applied sciences

MDPI

Article

Evaluating Activation Functions in GAN Models for Virtual Inpainting: A Path to Architectural Heritage Restoration

Ana M. Maitin [1], Alberto Nogales [1], Emilio Delgado-Martos [2], Giovanni Intra Sidola [2], Carlos Pesqueira-Calvo [2], Gabriel Furnieles [1] and Álvaro J. García-Tejedor [1,*]

[1] CEIEC Research Institute, Universidad Francisco de Vitoria, Ctra. M-515 Pozuelo-Majadahonda km. 1, 800, Pozuelo de Alarcón, 28223 Madrid, Spain; a.maitin@ceiec.es (A.M.M.); alberto.nogales@ceiec.es (A.N.); gabifurnielesgarcia@gmail.com (G.F.)

[2] Architecture School, Universidad Francisco de Vitoria, Ctra. M-515 Pozuelo-Majadahonda km. 1, 800, Pozuelo de Alarcón, 28223 Madrid, Spain; e.delgado.prof@ufv.es (E.D.-M.); giovanni.intra@ufv.es (G.I.S.); c.pesqueira.prof@ufv.es (C.P.-C.)

* Correspondence: a.gtejedor@ceiec.es

Abstract: Computer vision has advanced much in recent years. Several tasks, such as image recognition, classification, or image restoration, are regularly solved with applications using artificial intelligence techniques. Image restoration comprises different use cases such as style transferring, improvement of quality resolution, or completing missing parts. The latter is also known as image inpainting, virtual image inpainting in this case, which consists of reconstructing missing regions or elements. This paper explores how to evaluate the performance of a deep learning method to do virtual image inpainting to reconstruct missing architectonical elements in images of ruined Greek temples to measure the performance of different activation functions. Unlike a previous study related to this work, a direct reconstruction process without segmented images was used. Then, two evaluation methods are presented: the objective one (mathematical metrics) and an expert (visual perception) evaluation to measure the performance of the different approaches. Results conclude that ReLU outperforms other activation functions, while Mish and Leaky ReLU perform poorly, and Swish's professional evaluations highlight a gap between mathematical metrics and human visual perception.

Keywords: architecture; cultural heritage; virtual restoration; deep learning; inpainting; generative adversarial networks

Citation: Maitin, A.M.; Nogales, A.; Delgado-Martos, E.; Intra Sidola, G.; Pesqueira-Calvo, C.; Furnieles, G.; García-Tejedor, Á.J. Evaluating Activation Functions in GAN Models for Virtual Inpainting: A Path to Architectural Heritage Restoration. *Appl. Sci.* **2024**, *14*, 6854. https://doi.org/10.3390/app14166854

Academic Editors: Rui Marques, Claudia Mondelli and Maria Giovanna Masciotta

Received: 12 April 2024
Revised: 27 July 2024
Accepted: 27 July 2024
Published: 6 August 2024

1. Introduction

Cultural heritage faces many threats to its security, both environmental, such as pollution and climate change, and man-made, such as the intentional destruction of cultural heritage. In this scenario, technology gives us unprecedented opportunities to preserve and protect it. Digital technologies, from 3D simulation to artificial intelligence and virtual/augmented reality, are being used to ensure cultural heritage preservation.

Among the tasks accomplished during the conservation of cultural heritage, restoration stands out. In particular, virtual restoration, a type of restoration, obtains images of the original state of a building by analyzing its ruins or a fragment of the building. These virtual restorations were carried out using the techniques of representation and planimetric restitution of the time, such as painting, engravings, and models. In all cases, the starting point was a theoretical assumption of the original state. This assumption was based on the buildings that maintained part of their integrity, on drawings and engravings from other periods, and even on texts and manuscripts that described the structure under study with greater or lesser fidelity [1].

Virtual restoration of historic buildings is a powerful tool for transferring knowledge to society. By combining advanced technology with the richness of cultural heritage,

new possibilities open for (a) education, (b) preservation, and (c) cultural promotion. These practices not only help keep the past alive but also enhance our understanding and appreciation of the cultural legacy we have inherited. In the field of education, these tools improve (i) accessibility to cultural content, allowing people from all over the world and of all backgrounds to access and explore historic buildings without the need to travel or create complex infrastructure to accommodate the sites. Additionally, the (ii) interactivity of these interventions can offer contextual historical, aesthetic–artistic, and architectural information. Finally, (iii) immersion in virtual reality (VR) and augmented reality (AR) environments provides experiences that can be more effective than traditional methods. Regarding heritage preservation, virtual restoration serves as a (i) detailed record of the current state of the building and allows for monitoring the lifespan of its construction elements, enabling the identification of parts at risk of deterioration to implement appropriate preventive measures. The virtual restoration models also offer significant possibilities to be used as (ii) references in physical restoration projects, providing very precise information about the original state. In terms of cultural promotion, virtual restorations can enhance (i) the attraction of cultural tourism to many regions, especially rural ones, in an innovative way and without the need for large investments in museum infrastructure. Moreover, mobile applications and AR-based tourism guides can provide (ii) personalized and enriched experiences for all types of visitors.

Artificial intelligence (AI) could play an important role in heritage preservation, not only for its use in the study of physical remains but also for its application in virtual restoration. Among all AI techniques, deep learning (DL) models have stood out for obtaining very good results in the field. DL was defined by [2] as those models that can learn the representation of a dataset using different levels of abstraction. Image restoration is linked to the virtual restoration of cultural heritage, which, in the case of DL models, can be framed as image-to-image translation.

Image-to-image translation is a method that uses data to translate the possible representation of a scene to another representation, usually by super-resolution, colorization, and inpainting. In [3], the aim of the latter is described as starting from a region of missing pixels, generating those that are realistic and semantically coherent with the existing ones. In our case study, starting from the image of a render representing a building in ruins, we intend to obtain the image of the restored building. As we do not have missing pixels but missing architectonical elements, we have defined our work as virtual restoration inpainting, a method to synthesize objects from pixels so that these objects are semantically coherent with the context represented in the rest of the image.

The motivation for this work arises because when it comes to virtual restoration. In this way, the appearance that the original construction must have had is based on assumptions and, therefore, on an uncertain solution. However, in some cases, references to determine the original state may be evident. For example, using historical photographs, in most cases, there is no reliable information available on how the missing elements could have been organized. The thesis we put forward is that the architectural language, based on a repetition of constructive, compositional, and structural patterns, can be decisive in intuiting effective solutions for virtual reconstruction. The study of the architectural language of a period or style from the analysis of these patterns can help to consolidate a new and more efficient reconstruction methodology by benefiting from deep learning models. The problem is that we need evaluations adapted to the problem of reconstructing the missing architectural elements of a building in such a way that they make sense from an architectural point of view.

