Virtual Restoration of Paintings Based on Deep Learning

Roman Sizyakin[a], Viacheslav Voronin[b], Aleksandra Pižurica[a],

[a]Department Telecommunications and Information Processing, TELIN-GAIM, Ghent University, Ghent, Belgium;
[b]Moscow State University of Technology "STANKIN", Moscow, Russia;

ABSTRACT

Over time, crack pattern (*craquelure*) inevitably develops in paintings as a sign of their ageing, sometimes accompanied by larger losses of paint (lacunas). In restoration treatments, cracks are typically not filled in, and virtual restoration is often the only option to "reverse" the ageing of paintings, simulating their original appearance. Moreover, virtual restoration can serve as an important supporting step in decision making during the physical restoration. In this research, we investigate the possibility of applying deep learning-based methods for virtual restoration. In particular, our crack detection method is based on a convolutional autoencoder (U-Net), and we employ a generative adversarial neural network (GAN) to virtually inpaint the detected cracks. We propose an original way of training the GAN model for painting restoration, which improves its practical performance. A series of experiments shows encouraging results in comparison with known methods, and indicates huge potential of deep learning for virtual painting restoratin.

Keywords: Virtual restoration of paintings, deep learning, crack detection, convolutional autoencoder (U-Net), generative adversarial network (GAN), machine learning, inpainting

1. INTRODUCTION

Virtual restoration is often the only plausible way to restore the original appearance of master paintings, which over time become inevitably affected by ageing and various kinds of deterioration, dominantly cracks and paint losses. In physical restoration treatments, painting cracks are typically left untouched unless at places where more severe painting losses are present. Although this conservation practice secures the authenticity of paintings, the ageing cracks still reduce the overall quality of visual perception and may hinder full appreciation of the original content laid down by the artist.

In this paper, we will focus on detecting and virtually inpainting cracks. Faithful automatic crack detection can provide invaluable support to art restorers, facilitating an objective insight into the current state of the painting and the evolution of deteriorations over time. Moreover, virtual inpainting serves in some cases as simulation to support the decisions that need to be made during the actual restoration process.

Crack detection methods often employ morphological filtering [1] due to its low computational complexity and high "Recall" metric. However, the detected crack maps typically contain many false-positives, and therefore morphological filtering is rarely used as an independent crack detection method but rather as a preprocessing step. The computational complexity of more advanced methods can be significantly reduced with such an effective preprocessing step that eliminates safely large areas where painting cracks are not present.

Most of the current crack detection methods are based on machine learning. A Bayesian approach [2, 3] forms feature vectors from the available image modalities and applies Bayesian Conditional Tensor Factorizations (BCTF) classifier [4]. The available set of imaging modalities often includes optical macrophotography, infrared macrophotography, infrared reflectography and X-ray images. Other modalities, like macro X-ray fluorescence or hyperspectral images are acquired in some cases as well, but these are still rather rare as require expensive equipment. The set of available imaging modalities is sometimes expanded artificially, creating virtual modalities, e.g., by applying various filters. The corresponding set of filters is typically optimized for each processed painting, which poses limitations in practice.

In general, the main problem with the existing multimodal crack detection approaches is their low resistance to intermodal shifts, which leads to an increase in false-positive responses. The problems arising from intermodal shifts can be alleviated by using patch-based convolutional neural networks (CNN) [5, 6]. By operating on small image patches, the convolutional neural network can effectively use both spatial and intermodal correlation to improve the crack detection accuracy and to improve the robustness to intermodal shifts. Most importantly, as with all deep learning methods, we now enjoy the advantage of not having to hand-engineer any filters, as the feature maps are now automatically synthesized inside the network during the training process. However, these methods yield excessive thickening of the actual crack

a) b) c)

Figure 1: Part of *Annunciation to virgin Mary* panel from the *Ghent Altarpiece*, a) Color image, b) Infrared image, c) X-Ray image

boundaries [7–9]. A possible solution to this problem is a combination of patch-based and vector-based techniques [10]. However, this approach does not always completely eliminate the problem of false crack thickening. Additionally, there is uncertainty with the choice of the patch size, which must be selected for each processed painting individually.

