# Damage Analysis and Digital Restoration of Artworks

Luca Bonacorsi†, Letizia Carpi‡ and Riccardo Reale§
Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy
†41774@studenti.unimore.it, ‡269484@studenti.unimore.it, §272401@studenti.unimore.it

***Abstract*—Manual inspection of paintings is a fundamental process to safeguard the cultural heritage of art collection in museums and galleries, but can be a very challenging and time-consuming task even for the most experienced art restorers. In this paper, we present a multimodal approach that is capable of detecting visual deteriorations in paintings, analyzing / classifying the damages, and providing restoration strategies to speed up the manual work of art restorers and help/improve conservation efforts. Our system uses a combination of classical image processing and deep learning solutions to detect damaged regions from the photographic representations of paintings, and geometric correction methods to rectify perspective distortions commonly found in photographies. We design and train two custom convolutional neural networks to segment and classify damage patterns (a U-Net and a ResNet), and perform image inpainting using OpenCV functions. We demonstrate the practicality of our approach by showing how this proposal can provide efficient tools for the objective evaluation of artwork damage and restoration planning.**

## I. Introduction

Museums, galleries and cultural institutions preserve countless works of art, representing centuries of creative expression and historical memory. However, over time, these works of art become subject to physical deterioration in the form of cracks, fading, surface abrasions and other visual anomalies that threaten their aesthetic value and material stability. Traditionally, analysing this damage requires manual inspection by experts, which can be time-consuming, subjective and inconsistent between different specialists. The challenge of this project is to use innovative tools that combine knowledge of art history with technological precision. In this paper, we present an automated approach that uses computer vision techniques to assist with the analysis of damage and digital restoration of artworks. This pipeline could be valuable for art conservators and restorers, as well as for students and researchers in visual computing, digital humanities, and heritage preservation. Our approach uses classical image processing techniques such as edge detection and denoising filters to highlight damaged areas and minimise the impact of visual noise in artwork photography. These pre-processed images are then corrected geometrically using a four-point homography algorithm to correct distortions caused by the camera angle or the artwork's curvature. The third stage of the system involves semantic analysis of damaged regions, initially explored through two mutually exclusive strategies. On one hand, we generated masks using classical image filters to localize defects such as cracks, offering a lightweight approach. On the other hand, we trained two deep neural networks: a U-Net for semantic segmentation of damaged areas, and a ResNet for binary classification to distinguish cracks from non-crack regions. After evaluating both methods, we opted for the deep learning approach, considering its superior ability to identify and classify complex damage patterns with higher accuracy and scalability.

Finally, the restoration phase is performed using OpenCV-based inpainting techniques and a custom-trained deep learning model. This model, originally designed for inpainting on photographs, was fine-tuned on our painting dataset to adapt it specifically for artistic textures and styles. This enables a plausible visual repair of damaged areas while preserving artistic coherence. For evaluation, we used three datasets: DAMAGED_AND_UNDAMAGED_ARTWORKS, Dataset Card for "ARTeFACT" and a small custom dataset. These provide different examples of artistic deterioration and will be discussed in greater detail in the following chapter, where their structure, relevance, and limitations are analyzed. This three-source approach supports a more comprehensive validation of the proposed methods, offering solid foundations for future improvements in automated artwork conservation.

In Section II, we present the three datasets used in our pipeline. In Section III, we overview the system and each component of our multi-stage architecture. In Section IV, we describe the geometric rectification of the input images. In Section V, we discuss a rule-based approach for crack segmentation, and in Section VI we describe our deep learning solution for crack segmentation. In Section VII, we present the results of our experimental evaluation and discuss future improvements, and in Section VIII we conclude the paper.

## II. Datasets

The development and validation of our pipeline relies on three distinct datasets with complementary characteristics: DAMAGED_AND_UNDAMAGED_ARTWORKS, Dataset Card for "ARTeFACT" and a small custom dataset of photographs captured from personal devices. Each served a specific function within the broader scope of damage detection, segmentation, and geometric correction.