The work presented in this paper is based on [3], where different methods were used to restore images of renders representing Greek temples in ruins. In that previous paper, one of the methods was based on the use of a segmented image in which the different architectonical elements were marked with different colors. Although this method obtains really good results, using this segmented image supposes a high expenditure in computational and human resources and time. Our research now explores approaches

based on direct training of GAN (Generative Adversarial Network) models, which means that from only the image of the render of the ruined Greek temple can we obtain its restored version. To this end, we have improved our dataset and fine-tuned the neural models by using different sets of hyperparameters and testing different activation functions among the most up-to-date options currently used in other DL fields.

Although this is an improvement from the previous work, the main innovation lies in the process used to evaluate the results of the task of reconstructing Greek temples from images of 3D renders. As the main objective is to reconstruct the architectonical elements, we provide a method to evaluate this task, avoiding problems that arise from the generation of artifacts, image quality, and other issues. These results are assessed through objective (using mathematical metrics) and subjective (asking an expert in the field) evaluations to measure the strengths of direct training that does not have intermediate steps as using segmented images.

As mentioned above, this work consists of a method that receives an image of a render representing a Greek temple in ruins and can directly detect which are the missing architectonical elements, returning the same image with the restored temple. For this purpose, we have used generative adversarial networks (GANs) that have been trained with only pairs of an image of the ruined temple and its corresponding image with the complete temple. Following, we list some works related to the actual research.

As this work can be framed in the field of data-driven image analysis, we can highlight the following works. In [4], a survey of data drive image methods using Machine Learning or Deep Learning methods applied to digital restoration in the field of cultural heritage is presented. A data-driven methodology for managing and conserving archaeological heritage sites using digital surveying tools and Heritage/Historic Information Modeling (HBIM) within a BIM environment is introduced by [5]. In [6], the parametric modeling of heritage buildings using Terrestrial Laser Scanning (TLS) or Structure from Motion (SfM) data within a Rhino + Grasshopper-ArchiCAD workflow for HBIM projects is enhanced. Apart from that, there are not many papers on data-driven image analysis in the field.

Deep learning has been a milestone for the restoration image task. In [7], coronary computed tomography angiography was restored by reducing noise using deep learning-based image restoration (DLR). Similar to this, a deep learning model is used in [8] to remove honeycomb patterns and improve resolution in another type of medical image, a fiber bundle. Another restoration image work with medical images is [9], where low-dose CT (LDCT) images are denoised by using GAN training with autoencoders and CNNs. FormResNet is a CNN-based model presented by [10] that restores images by learning the structured details and recovering the latent clean image. Other works using similar approaches but applied to X-ray Computed Tomography (CT) are [11,12]. Finally, a model based on Transformers is applied for image restoration in [13].

In the case of architecture and cultural heritage, we find a review paper authored by [14]. In [15], GANs are used for the reconstruction of 3D models of architectural elements. Another work is [16], which presents a semi-supervised 3D model reconstruction framework with GAN models. It is applied to the reconstruction of scenes containing architectonical buildings. An approach aimed to reconstruct the faces of statues with a particular type of GAN called Wasserstein is described in [17]. The research proposed by [18] uses CNNs to classify cultural heritage images, enhancing image management and search efficiency. The paper by [19] presents ReCRNet, a deep learning model that outperforms existing methods in detecting concrete cracks, making it highly effective for monitoring the structural health of historical buildings. Then, an automated deep learning system using Faster R-CNN to detect and classify four types of damage in outdoor stone cultural properties is presented in [20]. Also, there is a study by [21] that presents a custom YOLOv5 deep learning model for detecting and localizing four types of defects in cultural heritage structures. Finally, in [22], a CNN system with transfer learning is developed to assess the condition of building facades in the historic Lasem District (Central Java).

This work builds on the methods presented in [3], with a focus on the performance using the direct method. This method, which does not rely on segmented images, reconstructs the temple from a single image of the ruins. In contrast, the segmented method described in [3] involves two steps: first, segmenting the image by architectural elements, and second, reconstructing the temple using the segmented image, which is less efficient in computational terms. Testing various hyperparameters and analyzing the results from an architectural perspective are key aspects of this study. The evaluation is based on subjective and objective methods that consider the reconstruction of the architectonical elements considering the problems produced by graphic artifacts and other issues.

The rest of the paper is structured as follows. Section 2 describes the dataset used to train the models and defines the methods applied in the work. Section 3 compiles all the results related to the training workflow and both evaluations: the objective and the subjective. Finally, Section 4 obtains some conclusions and points out several future works.

2. Material and Methods

2.1. Dataset Description

The architecture of the classical Greek period, especially the Doric order, is sufficiently systematic to attest to constructive and compositional patterns based on the architrave scheme formed by columns and entablatures. The dataset incorporates 30 buildings from this period that correspond to different configurations of the classical temple, depending on the number of columns and their organization around the sanctuary wall.

All these buildings have been modeled in 3D in their original state and progressively destroyed in three stages to enrich the number of possibilities for viewing and therefore analyzing the ruined structure. This decision was based on the impossibility of having pairs of photographs of restored temples and their previous ruins. The software used in this case was SketchUp (https://www.sketchup.com/, accessed on 11 April 2024), a 3D modeling program, V-Ray (https://www.chaos.com/es/vray/sketchup, accessed on 11 April 2024). NEXT version, computer-generated rendering software, and 3DSMax (2023 Autodesk).

For the time being, this study focuses on the formal and volumetric aspects of the building and, therefore, has not incorporated added sculptural elements in pediments, metopes, and acroteries, nor the striking polychromies that covered the stone that made up the temple. In this dataset, care has been taken in the application of more realistic stone textures in conjunction with a global lighting system that provides more nuances to the areas in shadow and half-light, being able to distinguish many more elements.

The position of the cameras that focus on the 3D model, which is used to obtain images, has also been studied. To obtain better images, care has been taken with the framing, maintaining the verticality of columns and walls, with the aim of not forcing the perspective and deforming the architectural elements excessively. Taking advantage of a complete circular path of the camera around the building, 360 images (512 $\times$ 512 px) have been obtained, capturing all the material and light nuances of the model as shown in Figure 1. This has provided 43,200 images to feed the learning of the neural network. As the perspectives of the four corners could lead to confusion in the recognition and computation of columns, it was decided to manually eliminate some of the images of that area, which in Figure 1 is called “threshold”.

In addition, a realistic environment has been modeled using a 3D mesh, incorporating a terrain texture, which serves as a background for the images of the temples. The ground on which the plinth rests also has some irregularity that interacts with the model and the shadows cast from it. The presence of the landscape is fundamental to the analysis of the images as it will force the neural model to discern the figure (the building) from the background. Figure 2 shows an example of a pair of ruined and restored temples.