More precise classification (with pixel-level precision) can be achieved with segmentational convolutional autoencoders and their modifications [11–14]. Such neural network architectures receive a full image as input data, and output a segmentation map, with pixel-level precision. During the training process, the filters of such an autoencoder adapt to texture features that can be linked/combined into a local group, for example, by color or texture features. Those texture areas of the image that cannot be linked/combined into a local group are smoothed, and in the process of expansion (deconvolution) are ignored on the resulting segmentation map. The main disadvantage of such networks is a complex learning process that requires a large number of labelled training samples. Also, in some cases, this type of neural networks may require a significant amount of time for training or may not converge at all, due to poor-quality labelling of training data.

In the case of virtual restoration, the detection of cracks is only the first stage. The second stage is virtual restoration (inpainting) of the areas detected in the first stage. The simplest way to fill in the damaged areas is the usual polynomial interpolation of the boundary undamaged pixels. This group of methods includes the work in which the Navier-Stokes equations are used as an interpolating function [15]. This type of method can be useful if the fill rate is a priority requirement. However, if the area to fill is extensive, the absence of texturing of the filled area can be a significant disadvantage. Methods based on the search of self-similar patches on an undamaged area of the image cope with this problem more successfully. After that, the found patches are used to reconstruct the damaged area [16–18]. The most difficult cases for this group of methods are cases when the lost area includes a semantically important object in the image. Semantically important areas can include, for example: the wheels of a car, the windows of a house, or the mouth and eyes on the face. Such areas cannot be restored using this group of methods due to the fact that undamaged areas may not contain duplicates of such semantically important objects. Reconstructing variational autoencoders (VAE) [19] and adversarial neural networks (GAN) [20–22] can partially and in some cases completely solve this problem. The main advantage of generating neural networks is the ability to restore areas containing important semantic information, even if duplicates of such areas are partially or completely absent in undamaged areas of the image. The ability to restore such areas of the image is achieved during training (“memorization”), using training images. Subsequently, in the reconstruction mode, these neural networks use parts of these “memorized” training images to fill in the damaged areas. The main disadvantage of VAE generating methods is blurriness of the reconstructed area, while GANs suffer from an unstable training process.

In this paper, we investigate the possibility of virtual restoration of paintings using a segmenting convolutional neural network (U-Net) for detecting cracks, as well as a generating adversarial network (GAN) for removing detected cracks.

2. PROPOSED METHOD

Cracks in paintings can have extremely varying appearance: as dark or bright lines of various elongation and curvatures. Moreover their thickness can very from fine, hair-like lines to very thick ones flowing into regions of missing paint. The main difficulty for detecting cracks in master paintings is a complex background and similarity of crack patterns with various painted features (line-like structures like eye lashes, hair, parts of inscriptions), and various textures. Due to the impossibility to visually distinguish such objects from cracks, it is important to combine the optical macrophotographs

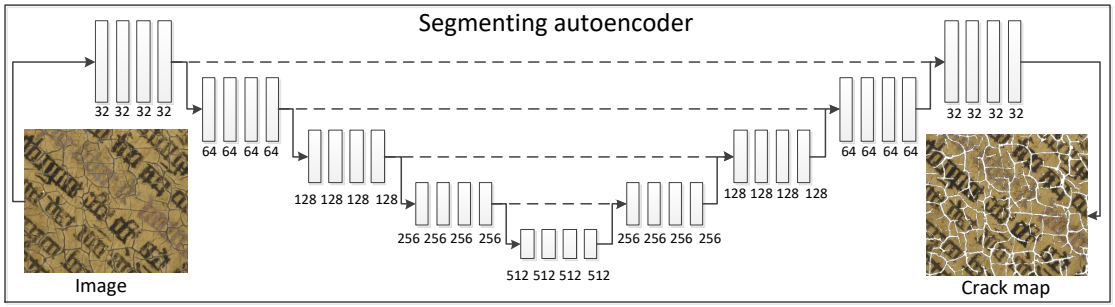


Figure 2: The proposed architecture of the segmenting autoencoder.

with other imaging modalities, such as infrared and X-ray images. In our work, we use multimodal acquisitions of the *Ghent Altarpiece* [23]*.

The challenge of detecting cracks is to construct a binary map on which the cracks are marked with value 1, and the undamaged areas are marked with 0. The input image $Y_{h,v}$ can be represented as:

$$Y_{h,v} = (1 - d_{h,v}) \cdot S_{h,v} + d_{h,v} \cdot c_{h,v} \tag{1}$$

where h, v are the spatial coordinates, $S_{h,v}$ is the undamaged content, $d_{h,v} \in \{0, 1\}$ a binary crack map of defects, and $c_{h,v}$ is the crack color.