TABLE I: Summary of datasets used in the project

| Dataset | Source | Images |
|---|---|---|
| *DAMAGED_AND_UNDAMAGED_ARTWORKS* | Kaggle | 533 |
| *Dataset Card for "ARTeFACT"* | Hugging Face | 418 |
| Custom artwork dataset | Internal and online | 33 |

### A. DAMAGED_AND_UNDAMAGED_ARTWORKS

This dataset comprises 533 images in total, organised into two categories: paired and unpaired samples. In the paired subset, each damaged artwork is directly associated with its undamaged counterpart, enabling comparative analysis and supervised learning.

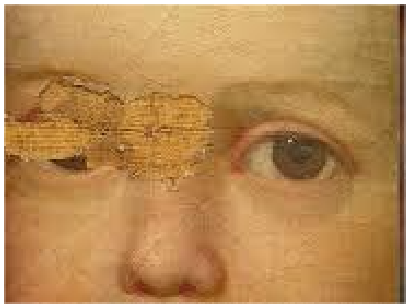

Fig. 1: Example from the paired dataset (damaged)

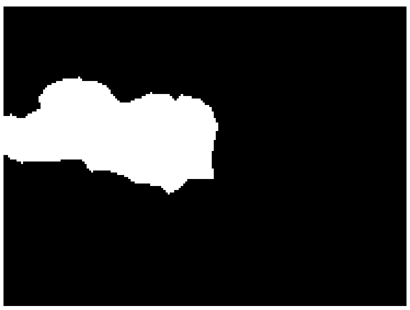

Fig. 2: Mask created using LabelMe

Fig. 3: Correspondent (undamaged)

The unpaired subset consists of standalone damaged artworks without reference images. To support segmentation tasks within the paired subset, we created our own damage masks by manually marking cracks and surface defects in the images using the LabelMe tool as in Figure 2. This gave us granular control over what constitutes 'damage' in our framework and enabled us to tailor the annotations to our objectives. However, the software's drawing mechanics limited the annotation process. It was laborious and imprecise to trace fine features, such as hairline cracks or irregular abrasions. The final masks therefore reflect a degree of interpretive subjectivity, introducing variability across samples.

### B. Dataset Card for "ARTeFACT"

The second dataset used in this study is the publicly available *Dataset Card for "ARTeFACT"*, hosted on Hugging Face [1]. It comprises 418 high-resolution images paired with pixel-accurate damage annotations, generated through standardized and consistent labeling protocols. These masks cover a variety of deterioration types, such as cracks, peeling, discoloration, and tearing, offering fine-grained supervision for training segmentation models.

Compared to other datasets, ARTeFACT provides greater annotation consistency and damage diversity, supporting experiments that require high spatial resolution and multi-class differentiation. However, its content spans multiple media formats, including photographs, posters, and mixed visual textures, many of which diverge from the aesthetic and material properties of painted artworks.

Although not entirely aligned with our dataset goals, ARTeFACT provided valuable diversity and pixel-level ground truth for initial experimentation, threshold calibration and robustness testing. When used judiciously, its inclusion enriched the capacity of the model to generalize, especially in the absence of large-scale annotated data specific to museum-grade paintings. Its very precise annotations allowed us to train models with better accuracy than the masks we created for the previous dataset.


(a) Original image (b) Correspondent mask

Fig. 4: ARTeFACT dataset

### C. Custom Dataset

In addition to the public datasets, we compiled a custom-made collection of 33 photographs, which were taken directly from our mobile phone camera rolls or found online. These images were captured under non-frontal conditions, alongside other artwork. These images were primarily used to test the homography and geometric rectification components of our pipeline, simulating practical scenarios commonly encountered in galleries, museums, or when taking personal photographs. Although no damage masks were available for this dataset, including it allowed us to evaluate the performance of perspective correction algorithms on distorted and unconstrained inputs. The custom images added an extra dimension to the preprocessing stage and helped to verify geometric transformations under more variable photographic conditions.

## III. Proposed Method

The proposed pipeline for artwork analysis and restoration consists of a multi-stage architecture designed to process photographic representations of paintings and identify visible forms of deterioration. The first component is an **image rectification module** that corrects perspective distortions introduced by oblique camera angles or curved canvases. This is achieved through contour extraction and projective homography, ensuring a frontal and standardized view of the artwork prior to further processing.