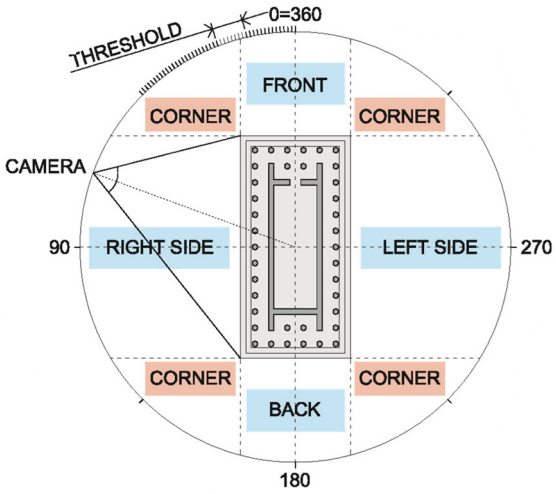

Figure 1. Circular path of the camera around the building.

Figure 2. An example of a ruined temple and its complete version.

To evaluate how the missing architectural elements can influence the performance of the method, each instance of the temples has been destroyed considering three different degrees. In Figure 3, we show an original complete temple and its different degrees of destruction, from less destructed to more destructed.

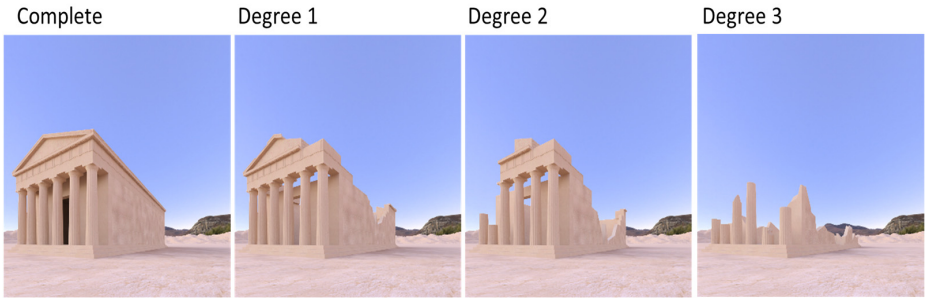

Figure 3. An example of a temple and its different degrees of destruction.

2.2. Methods

In this section, we provide a formal description of all the deep learning methods and parameters related to the proposed trainings for virtual inpainting restoration. First, the pre-processing methods applied so the models could be trained faster and more accurately are presented. Second, GAN networks are defined, and the architectures of the discriminator and generator are detailed. Third, the activation functions that are evaluated in this work are mathematically formalized. Finally, the training process for the GAN is described step by step, detailing the role of the dataset, the generator, and the discriminator.

2.2.1. Data Pre-Processing

The proposed training needs some transformation to optimize it. Before introducing the data into the model, we normalized the images that were in a range from 0 to 255, transforming them to a range of $[-1, 1]$.

2.2.2. Generative Adversarial Networks

The training was conducted using GAN, which is a method introduced by [23], and included two neural models that compete to generate and improve synthetic data. The first model, the generator, creates new instances. The second model, the discriminator, evaluates the created data. Then, both models compete, so the generator tries to convince the discriminator that the synthetic data (created by the generator) belong to the initial dataset.

2.2.3. Generator and Discriminator Architectures

We have decided to use a similar architecture for both the generator and the discriminator. These architectures consider pix2pix as a starting point [24]. The generator uses an autoencoder, while the discriminator is of Markovian type.

Autoencoders are based on the use of convolutional–deconvolutional architectures of 2 dimensions. They were introduced by [25] and are aimed to receive and input data that are downscaled until a minimal piece of data represents its essential information, which is later upscaled to obtain the input data or an expected one. As the aim of our work is the virtual reconstruction of images containing ruined temples, this architecture fits perfectly by using these images as input data and returning the image of the same temple but adding the missing architectonical elements. Apart from that, we are adding skip connections to the model.

A Markovian decoder is a particular type that does not evaluate the generated images as a whole but evaluates different patches separately. By doing this, the discriminator has more capacity to evaluate local textures, which is important in this case. The discriminator uses a sliding window that considers the local continuity and context details. This is important when evaluating how they have been added to the missing architectonical elements. The sliding window consists of an NxN patch that goes through the whole image and evaluates whether each part belongs to an original image or a created one. In summary, the discriminator uses a 2D CNN architecture to obtain the main features of the image and then a Markovian patch that evaluates them as belonging to the original dataset or being synthetic. The result is a matrix with values going from 0 to 1, depending on the previous evaluation. If the value is near this part of the image, it is synthetic, and if it is near 1, the part of the image is original.

2.2.4. Activation Functions

Deep learning models comprise a wide range of hyperparameters that must be tuned to improve their performance. The choice of the proper activation function has been demonstrated to be critical [26]. Activation functions are defined by [27] using the following equation.

$$g: R \rightarrow R \quad (1)$$

g is a function that is differentiable almost everywhere. Its behavior consists of computing the weighted sum of input features plus a bias to decide whether the neuron is activated. A way to measure the performance of direct training is to test different activation functions and evaluate them. In this case, we have decided to use the Rectified Linear Unit (ReLU), Leaky ReLU, Swish, and Mish. In the following, we mathematically formalized these activation functions.

ReLU was introduced by [28], which thresholds the output value at 0 when the calculations in the neuron are smaller than 0 and the value itself when it is greater or equal to 0. Equation (2) formalized this activation function.

$$y_i = \begin{cases} x_i \text{ if } x_i \geq 0 \\ 0 \text{ if } x_i < 0 \end{cases} \quad (2)$$

Leaky ReLU is a modification of ReLU introduced by [29]. Instead of returning 0 in the first condition, it produces a small gradient. Equation (3) describes this activation function.

$$y_i = \begin{cases} x_i \text{ if } x_i \geq 0 \\ \frac{x_i}{\alpha_i} \text{ if } x_i < 0 \end{cases} \quad (3)$$

In this equation, α_i is a parameter with a fixed value that ranges from 1 to ∞.

Swish is an activation function described by [30]. It seems to be a nonlinear interpolation between the linear function and ReLU with smooth behavior. Equation (4) describes this activation function.

$$y_i = x \cdot \sigma(\beta x) \quad (4)$$

In the equation above, $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function, and β is a constant or trainable parameter.

Mish is a self-regularized non-monotonic function introduced by [31], whose mathematical formalization can be found in Equation (5).

$$y_i = x\tanh(\text{softplus}(x)) = x\tanh(\ln(1 + e^x)) \quad (5)$$

2.2.5. The Training Workflow

As said above, this paper aims to evaluate and explore the performance of direct training that receives the image of a ruined temple and returns that temple reconstructed by adding the missing architectonical elements. So, for the training, we need pairs of a temple in ruins and the same temple complete. This algorithm comprises 4 stages that are repeated for each epoch with batches of n images from the training dataset. The 4 stages are described as follows:

1. n images of ruins $\{x_1,\ldots,x_n\}$ are taken from the training dataset;
2. n equivalent images of the reconstructed temple $\{y_1,\ldots,y_n\}$ are taken from the training dataset;
3. The generator's hyperparameters are tuned with the gradient descent;
4. The discriminator's hyperparameters are tuned with the gradient descent.