2.1 Crack detection

In our work, to detect cracks, we use an extended version of the segmenting autoencoder (U-Net), which was originally proposed for the segmentation of medical images [12]. The network architecture is illustrated in Figure 2.

All convolutional autoencoders, both segmenting and generating, are based on the operation of N-dimensional convolution of input data and filters. Equation for this operation can be defined as:

$$x_{h,v}^{l,c} = f(\sum_{h} \sum_{v} \sum_{c} x_{h+m,v+n}^{l-1,c} \cdot k_{h,v}^{l,c} + b), \tag{2}$$

where $x_{h,v}^{l,c}$ is the feature map at layer l from modality c, $k_{h,v}^{l,c}$ is the corresponding convolution kernel, $x_{h+m,v+n}^{l-1,c}$ is the feature map from the previous layer, f is the activation function of the hidden layer, and b is a bias.

The training process consists in setting up the filters for convolution so that when the input data passes through all the layers of the neural network, the loss function is minimal. Here we use Sörensen–Dice coefficient [24, 25] for loss estimation, which shows the measure of the area of correctly marked segments and can be defined as:

$$Loss = \frac{2|x \cap d|}{x + d} \tag{3}$$

where x and d - is estimated and ground truth crack maps, respectively.

Next, the proposed architecture has following layers parameter: $C^0 5$, $C^1 32$, $C^2 32$, $C^3 32$, $C^4 32$, MP^5, $C^6 64$, $C^7 64$, $C^8 64$, $C^9 64$, MP^{10}, $C^{11} 128$, $C^{12} 128$, $C^{12} 128$, $C^{14} 128$, MP^{15}, $C^{16} 256$, $C^{17} 256$, $C^{18} 256$, $C^{19} 256$, MP^{20}, $C^{21} 512$, $C^{22} 512$, $C^{23} 512$, $C^{24} 512$, US^{25}, $C^{26} 256$, $C^{27} 256$, $C^{28} 256$, $C^{29} 256$, US^{30}, $C^{31} 128$, $C^{32} 128$, $C^{33} 128$, $C^{34} 128$, US^{35}, $C^{36} 64$, $C^{37} 64$, $C^{38} 64$, $C^{39} 64$, US^{40}, $C^{41} 32$, $C^{42} 32$, $C^{43} 32$, $C^{44} 32$, $C^{45}(sigm) 3$ where C^h - denotes a convolutional layer with index h, digit after C^h denotes a number of feature maps for current layer, MP^h - Max-pooling operation, US^h - Up-sampling operation and $(sigm)$ is denote logistic sigmoid activation function. All other layers use the exponentially

*Image Gallery: Closer to Van Eyck, Rediscovering the Ghent Altarpiece, http://closertovaneyck.kikirpa.be/

linear unit (ELU) [26] as activation function, which is a more efficient version of the activation function ReLU [27,28] and Leaky ReLU [29], and allows to achieve convergence of the neural network faster and higher accuracy, as well as exclude the process of batch normalization [30]. The equation can be written as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ a(e^x - 1) & \text{if } x \leq 0, \end{cases} \tag{4}$$

where $a > 0$ is a hyperparameter that controls the value at which the ELU saturates for negative inputs.

All convolutional layers have a spatial filter size of 3 × 3 pixels. For training we use Adam optimization [31], with a learning rate of 0.00005. The training process took in average 5000-10000 iteration, with a batch size of 3 pair images/masks with spatial size of 256×256 pixels.

2.2 Crack removal

For virtual crack inpainting, we employ a generative adversarial neural network (GAN) [32]. The main feature of this type of deep learning models is the use of the classification results from the convolutional neural network within the loss function of the generating autoencoder. As a result, the GAN model can include several networks. The most common case is the use of one autoencoder and two discriminators: local and global [†]. A local discriminator is needed to evaluate the quality of the directly restored area, and a global discriminator is needed to evaluate the semantic quality of the whole image, including the reconstructed and undamaged parts. This architecture allows to achieve a sharper generated image in comparison with a standard autoencoder.