Following rectification, the pipeline diverges into two distinct branches for the segmentation of damaged regions, particularly cracks and surface anomalies. The first is a **rule-based approach**, built using classical image processing techniques such as contrast stretching, adaptive thresholding and morphological filtering. This branch wants to offer a lightweight and interpretable solution, suitable for scenarios with limited data. The second is a **deep learning-based strategy**, composed of supervised convolutional neural networks (CNNs), including a U-Net for pixel-wise segmentation and a ResNet for binary classification. These models leverage large-scale annotated datasets and residual connections to improve generalization and gradient stability during training.

The outputs of both segmentation paths wanted to be consolidated and serve as inputs for the final stage: **image inpainting**. In this phase, detected damage masks are used to guide visual restoration via diffusion-based techniques such as Telea's method. This ensures a coherent filling of deteriorated zones while preserving the artistic texture and color continuity of the original image. The modular design of the pipeline enables flexibility in experimentation, supporting comparative evaluation of segmentation strategies and seamless integration of damage detection with restoration.

## IV. Geometric rectification

Due to variable acquisition angles and non-frontal viewpoints, many images of paintings exhibit significant perspective distortions. To address this issue, we initially experimented with the GitHub project *Automated Rectification of Image* [2], which estimates vanishing points from detected line segments and applies a homographic projection that pushes one vanishing point to infinity to simulate parallel geometry. The approach builds a custom homography by modifying the third row of the matrix to flatten perspective cues and recover frontal views.

Unfortunately, this method produced unsatisfactory results in our context: when applied to photographs of artworks, the algorithm often failed to correctly localize the painting. The Original pipeline was based on edge detection using the Canny operator, which was ineffective in the presence of low-contrast boundaries, textured surfaces or uneven lighting. To overcome this limitation, we substituted adaptive thresholding and morphological closing to enhance contour visibility.

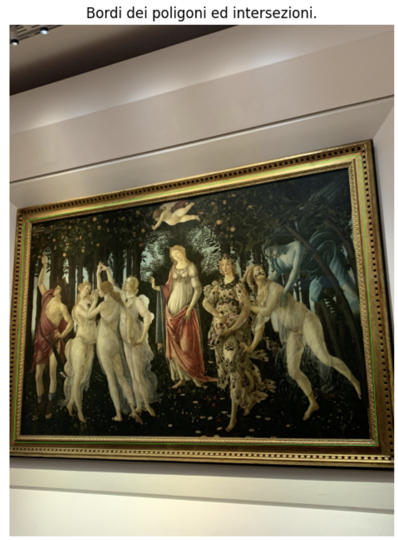


Fig. 5: Border of the painting


Fig. 6: Result of the rectification

We thus implemented a more robust pipeline tailored for artwork rectification. The image is preprocessed using bilateral filtering to preserve edges while reducing noise, followed by adaptive mean thresholding to extract high-contrast regions. Contours are computed and filtered using geometrical criteria, selecting only convex polygons with $4$ to $6$ vertices and a minimum area threshold. Once a valid quadrilateral contour is identified, we extract the ordered corner points $\{\mathbf{x}_i\}$ and compute a projective homography matrix $H \in \mathbb{R}^{3\times3}$ that maps the detected vertices to an ideal rectangular configuration $\{\mathbf{x}'_i\}$:

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \text{with} \quad (x', y') = \left(\frac{x'}{w}, \frac{y'}{w}\right) \tag{1}$$

In scenarios where the painting's contour is ambiguous or not strictly rectangular, we also estimate two vanishing points by computing intersections of extended edge pairs using linear algebra. A selected vanishing point $\mathbf{v}$ is then pushed to infinity via affine normalization, modifying $H$ such that the perspective convergence is suppressed. This refinement follows the transform:

$$H_{\text{vp}} = T_{\text{shift}} \cdot (T_{\text{center}}^{-1} \cdot H_{\text{mod}} \cdot T_{\text{center}}) \tag{2}$$

where $T_{\text{center}}$ centers the transformation around the image midpoint, and $H_{\text{mod}}$ encodes the directionality of the vanishing point. To ensure proper canvas alignment, the transformed image corners are analyzed and shifted via $T_{\text{shift}}$ to guarantee positive coordinates.

