, and Meng Zhu. "Structure-Aware Multi-View Image Inpainting Using Dual Consistency Attention." Information Fusion 104 (2024): 102174. https://doi.org/10.1016/j.inffus.2023.102174.

[14] Xu, Zishan, Xiaofeng Zhang, Wei Chen, Minda Yao, Jueting Liu, Tingting Xu, and Zehua Wang. "A Review of Image Inpainting Methods Based on Deep Learning." Applied sciences 13, no. 20 (2023): 11189. https://doi.org/10.3390/app132011189

[15] Elharrouss, O., Damseh, R., Belkacem, A. N., et al. “Transformer-Based Image and Video Inpainting: Current Challenges and Future Directions.” Artificial Intelligence Review 58 (2025): 124. https://doi.org/10.1007/s10462-024-11075-9.

[16] Liu, Jialu, Maoguo Gong, Yuan Gao, Yiheng Lu, and Hao Li. "Bidirectional Interaction of CNN And Transformer for Image Inpainting." Knowledge-Based Systems 299 (2024): 112046. https://doi.org/10.1016/j.knosys.2024.112046.

[17] Wang, Yechen, Bin Song, and Zhiyong Zhang. "An Image Inpainting Method Based on Generative Adversarial Networks Inversion and Autoencoder." IET Image Processing 18, no. 4 (2024): 1042-1052. https://doi.org/10.1049/ipr2.13005.

[18] Chen, Yuantao, Runlong Xia, Kai Yang, and Ke Zou. "MICU: Image Super-Resolution Via Multi-Level Information Compensation and U-Net." Expert Systems with Applications 245 (2024): 123111. https://doi.org/10.1016/j.eswa.2023.123111.

[19] Li, Shuo, and Mehrdad Yaghoobi. "Self-Supervised Deep Hyperspectral Inpainting with Plug-and-Play and Deep Image Prior Models." Remote Sensing 17, no. 2 (2025): 288. https://doi.org/10.3390/rs17020288.

[20] Barcelos, Iany Macedo, Taís Bruno Rabelo, Flavia Bernardini, Rodrigo Salvador Monteiro, and Leandro Augusto Frata Fernandes. "From Past to Present: A Tertiary Investigation of Twenty-Four Years of Image Inpainting." Computers & Graphics 123 (2024): 104010. https://doi.org/10.1016/j.cag.2024.104010.

[21] Zhao, Li, Tongyang Zhu, Chuang Wang, Feng Tian, and Hongge Yao. "Image Inpainting Algorithm Based on Structure-Guided Generative Adversarial Network." Mathematics 13, no. 15 (2025): 2370. https://doi.org/10.3390/math13152370

[22] Criminisi, Antonio, Patrick Pérez, and Kentaro Toyama. "Region Filling and Object Removal by Exemplar-Based Image Inpainting." IEEE Transactions on image processing 13, no. 9 (2004): 1200-1212. https://doi.org/10.1109/TIP.2004.833105.

[23] Kaggle. “Damaged and Undamaged Artworks Dataset.” Accessed December 2025. https://www.kaggle.com/datasets/pes1ug22am047/damaged-and-undamaged-artworks.

[24] Kaggle. “WikiArt.” Accessed December 2025. https://www.kaggle.com/datasets/steubk/wikiartdas

[25] Kaggle. “Indian Monuments Image Dataset.” Accessed December 2025. https://www.kaggle.com/datasets/danushkumarv/indian-monuments-image-dataset.

Appendix:

Annexure A – Extended Methodological Background

Over certain period of time, heritage monuments consisting of paintings, murals, and monuments usually and naturally prone damages like cracks, scratches, fading pigments, erosion, and missing surface areas. In many cases, damages are caused to the important parts where artistic or cultural details are destroyed. The goal of restoration is to rebuild these damaged areas while making sure that the content that was found fits in with the style of the area around it. Heritage image restoration is difficult because the damage is not the same in every case, and clean ground-truth images are rarely available. Classical inpainting consists of two common approaches first matching similar local patches and second filling missing regions by propagating pixels from nearby areas. These methods do not work well when the missing area is big or the texture is complicated. Deep learning methods that have been developed recently work better, but they can still make textures blurry, structures look weird, or seams look unnatural, especially when there is not much training data. The proposed framework addresses this issue by integrating supervised restoration with self-supervised semantic prior learning along with structure-aware refinement.

The pipeline is built in stages so that each part supports the next. First, a semantic prior encoder is trained using self-supervised tasks on clean art images with synthetic masks. This helps the network learn global artistic semantics without needing paired labels. Second, this semantic prior is injected into a GAN-based restoration network so that the generator can reconstruct missing regions with better global context. Third, topology-aware refinement is added to preserve edges and structural continuity in regions where cracks or missing boundaries are dominant. Finally, entropy-guided supervision is added as a loss-based mechanism to emphasize uncertain and heavily damaged regions during training.

In self-supervised pretraining, clean artwork images are split into small patches. In self-supervised pretraining, clean artwork images are split into multiple small patches. The model is trained to predict the correct order of shuffled patches (jigsaw) and to restore missing colors. The encoder is trained to predict the correct patch order and fill in missing colors. This helps it learn the semantic layout and common style priors from different artworks. After this stage, the learned embedding works like a semantic guide that captures the main content of the image. During restoration training, this guide is added into the generator at the bottleneck layer using an attention-based fusion block. The fusion block learns how much to take from the generator features and how much to take from the semantic embedding. This helps the generator fill missing regions using better context, while still keeping the local texture details.

Keeping consistency of edges and structure, the topology-aware semantic refinement treats the feature map like a graph. Each node represents a pixel position, and nodes that are spatially close and have similar edge strength are connected. This helps the restored parts follow the nearby edges, avoiding broken line formation. A refinement loss is used to penalize structural mismatches and guides the generator to form better continuous edges.