In depth, the training dataset is introduced in the generator autoencoder until it finds a mapping between both types of images (the image with the temple in ruins and its reconstruction). This is achieved by benefiting from the encoder, which reduces the input data size by applying convolutional layers (encoder) until the information is reduced to a small piece of data containing the main features. Then, this piece of data is upscale (decoder), trying to convert it to the output data (restored temple). This output image is what needs to be evaluated. This task is performed by the discriminator, which will evaluate if the image belongs to the original dataset or has been created by the generator. This information provided by the discriminator is used by the generator to improve its performance by tuning its hyperparameters. Figure 4 describes the training process.

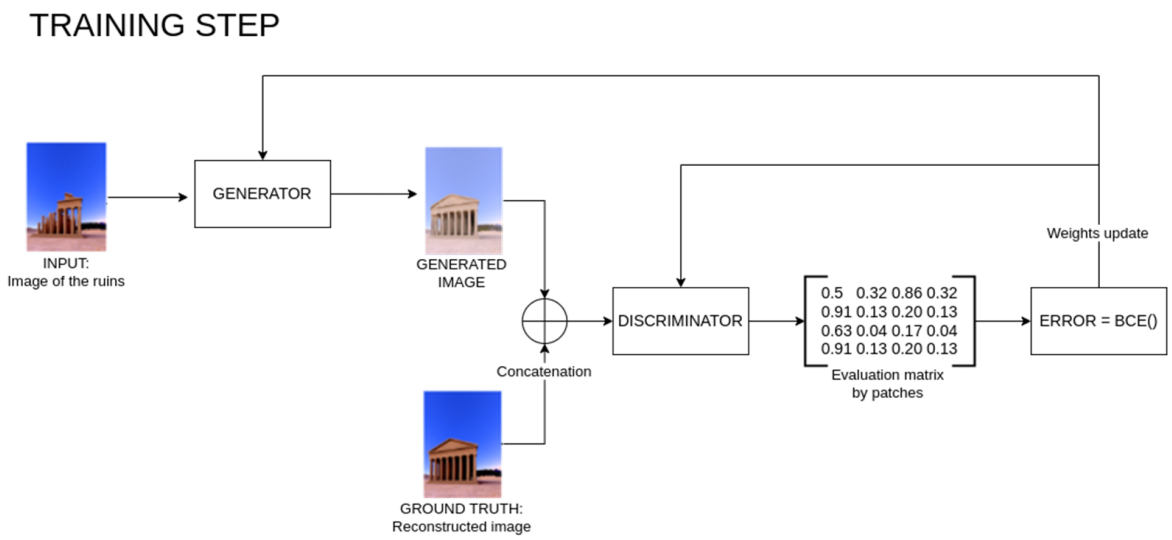

Figure 4. Workflow of training stage.

If we consider $x \in X$ as the input image of the model that represents the temple in ruins belonging to the training dataset, then $y \in Y$ will be the expected results, which, in this case, is the same temple with missing architectonical elements. The generator G reconstructs image y_{gen} and also denotes $G(x) = y_{gen}$. The discriminator produces two outputs that measure the distances between images. First, how different are the image of the ruined temple x and its corresponding image of a complete temple y denoted by $D([x,y]) = m_{real}$. Second, the same calculation between the ruined temple and the reconstruction made by the generator, $D([x,y_{gen}]) = m_{gen}$. In both cases, the outputs represent matrices of 0's and 1's. To evaluate the generator's loss, the following function is used.

$$G_{loss} = -\frac{1}{n}\sum_{i=1}^{n} \log(D(x, G(x))) + \lambda L_1(G) \quad (6)$$

where $\mathcal{L}_1$ is the L_1 distance between the expected value y and G(x) to minimize it.

Then, the loss of the discriminators is evaluated. For this task, both the generated image and the expected image (one from the training dataset) are used, and D_{loss} is evaluated using the following equation that scales the results by $\frac{1}{2}$ based on the pix2pix paper.

$$D_{loss} = -\frac{1}{2}[-\sum_{i=1}^{n} \log(D(x, y)) + \log(1 - D(x, G(x)))] \quad (7)$$

To evaluate the direct training, we have designed several experiments that differ in the activation function. During this training, the dataset was split into training, validation, and test by using 2 complete temples for validation and 1 for the test. This decision has been taken considering the similarity between the images of one temple. The test temple has not been chosen randomly due to this condition. In this way, we have ensured that this temple is sufficiently different from those used in training. In terms of number, the training set comprises 9720 images, the validation set 720, and the test set 360.

2.3. The Architecture of the Models

The GAN architecture is used to generate new realistic data similar to the training set through two main components that are trained together: the generator and the discriminator. The generator's main objective is to create realistic samples of the data, while the discriminator's objective is to identify whether the input data are created by the generator or belong to the training set. The training process involves a competitive method where both models strive to outsmart each other. This approach enhances the results and achieves a higher degree of realism.

The hyperparameters of both architectures were obtained after using a grid search strategy, which consists of creating a set of hyperparameters and combining different values to obtain the model with the best performance [32]. Values used for this hyperparameter tuning are compiled in Table 1.

Table 1. Hyperparameters and values for the grid search.

Hyperparameter	Values
Learning rate	2×10^{-4}, 2×10^{-3}
Batch	6, 14
Normalization types	Instances, Batch
Optimizer	RSMProp, Adam

From the table above, we can define the different terms. The learning rate quantifies the step size during model training to achieve a local minimum [33]. In this case, it can take the values 0.0002 and 0.002. Batch refers to a subset of the training dataset used to train the model during one iteration of the training process, which can be 6 or 14 pairs of images. Then, normalization techniques are used to improve the performance and training stability of the models. In this case, we have used Batch Normalization and Instance Normalization. The former normalizes the inputs of each layer across a batch of data [34]. The latter normalizes the inputs of each layer for each instance (sample) individually rather than across the batch [35]. Finally, ref. [36] defines the optimizer as the algorithm used to adjust the weights and biases of a neural network to minimize the loss function during training. In this case, we have used RSMProp and Adam. RMSprop is an optimization algorithm that modifies the learning rate of each parameter individually based on a moving average of the squared gradients [37]. Adam is an adaptive learning rate optimization algorithm tailored for training deep neural networks [38].

The generator in our model begins with an input layer sized 512 × 512 × 3 due to the colored images. This input is processed by convolutional blocks, each consisting of a 2D Convolutional layer, Batch Normalization layer, and Leaky ReLU activation function. The number of neurons in these blocks increases as follows: 64, 128, 256, 512, 512, 512, 512, and 512. After reducing the image to a smaller representation, the model then upsamples the data back to the original image size using deconvolutional blocks. These blocks contain a Transposed 2D Convolutional layer, a Dropout layer, and the activation function to evaluate. The neuron counts for these blocks are 512 for the first four, then 256, 128, and 64 for the subsequent ones. The final layer maps the output to RGB channels using a 2D convolutional layer with three neurons and a hyperbolic tangent activation function. Skip connections are used between corresponding convolutional and deconvolutional blocks to preserve features lost during downsampling.