Finally, the frontal projection is computed by warping the image using the inverse homography:

$$I_{\text{rectified}}(x', y') = I(H^{-1}(x', y')) \tag{3}$$

This rectification workflow reliably restores frontal views of paintings, enabling standardized geometric conditions for subsequent stages such as damage segmentation, restoration, and visual analysis.

## V. Rule-Based Crack Segmentation Approach

Before adopting learning-based architectures for damage detection, we investigated a rule-based segmentation pipeline that used only classical image processing techniques. The primary objective was to identify surface deterioration, particularly cracks, using deterministic operators to avoid the need for labeled data and model training. This phase of experimentation was inspired by the Kaggle notebook *Projcv* [3], which provided a reference structure for modular image analysis pipelines.

The implemented approach consisted of sequential preprocessing steps including Gaussian denoising, contrast stretching, grayscale conversion and various thresholding algorithms (global, Otsu, and adaptive). The components were implemented using OpenCV and parameters were manually tuned to adapt to the visual characteristics of artwork images. Specifically:

- Noise Reduction: A Gaussian blur with kernel size $5 \times 5$ was applied to remove high-frequency components.
- Contrast Enhancement: Grayscale levels were linearly stretched to improve visibility of fine patterns and edge-like structures.
- Thresholding: Classical binarization techniques were evaluated. Initially, the pipeline employed Otsu's method to compute a global threshold by minimizing intra-class variance in the grayscale histogram. While theoretically optimal under unimodal intensity distributions, its application to artwork images proved limited: spatial variability in lighting and pigment density caused heterogeneous intensity peaks, resulting in masks that inadequately captured fine damage regions.

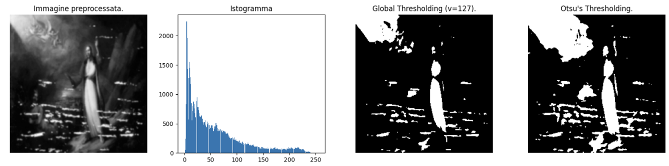


Fig. 7: Otsu thresholding

To improve spatial selectivity, the pipeline was subsequently extended with adaptive thresholding methods. Specifically, both mean-based and Gaussian-weighted variants were tested, computing per-pixel thresholds within local neighborhoods. These approaches offered improved responsiveness to regional contrasts and partially mitigated the oversegmentation observed with global binarization.

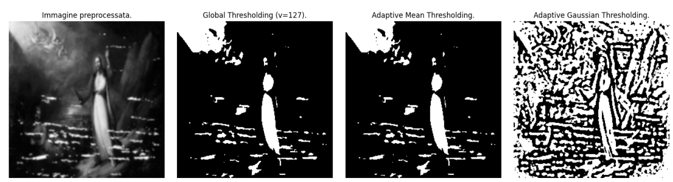


Fig. 8: Adaptive thresholding

Although the pipeline produced plausible masks in select scenarios, the results lacked generality across varied images. Thresholding algorithms responded primarily to intensity gradients, often misclassifying decorative textures, edges of brushstrokes, and frame contours as damage. Cracks were not consistently highlighted, and masks frequently contained spurious structures unrelated to actual deterioration.

To mitigate these limitations, we also explored semi-manual mask generation using image editing tools such as GIMP. Damage zones were manually isolated using curves, threshold levels, and brush tools, and the same operations were then reproduced with OpenCV to test automation feasibility. However, this method proved impractical for large-scale annotation: each image required bespoke parameters depending on contrast range, lighting conditions, medium type, and surface reflectivity. After extended experimentation, we concluded that the rule-based strategy was insufficient for robust segmentation. The high sensitivity to local contrast and lack of semantic understanding made it unsuitable for consistent application. Consequently, this exploratory branch was abandoned in favor of supervised deep learning solutions, which demonstrated superior generalization, reduced false positives, and better adaptation to heterogeneous input.