The main difficulty in using this type of generating networks in practice comes from two main limitations: The first is the high probability of collapse of the training process, and the second is the lack of sufficient training data. So the first constraint is solved by independently adjusting the learning rates and the number of iterations for each of the neural networks that are part of the generating adversarial network, adding a stabilizing error (for example, MSE) that reduces the likelihood of an explosive growth of the loss function, or using more stable activation functions, etc. The second limitation can be solved if the image that needs to be restored is semantically related to images from the training set. Most well-known works follow this approach. However, in practice, this condition is difficult to achieve: the images may contain semantically different content. For example, one painting may include people with different types of clothing, and other some mythical creatures and/or nature and various objects. A partial solution may be to combine a large number of semantically different data sets, however, in this case, the training process of the adversarial network may take an exceptionally long period of time, which may completely exclude the possibility of their application in practice. Therefore, in our work, we propose to use the remaining undamaged parts of the painting as a training data set. This is possible due to the fact that usually panel paintings have extremely high resolution and undamaged areas between the cracks can be used to form a training set. For our case, using a patch with a spatial size of 24×24 pixels is optimal in terms of the total number of samples and their textural variability. Additionally, we reject the local discriminator and keep only the global one. The architecture of proposed the generating adversarial neural network that we use is illustrated in Figure 3.

The reconstructing network has the following architecture: $C^0 4$, $C^1 64$, $C^2 64$, $C^3 64$, $C^4 64$, MP^5, $C^6 128$, $C^7 128$, $C^8 128$, $C^9 128$, MP^{10}, $C^{11} 256$, $C^{12} 256$, $C^{12} 256$, $C^{14} 256$, US^{15}, $C^{16} 128$, $C^{17} 128$, $C^{18} 128$, $C^{19} 128$, US^{20}, $C^{21} 64$, $C^{22} 64$, $C^{23} 64$, $C^{24} 64$, $C^{25}(sigm) 3$ and global discriminator: $C^0 4$, $C^1 64$, $C^2 64$, $C^3 64$, MP^4, $C^5 128$, $C^6 128$, $C^7 128$, MP^8, $C^9 256$, $C^{10} 256$, $C^{11} 256$, $FC^{12}(sigm) 3$, where FC^h - denotes a fully connected layer with logistic sigmoid activation function. As input data for the layer $C^0 4$, a color image with a randomly deleted area is used together with a binary mask of the deleted area [‡]. All convolutional layers of the generating and discriminating networks use an exponentially linear unit (ELU) as an activation function.

The loss function for reconstructing network is determined according to the equation:

$$Loss_G = \lambda L_{abs} + L_{adv}, \tag{5}$$

[†] Usually standard convolutional neural networks are used as discriminators

[‡] To form a binary mask, a random section from the full map of cracks obtained at the crack detection stage is used

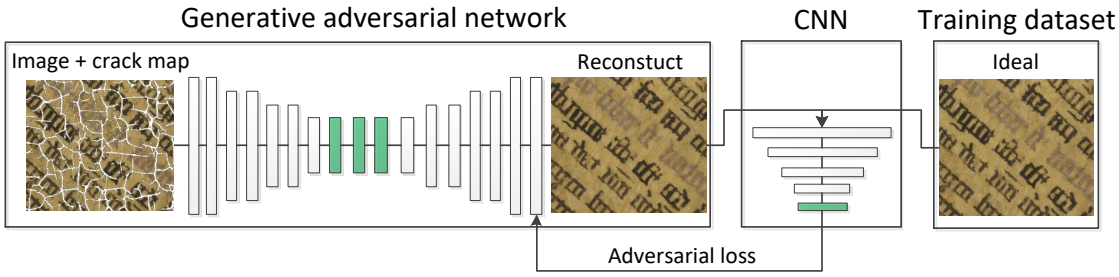
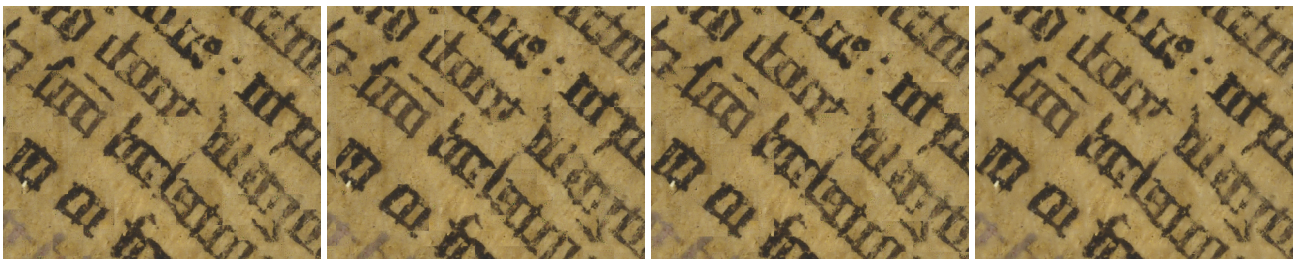