The discriminator has two input layers, each of size 512 × 512 × 3, one for the generated image and one for the original image. These inputs are combined into a single layer of size 512 × 512 × 6 and processed through a series of convolutional blocks, each with a 2D Convolutional layer, Batch Normalization layer, and Leaky ReLU activation function. The neuron counts in these blocks are 64, 128, 256, and 512. Following this, a final 2D convolutional layer with a sigmoid activation function and a 4 × 4 filter outputs a 62 × 62 × 1 probability map, with values ranging from 0 to 1, indicating whether the input image is real or generated.

2.4. Objective Evaluation

The initial evaluation of the models is conducted using mathematical functions to obtain an objective assessment. We measure the pixel-wise accuracy by comparing the generated images with the expected images. An ideal generated image would have all its pixels exactly matching the corresponding pixels in the expected image. For this evaluation, we utilized images of three temples that were not included in the training phase. Each temple has three levels of destruction, resulting in a total of 1080 images (3 levels × 360

images per level). From this dataset, we randomly selected 45 images from each perspective of the temple for evaluation and calculated the pixel match accuracy. This process was repeated for each model trained with different activation functions. Additionally, we calculated the structural similarity (SSIM) index, which assesses the structural similarity between images while separating the effects of luminance and contrast. The formula for this metric is provided in the following equation.

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \quad (8)$$

In this equation, x and y are the images, l is the luminance, c is the contrast, and s is the structure. Then, the influence of these characteristics is provided by alfa and β, whose values are greater than zero.

The generative models have a problem when creating new images; in some cases, artifacts like flashes or sparkles can be included. This fact adds noise that can be reflected in the evaluation of the generated image, but it does not have to affect the main aim of the work, which is the reconstruction of the architectonical elements. Due to this issue, we proposed two strategies to evaluate the temple reconstruction: one using the whole generated image and another that applies a mask and only considers the non-matched pixels (that is, the part corresponding to the architectonical missing elements in the image and that the model seeks to reconstruct) between the image with the complete temple and the image with the temple in ruins. An example of the obtained mask is shown in Figure 5.

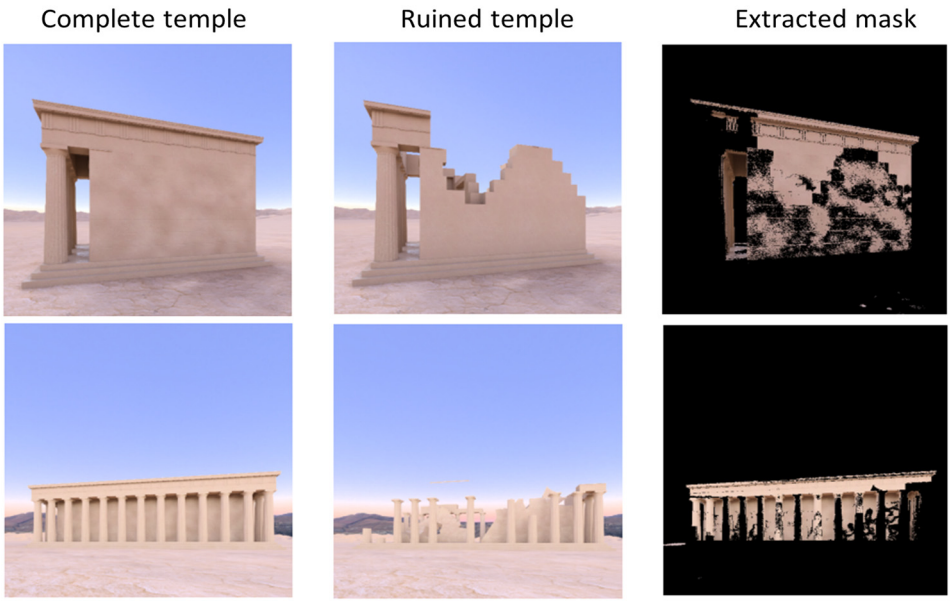

Figure 5. Examples of how the mask is obtained in this strategy.

3. Results and Discussion

We first implement the strategy that allows for the evaluation of the whole generated image, which is why it considers not only the reconstruction of the temples but also the artifacts generated in the whole image. Since the images share a large percentage of pixels, we have calculated a baseline to compare the results objectively, which measures the difference in pixels between the image with the complete temple and the image of the ruined temple. This is a way to know the thresholds from which the reconstruction begins. Table 2 compiles this information considering the three degrees of destruction and the different perspectives using the whole image. Values compute the average values and standard deviations.

Table 2. Baseline measures for SSIM using the whole image.

Perspective	Degree 1	Degree 2	Degree 3
Frontal	95.91% ± 0.87	87.11% ± 5.29	77.65% ± 7.03
Front right vertical	96.68% ± 2.00	90.06% ± 5.84	82.10% ± 9.47
Right side	96.48% ± 3.05	90.54% ± 5.60	83.90% ± 8.36
Rear right vertical	95.22% ± 3.83	86.04% ± 9.83	80.05% ± 9.71
Rear	93.54% ± 3.37	81.93% ± 13.04	77.36% ± 11.53
Rear left vertical	96.58% ± 2.12	87.15% ± 9.16	83.83% ± 9.77
Left side	88.28% ± 0.59	91.60% ± 3.75	87.85% ± 5.32
Front left vertical	98.20% ± 0.83	91.17% ± 4.19	84.64% ± 6.15
Total	**96.36% ± 1.44**	**88.2% ± 3.07**	**82.17% ± 3.39**

Results support the evidence that the reconstruction from degree 1 to degree 3 gets more complex. For degree 1, we can see a strange case with the left side perspective whose average value is under the rest of the cases. This problem is related to the contrast of the images belonging to the perspective. Another problem occurs with the rear perspective for degree 2 (lower average values and big standard deviations). In the case of degree 3, it only affects the frontal and rear perspectives (lowest average values with big standard deviations). This could be related to the resemblance between the front and rear parts, which causes the latter to be mistaken for the former. Apart from that, if we evaluate the total values, we can see that the stability of the baseline is very high with no big differences.

3.1. Objetive Evaluation

In the next step, we evaluate the different activation functions as an objective evaluation based on mathematical metrics. To this end, we provide the SSIM between the real image and the synthetic one (Table 3) and then the difference between the values of this table and the ones in Table 1, which contains the baseline (Table 4). In all these tables, each row corresponds to one of the direct trainings (depending on the activation function) and the columns to the destruction degree. The results in the cells contain the mean value of all the views with their standard deviation. We have provided the standard deviation in each destruction degree, as perspectives of the temples affect some cases. As this evaluation uses the entire image and not only the missing architectonical elements, the influence of visual artifacts is evaluated. To uncover information about these specific instances, we have a more thorough examination of the reconstructions during the objective assessment.

Table 3. Evaluations of four trainings with the whole image using SSIM.