## VI. Deep Learning Approach

### A. Crack Classification with ResNet50

To address the binary classification task distinguishing cracked artworks from intact ones, we designed and trained a deep convolutional neural network based on the ResNet50 [4] architecture. ResNet50 leverages residual learning via skip connections that alleviate vanishing gradients, enabling efficient optimization of deep networks. The model is composed as follows:

*a) Dataset Preparation.:* The training corpus was constructed by aggregating labeled images from two directories: one containing images affected by cracks ($y = 1$), and the other containing healthy surface patches ($y = 0$). Images were resized to $224 \times 224$ pixels and converted into normalized tensors using the preprocessing function recommended by ResNet. A stratified data split was applied to preserve class balance across train (70%), validation (15%), and test (15%) partitions.

*b) Model Architecture.:* The core of the model consists of a ResNet50 backbone pre-trained on ImageNet, with frozen weights during the initial training phase. On top of the feature extractor, we appended a global average pooling layer, a fully connected layer with 256 ReLU-activated units, and a dropout layer ($p = 0.5$) for regularization. The final output is a single neuron with sigmoid activation for binary prediction. The model was compiled with binary cross-entropy loss and optimized using the Adam optimizer.

*c) Training Strategy.:* To address class imbalance, frequency-based sample weighting was applied during training. The procedure was conducted in two phases:

1) Initial training: The base ResNet50 was frozen while training the newly added layers for 10 epochs, with early stopping based on validation loss.
2) Fine-tuning: The last ten layers of ResNet50 were unfrozen to allow feature refinement. Training continued for 20 epochs with reduced learning rate ($1 \times 10^{-5}$), early stopping, and learning rate scheduling via `ReduceLROnPlateau`.

*d) Evaluation and Deployment.:* After training, model weights were saved and reused for subsequent inference. Final accuracy was assessed on a held-out test set, and performance was deemed stable. An interactive module was implemented for real-time classification, predicting whether an uploaded image contains surface damage. For any input image $I$, the binary prediction is computed via:

$$\hat{y} = \sigma(f_{\text{ResNet}}(I_{\text{norm}})), \quad \text{with} \quad \hat{y} \in [0, 1] \tag{4}$$

where $\sigma$ is the sigmoid activation and $f_{\text{ResNet}}$ is the forward pass through the trained network.

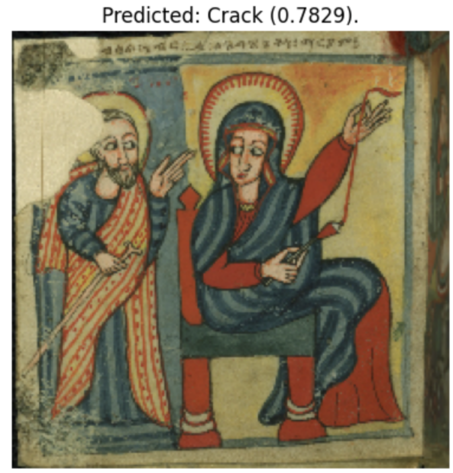


Fig. 9: Crack prediction

*e) Limitations.:* Despite the model producing promising results, it exhibited limitations when applied to ambiguous or visually complex samples. In particular, incorrect predictions occurred when surface textures were similar to mini-crack features or when contrast gradients compromised feature learning. These misclassifications can be attributed partly to the limited size and diversity of the dataset and partly to the approximate nature of the manually annotated masks used during training. The lack of pixel-perfect ground truth reduces supervision fidelity, thereby affecting the model's capacity to generalise to unseen patterns.

### B. Crack Segmentation with U-Net

To perform pixel-wise segmentation of damaged regions, we adopted the U-Net architecture, a symmetric encoder–decoder convolutional neural network originally designed for biomedical image segmentation and widely used in damage localization tasks. The model comprises a contraction path that captures contextual features via convolution and max-pooling operations, followed by an expansive path that enables precise localization using transposed convolutions and skip connections.