Figure 3: The proposed GAN-based model for virtual restoration.



a) b) c) d)

Figure 4: An example of the edge coherence problem in the independent processing of small patches of a large image. a,b,c) An example of removing cracks, provided that each subsequent processing begins with a shift of 0, 9 and 18 pixels, respectively, d) The result of combining all the images into one using the median filter.

$$L_{abs} = |x_{trn} - G(x_{def})|, \tag{6}$$

$$L_{adv} = \mathbb{E}[\log(1 - D(G(x_{def})))] \tag{7}$$

where x_{trn} - undamaged image for training, $G(x_{def})$ - reconstructed image, λ - coefficients of proportionality, which is used to align the loss order.

The L_{adv} error allows to achieve a higher sharpness of the reconstructed area and the L_{abs} loss allows to achieve a more stable learning process.

The task of the discriminator is to determine which of the images is the original and which is reconstructed. Loss function for discriminator are calculated according to the equation:

$$Loss_D = \mathbb{E}[\log(D(x_{trn}) + \log(1 - D(G(x_{def})))] \tag{8}$$

where x - source image, the size of which depends on what discriminator is used.

This configuration of loss functions leads to a adversary between two neural networks. Since the generator has a larger number of layers, one iteration of the training includes two steps of the generator and one step of the discriminator. Additionally we use Adam optimization with a different learning rate of 00008 and 00004, generator and discriminator, respectively. The training batch includes 200 samples with the size of 24×24 pixels.

Due to the fact that in our work we apply a generating adversarial network to small patches of the size of 24×24 independently, there is a problem of their incoherence at the edges, when combined into a full restored image. This problem is shown in Figure 4.

a) b)

Figure 5: Example of training dataset for crack detection:a) Source dataset for *Singing angels* panel, b) Extended dataset for *Singing angels* panel

To solve this problem, we process the full image several times using a small shift of 3 pixels for each iteration of the restoration. For example, if the first time the starting position for processing was the upper-left corner with the beginning of [0,0], then at the second iteration of processing, the starting position will be the value [3,3]. Example of such shift for 0,9 and 18 pixels illustrate in Figure 4(a,b,c) respectively.

Since we use the patch size of 24×24, we have 8 versions of the restored images in total. After that, the 8 versions of the reconstructed images are combined into one using the median filter. As a result, the final image contains only the pixels that received the highest probability among the 8 images, while the abnormal pixel values are rejected. The result of this operation shown in Figure 4(d)

3. EXPERIMENTAL RESULTS

To evaluate the effectiveness of crack detection and crack removal, we will use two digitized multimodal panel paintings from the *Ghent Altarpiece* [23]: *Virgin Annunciate* and *Singing Angels*. These images have extremely high resolution, so only small parts of them are shown here.

For comparison, we use the following well-known techniques for detecting cracks: MCNC method with improved crack boundary localization [6], Bayesian Conditional Tensor Factorization method (BCTF) [2], CNN-based method that was proposed for crack detection in roads [5] and a deep feature fusion network (DFFN) classifer from [33].

To test the effectiveness of the reconstructing adversarial neural network, we use two patch-based methods: exemplar based method (EBM) [16] and context-aware image inpainting using MRF [17].

To evaluate quantitative measures, we use the following metrics:

$$FA = \frac{FP}{AlPx - DfPx}, \quad FM = \frac{FN}{AlPx - UdPx} \tag{9}$$

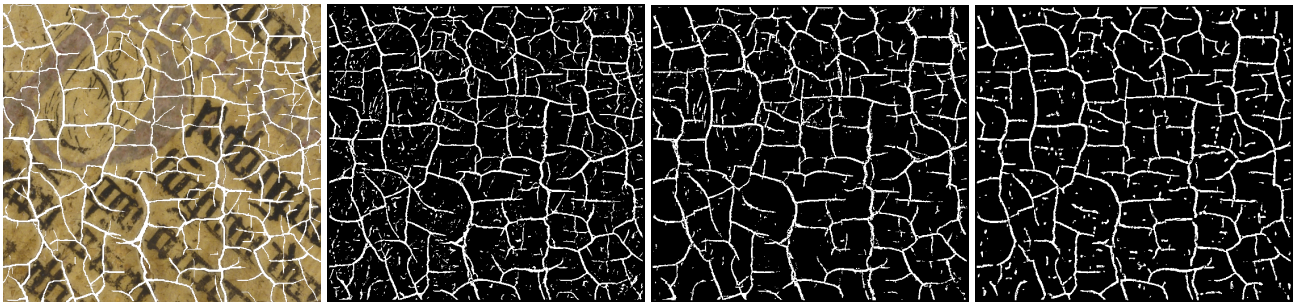
$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{10}$$