Activation Function	Degree 1	Degree 2	Degree 3
ReLU	**95.09% ± 1.64**	**90.37% ± 1.74**	**85.76% ± 2.01**
Leaky ReLU	91.76% ± 1.61	88.06% ± 1.55	83.96% ± 1.70
Swish	90.70% ± 2.24	87.59% ± 2.73	83.82% ± 2.36
Mish	92.94% ± 2.25	88.88% ± 2.04	85.99% ± 3.77

Table 4. Differences between SSIM for the whole image and the baseline for the four trainings.

Activation Function	Degree 1	Degree 2	Degree 3
ReLU	−1.25% ± 0.63	2.17% ± 1.54	3.59% ± 1.62
Leaky ReLU	−4.59% ± 1.46	−0.14% ± 1.90	1.79% ± 1.87
Swish	−5.22% ± 1.24	−0.61% ± 0.77	1.65% ± 1.42
Mish	−3.41% ± 1.25	0.67% ± 1.11	2.56% ± 1.55

Considering the results in the table above, which shows the average values and standard deviations, we conclude the following. It should be noted that going from grade 1 to grade 3, we find an improvement in the baseline produced by the reconstructed image,

which ranges from around 7% to 10%. It is worth emphasizing that in the case of degree 3, the part to be reconstructed is larger, and therefore, there is a greater range of improvement in this image. Concerning degrees 1 and 2 of destruction, we observe that the activation function that incorporates a higher rate of artifacts is Swish, followed by Leaky ReLU, Mish, and ReLU, obtaining worse results for the baseline. Since the values provided do not show a large significant difference, we can deduce that the reconstructions are more complicated in addition to having large artifacts that differ from the real image. Looking at the standard deviations, the usage of different types of temples has no big influence. Considering Table 2 values, the performance of the activation functions worsens in this order: ReLU (bolded for being the best), Mish, Leaky ReLU, and Swish.

The values in Table 3 are subtracted from the reference values obtained by the similarity between images obtained in Table 2, resulting in the values in Table 4. Again, we show the average values and standard deviations.

Although grade 3 shows an improvement concerning the baseline, it has a larger improvement interval, while grade 1 is affected by the artifacts generated by showing a greater resemblance between the ruined image and the real image. The standard deviations are very low, which means that the different types of temples in the dataset do not pose a problem for the reconstruction. As in degree 1, all the values are negative. This means that all the generated images have worsened the baseline due to the creation of artifacts. Therefore, we can rank the performance of the activation functions in terms of artifact generation from best to worst for grade 1 as ReLU, Mish, Leaky ReLU, and Swish. For the rest of the grades, we find that these differences are not so evident, although ReLU is the one with the best performance index in all of them. This order coincides with that obtained in the previous table.

Due to the problem with artifacts in the generated images, we have concluded that evaluation of the entire image is not sufficient. To evaluate the reconstruction of the architectural elements, we have designed an experiment based on the application of a mask that hides the matching pixels so that we can evaluate only those reconstructed elements and, therefore, obtain a complementary. Using this assessment, we focus on measuring how well the missing architectural elements of a temple are reconstructed, which is the main goal of this work. To apply the mask, we find the pixels that have the same value in the image generated with the reconstructed temple and in the real image with the complete temple. All these pixels are assigned black and will not be taken into account when calculating the SSIM. It should be noted that not only the reconstructed area of the temples will be isolated but also other elements, such as shadows, stones, etc., that do not match the real image. In this case, it is not necessary to use a baseline.

So, the following table measures the SSIM between the reconstructed temples and the image with the complete temples but uses, in both cases, the mask trying to isolate the area, obtaining the missing elements to be reconstructed. In Table 5, the average results and standard deviations are compiled, and those corresponding to the activation function that performs the best are bolded.

Table 5. Masked evaluation of four trainings using SSIM.

Activation Function	Degree 1	Degree 2	Degree 3
ReLU	**67.56% ± 3.31**	**58.75% ± 4.56**	**53.60% ± 6.78**
Leaky ReLU	63.05% ± 4.90	55.33% ± 4.67	51.21% ± 7.28
Swish	62.35% ± 4.73	55.08% ± 5.02	52.26% ± 6.97
Mish	63.86% ± 5.17	56.18% ± 4.78	52.82% ± 6.91

Looking at the results of Table 4, we confirm that it is easier to reconstruct missing elements in the case of degree 1, and it gets more difficult while augmenting the area of missing elements. The standard deviations do not change much, which means that there are no big differences between the performances regarding the perspectives and the application to different types of temples. Even though the differences, in this case, are not

very high, this evaluation seems to confirm the order of performance for the activation functions ReLU, Mish, Leaky ReLU, and Swish, which becomes more evident in the case of degree 1. Again, the values of the metrics conclude that activation functions remain in the same order but still have low differences. To support these results, we made a subjective evaluation based on the perspective of an expert in the field. In Figure 6, we show an example of how the best activation function (ReLU) can reconstruct a temple. It shows the temple in ruins, how it was reconstructed using our method, and how it should be reconstructed perfectly.

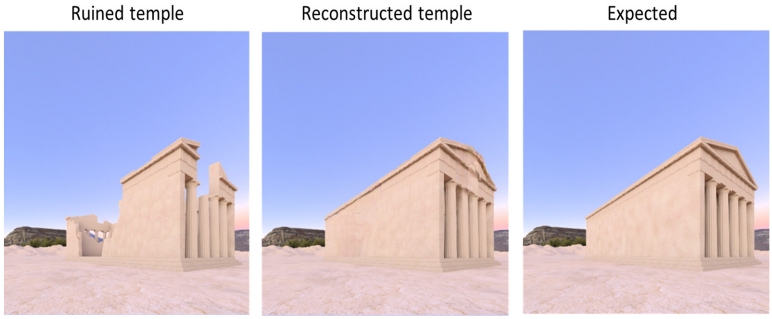

Figure 6. Performance of the method using ReLU as activation function.

3.2. Subjective Evaluation

Considering that the present work generates results based on visual perception and as many assumptions and possible conclusions have been obtained with the objective evaluation, we need an in-depth evaluation based on the experience of an expert in the field. Although all the previous results confirm that the performance of the activation functions is ReLU, Mish, Leaky ReLU, and Swish, we want a confirmation and an in-depth study based on a subjective evaluation. This subjective evaluation was approached by involving a professional who evaluated six important features of the images by adding a textual description for each of them. As can be seen in Figure 7, there are significant differences between the used activation functions.