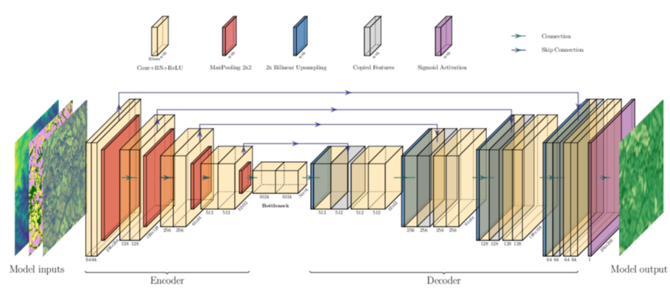


Fig. 10: U-Net architecture

Each block in the encoder consists of two consecutive convolutions with ReLU activation, followed by spatial downsampling. The decoder mirrors this structure with upsampling and concatenation of corresponding encoder features to recover spatial detail. Network input consists of RGB images that have been resized and symmetrically padded to $1024 \times 1024$ pixels using high-quality Lanczos interpolation. Input normalisation is performed on a per-channel basis using ImageNet statistics to support model generalisation.

To ensure deterministic behavior and reproducibility of results across hardware, all random seeds (Python, NumPy, PyTorch, CUDA) are explicitly fixed, and deterministic algorithms are enforced via low-level configuration parameters (‘torch.use_deterministic_algorithms‘, ‘CUBLAS_WORKSPACE_CONFIG‘, etc.).

The segmentation model was trained using a composite loss function that combines binary cross-entropy and soft Dice loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}} + \left(1 - \frac{2 \cdot \sum(p \cdot g) + s}{\sum(p + g) + s}\right) \tag{5}$$

where $p$ is the predicted probability, $g$ the ground truth mask, and $s$ a smoothing constant to prevent division by zero. Notably, the binary cross-entropy loss $\mathcal{L}_{\text{BCE}}$ is computed exclusively over valid mask regions to exclude padded borders from optimization.

Training is performed using the Adam optimizer [5] with a learning rate of $1\text{e}{-4}$ and mixed-precision updates via gradient accumulation over four steps. A model checkpoint is saved only when validation loss improves, avoiding overfitting. The network was trained on the ARTeFACT dataset, selected for its highly accurate annotated damage masks. Although the dataset includes images not fully representative of our specific use case, its segmentation quality provided a strong foundation. The trained model effectively highlights crack patterns resembling those found in ARTeFACT annotations, enabling generalization to structurally similar domains.

During inference, the output probability maps are binarized using a threshold of $0.23$, followed by morphological opening and closing to suppress noise. Small isolated regions are removed, and slight dilation ensures complete coverage of crack structures. The final segmentation is then overlaid onto the original image to facilitate visual inspection and quantitative analysis. The model delivers high-fidelity segmentation, accurately delineating fracture contours consistent with the training annotations, particularly on images containing visible and contrast-rich cracks, visually similar to those present in the ARTeFACT dataset, as illustrated in Figure 11.


Fig. 11: U-Net crack segmentation

### C. Inpainting of damaged regions

To restore visual consistency in artwork images affected by cracks or surface deterioration, we implemented a post-processing inpainting step based on predicted damage masks. These masks were derived from segmentation outputs and binarized using a threshold value $t = 0.09$:

$$M(x, y) = \begin{cases} 1 & \text{if } P(x, y) > t \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $P(x, y)$ is the predicted crack probability at pixel $(x, y)$, and $M$ is the resulting binary damage mask. We applied OpenCV’s inpainting algorithm using Telea’s method [6], a fast-marching strategy that propagates surrounding pixel information inward along isophote directions. The method is well-suited for fine restoration tasks due to its speed and edge-aware diffusion behavior.

Formally, the reconstructed image $I_{\text{inpainted}}$ is obtained via:

$$I_{\text{inpainted}} = \text{Inpaint}_{\text{Telea}}(I_{\text{original}}, M, r) \tag{7}$$

where $r$ is the inpainting radius and $I_{\text{original}}$ is the RGB input image resized to $256 \times 256$ for consistency with the inference pipeline [7]. In Figure 12 is possible to appreciate the result of this inpainting method. We are satisfied so we decided to include it in our final project


Fig. 12: Telea inpainting result

We also experimented with the LaMa (Large Mask Inpainting) framework [8], a recent deep-learning-based approach relying on Fourier convolutions and perceptual priors. Although LaMa performs well for large structured content removal, it proved less effective on small-scale crack restoration. In our tests, LaMa occasionally introduced visible texture inconsistencies and semantic hallucinations, whereas Telea’s diffusion maintained visual coherence without altering the artistic style.

## VII. Results and Future Improvements

The proposed multimodal pipeline for damage analysis and restoration produced satisfactory results, particularly in the deep learning segment. The segmentation models based on U-Net and ResNet, both incorporating residual connections, successfully mitigated gradient vanishing and offered reliable crack localization. Visual inspection confirmed high-fidelity detection of damage regions, especially in areas structurally similar to the training set, such as those found in ARTeFACT.

The threshold-tuned visualization strategy further enhanced interpretability and supported qualitative damage assessment.

Nonetheless, the rule-based segmentation path—untrained and sensitive to lighting and contrast—remains the main bottleneck in generalization. Future improvements could involve dimensionality reduction techniques like Principal Component Analysis (PCA) combined with Local Binary Patterns (LBP) to better isolate texture anomalies without relying on intensity distributions. Additionally, object detection frameworks could be integrated to subtract semantic foreground elements (e.g., figures, ornaments) from damage masks, thus refining prediction to highlight only structurally relevant cracks. For the classification pipeline based on ResNet, future work should focus on expanding the dataset and refining annotation protocols to improve both spatial resolution and semantic accuracy. Further architectural advances may consider transformer-based models with self-attention mechanisms, capable of modeling long-range dependencies and selectively enhancing attention toward fine deterioration features in complex artistic compositions.

Beyond segmentation, future experimentation in the restoration stage may benefit from alternative inpainting strategies. Methods such as PatchMatch, which propagates texture from neighboring undamaged regions via randomized patch sampling, could offer context-aware reconstruction without requiring deep training. Additionally, learning-based inpainting models fine-tuned on artwork textures may improve stylistic coherence and semantic plausibility, especially when operating on highly structured or color-sensitive surfaces. These strategies could enhance the perceptual quality of restoration while reducing visual artifacts introduced by traditional diffusion techniques.

## VIII. Conclusion

This project highlights the feasibility and benefits of employing automated tools to support experts in heritage conservation, offering scalable and interpretable solutions for damage assessment. By bridging classical vision techniques with deep learning, the pipeline provides a versatile framework that in future could be applicable in museum environments, research settings, and digital art preservation workflows. The approach aligns with the broader goal outlined at the beginning: to reduce the burden of manual inspection while improving objectivity and reproducibility in restoration planning.

Despite current limitations, especially in untrained and rule-based stages, the promising results achieved through learning-based architectures and standardized datasets indicate a viable path forward. Future enhancements, guided by richer annotations, expanded datasets and advanced architectures, will pave the way for increasingly intelligent, accurate and context-aware restoration systems. In this light, the integration of such multimodal tools can serve not only as technical solutions but as assistive instruments for art historians, conservators and students in preserving cultural memory through digital innovation.

## References


[1] I. Daniela, "Dataset card for "artefact"," https://huggingface.co/datasets/danielaivanova/damaged-media, Vis. June 2025.

[2] S. Chilamkurthy, "Automated rectification of image," https://github.com/chsasank/Image-Rectification, Vis. June 2025.

[3] Y. B. Vedangit, "proj cv notebook," https://www.kaggle.com/code/vedangit/proj-cv/notebook, Vis. May 2025.

[4] PyTorch, "Resnet50," https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html, Vis. July 2025.

[5] ——, "Adam," https://docs.pytorch.org/docs/stable/generated/torch.optim.Adam.html, Vis. June 2025.

[6] A. C. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004. [Online]. Available: https://www.olivier-augereau.com/docs/2004JGraphToolsTelea.pdf

[7] OpenCV, "Telea," https://docs.opencv.org/3.4/df/d3d/tutorial_py_inpainting.html, Vis. June 2025.

[8] S. Lopatin *et al.*, "Image inpainting via lama," https://github.com/advimman/lama, Vis. July 2025.