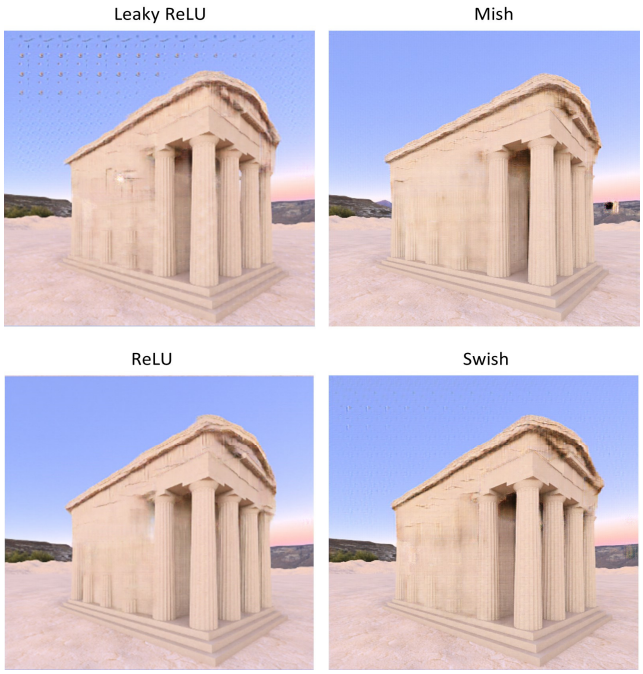

Figure 7. Performance of the different activation functions for the same example.

In this case, we have randomly selected 24 images for each activation function, which have been qualitatively evaluated by an architect considering the following relevant aspects that can be seen in the image: background, general shape, shadows, building interior, level of detail, and level of sharpness. Table 6 compiles this information for each characteristic.

Table 6. Quality evaluation for generated images.

Activation Function	Background	General Shape	Shadows	Building's Interior	Level of Detail	Sharpness
ReLU	It seems to perform well.	The general shape is well defined most of the time.	Shadows are coherent.	It is coherent, and the interior details can be appreciated.	It tries to reconstruct the finest details.	Best in sharpness.
Leaky ReLU	In some images, artifacts can be found.	The shapes are not accurate.	Shadows are coherent with the reconstruction.	It is coherent, but it lacks the finest details.	Low details. It has a big problem with the straight lines.	Low sharpness.
Swish	It seems to perform well.	Better definition of the contour. In some instances, a temple generates artifacts in an area in which there is no need for reconstruction.	Shadows are coherent but seem to be more diffuse.	It is coherent, and the interior details can be appreciated, but it generates artifacts along the temple.	It performs well in reconstructing particular elements.	Good sharpness but is worse than ReLU, as it has some problems creating transparencies between the background and the reconstructed parts of the temple.
Mish	It seems to perform well.	The shapes have good contours, but there are some alterations in the straight lines.	Shadows are coherent.	It is coherent, and the interior details can be appreciated.	Details can be intuited. It has problems with the straight lines.	Low sharpness but is better than Leaky ReLU.

If we evaluate the activation functions in depth, we can see the following common points. ReLU has a good definition in the interiors but has some problems discriminating between the temple and the background. The general shape is better defined with less blurring. It seems that it tries to represent the triglyphs with more coherence. It is the one with better sharpness. The shadows are coherent, and the background is well defined. It shows transparencies (a mix between the background and the part of the temple to be reconstructed) but to a lesser degree than the rest of the activation functions. As it has fewer cases of transparencies, it seems to make better reconstructions influencing the metrics as the similarities with the pixels of the ground truth are higher. It seems to make more accurate reconstructions instead of doing changes related to the overfitting (for example, it tries to reconstruct the corners while the other tends to round them).

In the case of Leaky ReLU, the interiors make sense, but details cannot be appreciated (they lack fine details). It has low sharpness, creating marked lines, but it is not well defined and has no precise forms. The shadows are suitable. In general, the reconstruction is not of good quality. It has a problem with the straight lines. With less sharpness, it makes worse reconstructions, so it is very sensitive to the light saturation of the image. It creates many transparencies that could be understood as artifacts. This problem with the transparencies affects the metrics having a higher error. It seems that the transparencies are proportional to the indecision of the model. More transparency indicates more indecision.

Regarding Swish performances, we can see that with the general shape, it performs similarly to ReLU. The shadows are coherent but seem a bit diffuse. The background is well reconstructed. It shows transparencies with less opacity compared to ReLU. It seems that the level of detail is better in particular architectural elements. It performs very defined reconstructions, but it has a lower sharpness due to the transparencies. In the interiors, we can see some artifacts. This problem with the artifacts affects the objective metrics, adding errors that prevent the evaluation of the reconstruction well and its coherence. The

shadows seem to be darker concerning the ground truth images. The brightness points are derived from the reconstructions and the transparencies caused by them. The corners are very badly reconstructed due to the lack of contrast between both walls generated by diffuse light. This activation function blurs more the reconstruction, while Leaky ReLU adds more noise.

Finally, Mish performs in a precise way with the general shape compared to Leaky ReLU but still being very diffuse. The details of the interiors are more defined, but there are no details in the triglyphs and lines of the columns. The shadows are coherent. The background is well represented. It shows much sensibility to the light saturation, which means that the facades with better contrast show a better reconstruction.

By comparing all the activation functions in total, the following statements are made. Among the four possibilities, Leaky ReLU is the worst. It shows many problems with straight lines, and it generates many artifacts. ReLU seems to obtain more definition, with less diffuse elements. It also shows better definition in small details, but Mish seems to have a better performance with the problem of element background. The reconstructions performed by ReLU and Swish are very similar in terms of sharpness and interior elements, but it seems that ReLU is a little bit better. This can be seen in a smaller number of transparencies and better-defined shadows. ReLU makes more detailed reconstructions, and Swish makes more defined ones. All the activation functions have problems with the light saturation (the perspectives with more contrast have a better reconstruction). Summarizing, the subjective evaluation considers the performance of the activation functions in the following order: ReLU, Swish, Mish, and Leaky ReLU.

This subjective evaluation suggests a general consideration: the predictive nature of generative AI could be a possible reason for the defects shown in generated images by the activation functions: transparencies, some lack of definition in the general shapes, low sharpness in the architectonical edges, blurriness, low level of detail, and the overlap between some elements of both images, due to contrast. The materiality and concrete situation of the built architectural elements have a univocal condition, which the predictive nature of artificial intelligence does not always seem to be able to restore faithfully, except by increasingly successful approximations.

4. Conclusions

The purpose of this work is to provide an accurate evaluation method for virtual image inpainting, specifically by assessing different activation functions in deep learning models used for restoring Greek temples. Using GANs, we focus on reconstructing missing architectural elements in images of temples with varying degrees of ruin. To ensure robust evaluation, we employed both objective and subjective methods. Objective evaluation used mathematical metrics on both whole images and reconstructed parts, while subjective evaluation involved an architect's expertise to validate these results.

Our findings indicate that the ReLU activation function outperforms others. Mish and Leaky ReLU did not perform well for this application, and Swish, although performing poorly in mathematical metrics, was close to ReLU in professional evaluations. This discrepancy suggests that current mathematical metrics do not fully capture human visual perception.

Several limitations were identified, such as overfitting in frontal and rear perspectives due to minimal differences between them and issues with lighting and material contrast affecting depth perception. This is particularly relevant when two columns with different degrees of ruin are viewed from the same perspective but are processed as a single column. Additionally, in some images, diffuse lighting significantly impacts the contrast, resulting in reduced visibility of walls forming a corner. These led to inaccuracies like rounded corners and transparency problems.

Future work will apply this method to more complex cases, such as Mudejar churches, which present greater architectural diversity. The composition and significant heterogeneity among the various architectural elements in these churches add substantial complexity to

the current problem of virtual restoration. Additionally, we aim to propose new evaluation metrics that better capture both the accuracy of architectural reconstruction and image quality, minimizing artifacts. Ultimately, we plan to validate our approach with real images, advancing toward reliable tools for cultural heritage restoration.

Author Contributions: Conceptualization, A.M.M., A.N. and G.F.; Formal analysis, A.M.M., A.N., G.I.S. and C.P.-C.; Funding acquisition, Á.J.G.-T.; Investigation, A.M.M. and A.N.; Methodology, A.M.M., A.N. and Á.J.G.-T.; Project administration, E.D.-M. and Á.J.G.-T.; Software, G.F.; Supervision, Á.J.G.-T.; Validation, E.D.-M., G.I.S. and C.P.-C.; Writing—original draft, A.N.; Writing—review and editing, A.M.M., E.D.-M., G.I.S., C.P.-C. and Á.J.G.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Project PID2021-126633NA-I00 supported by MICIU/AEI/10.13039/501100011033 and FEDER, UE.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stanley-Price, N. The Reconstruction of Ruins: Principles and Practice. In *Conservation: Principles, Dilemmas and Uncomfortable Truths*; Richmond, A., Bracker, A., Eds.; Elsevier: Amsterdam, The Netherlands, 2009.
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
3. Nogales, A.; Delgado-Martos, E.; Melchor, Á.; García-Tejedor, Á.J. ARQGAN: An evaluation of generative adversarial network approaches for automatic virtual inpainting restoration of Greek temples. *Expert Syst. Appl.* **2021**, *180*, 115092. [CrossRef]
4. Basu, A.; Paul, S.; Ghosh, S.; Das, S.; Chanda, B.; Bhagvati, C.; Snasel, V. Digital Restoration of Cultural Heritage with Data-Driven Computing: A Survey. *IEEE Access* **2023**, *11*, 53939–53977. [CrossRef]
5. Saricaoglu, T.; Saygi, G. Data-driven conservation actions of heritage places curated with HBIM. *Virtual Archaeol. Rev.* **2022**, *13*, 17–32. [CrossRef]
6. Andriasyan, M.; Moyano, J.; Nieto-Julián, J.E.; Antón, D. From Point Cloud Data to Building Information Modelling: An Automatic Parametric Workflow for Heritage. *Remote Sens.* **2020**, *12*, 1094. [CrossRef]
7. Tatsugami, F.; Higaki, T.; Nakamura, Y.; Yu, Z.; Zhou, J.; Lu, Y.; Fujioka, C.; Kitagawa, T.; Kihara, Y.; Iida, M.; et al. Deep learning—Based image restoration algorithm for coronary CT angiography. *Eur. Radiol.* **2019**, *29*, 5322–5329. [CrossRef] [PubMed]
8. Shao, J.; Zhang, J.; Huang, X.; Liang, R.; Barnard, K. Fiber bundle image restoration using deep learning. *Opt. Lett.* **2019**, *44*, 1080–1083. [CrossRef] [PubMed]
9. Choi, K.; Lim, J.S.; Kim, S. StatNet: Statistical image restoration for low-dose CT using deep learning. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 1137–1150. [CrossRef]
10. Jiao, J.; Tu, W.-C.; He, S.; Lau, R.W.H. Formresnet: Formatted residual learning for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 38–46.
11. SMukherjee, S.; Dittmer, S.; Shumaylov, Z.; Lunz, S.; Öktem, O.; Schönlieb, C.B. Data-Driven Convex Regularizers for Inverse Problems. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 13386–13390.
12. Mukherjee, S.; Carioni, M.; Öktem, O.; Schönlieb, C.B. End-to-end reconstruction meets data-driven regularization for inverse problems. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21413–21425.
13. Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; Li, H. Uformer: A general u-shaped transformer for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17683–17693.
14. Mishra, M.; Lourenço, P.B. Artificial intelligence-assisted visual inspection for cultural heritage: State-of-the-art review. *J. Cult. Herit.* **2024**, *66*, 536–550. [CrossRef]
15. Kniaz, V.V.; Remondino, F.; Knyaz, V.A. Generative Adversarial Networks for Single Photo 3D Reconstruction. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 403–408. [CrossRef]
16. Yu, C. Semi-supervised three-dimensional reconstruction framework with GAN. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 October 2019; pp. 4192–4198.
17. Theodorus, A. Restoration of Damaged Face Statues Using Deep Generative Inpainting Model. Master's Thesis, University of Twente, Enschede, The Netherlands, 2020.

18. Abed, M.H.; Al-Asfoor, M.; Hussain, Z.M. Architectural heritage images classification using deep learning with CNN. In Proceedings of the 2nd International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understandingco-Located with 16th Italian Research Conference on Digital Libraries (IRCDL 2020), Bari, Italy, 30–31 January 2020; pp. 1–12.
19. Reis, H.C.; Khoshelham, K. ReCRNet: A deep residual network for crack detection in historical buildings. *Arab. J. Geosci.* **2021**, *14*, 2112. [CrossRef]
20. Kwon, D.; Yu, J. Automatic damage detection of stone cultural property based on deep learning algorithm. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 639–643. [CrossRef]
21. Mishra, M.; Barman, T.; Ramana, G.V. Artificial intelligence-based visual inspection system for structural health monitoring of cultural heritage. *J. Civ. Struct. Health Monit.* **2024**, *14*, 103–120. [CrossRef]
22. Dini, S.F.; Wibowo, E.P.; Iqbal, M.; Bahar, Y.N.; Alfiandy, A. Applying Deep Learning and Convolutional Neural Network System to Identify Historic Buildings: The 'Little China' Building in Central Java, Indonesia. *ISVS E-J.* **2023**, *10*, 187–200.
23. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. [CrossRef]
24. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
25. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
26. Szandała, T. Review and comparison of commonly used activation functions for deep neural networks. In *Bio-Inspired Neurocomputing*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 203–224.
27. Gulcehre, C.; Moczulski, M.; Denil, M.; Bengio, Y. Noisy activation functions. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 3059–3068.
28. Hahnloser, R.H.R.; Sarpeshkar, R.; Mahowald, M.A.; Douglas, R.J.; Seung, H.S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **2000**, *405*, 947–951. [CrossRef] [PubMed]
29. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013; p. 3.
30. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
31. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681.
32. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 2546–2554.
33. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
34. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
35. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv* **2016**, arXiv:1607.08022.
36. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
37. Tieleman, T.; Hinton, G. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.