where FA - probability of false alarm, FM - probability of false missing pixels containing cracks, P - precision, R - recall, F_1 - F_1-measure, TP - true positive, FP - false positive, FN - false negative, $DfPx$ - total amount of pixels belonging to a crack, $UdPx$ - total amount of pixels not belonging to a crack, and $AlPx$ - total amount of pixels in the image.

3.1 Crack detection

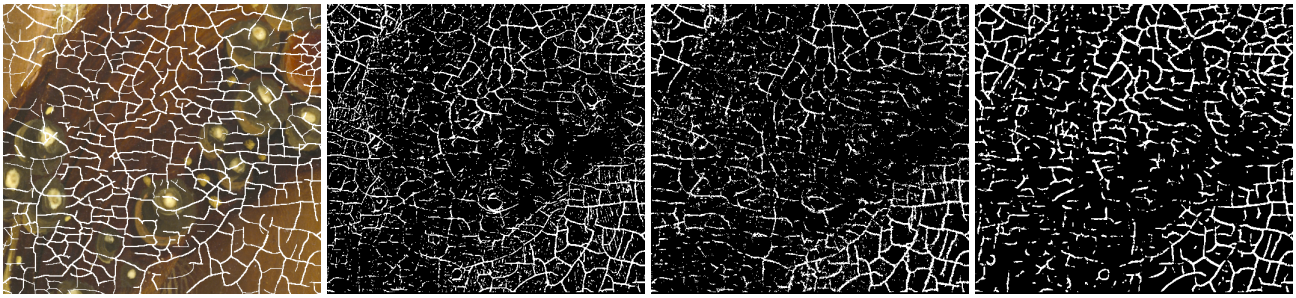
In the first part of the crack detection experiment, we use a training data set the same as in [Cornelis B. et al.] [2] and [Sizyakin R. et al.] [6]. An example of data with label from this set is illustrated in Figure 5(a) for *Singing angels* panel.

The main challenge for the proposed architecture (mUNET), based on the convolutional autoecoder (U-Net), is due to the fact that the data in this set is not fully marked, that is, some cracks were left without marking. The result of such an incomplete markup was that the proposed network architecture did not converge when cracks were detected on the *Singing Angels* panel. To solve this problem, we have marked up the training data as shown in Figure 5(b). There was no convergence problem for the *Virgin Annunciate* panel, since the data set was initially marked up almost completely for it.



a) b) c) d)

Figure 6: Example of crack detection: a) Part of *Virgin Annunciate* panel, b) Crack map of BCTF, c) Crack map of MCNC, d) Crack map of mUNET+C



a) b) c) d)

Figure 7: Example of crack detection: a) Part of *Singing Angels* panel, b) Crack map of BCTF, c) Crack map of MCNC, d) Crack map of mUNET+C

In addition to convergence problems, this segmenting network architecture has a high sensitivity to the accuracy and completeness of crack marking. That is, all the cracks from the training set should be completely painted up, which is very difficult to realize in practice. Failing to satisfy this condition results in imprecise detection of the thickness of the cracks: in the resulting map, all the detected cracks have approximately the same thickness. To solve this problem, we use the technique proposed in our earlier work [6], which allows us to reduce the number of false positives caused by excessive thickening of the crack boundaries. The essence of the technique is to apply a shift coefficient, which imposes a penalty on pixels that are beyond the actual boundaries of cracks on the binary map. In the tables 1 and 2, this approach, is denoted as (mUNET+C).

It is important to note that vector-based and patch-based machine learning methods do not suffer from such limitations, and it is enough to mark up only the pixels belonging to the center of the crack, without the need for full painting. These two limitations are fundamental in the application of segmenting autoencoders in practice. Since the process of complete marking of training data with high accuracy can take a significant period of time, which in some cases may call into question the reasonableness of using this method.

Table 1: Comparison of different methods for crack detection on a panel from the *Ghent Altarpiece*.

Annunciation virgin Mary panel					
Method	Recall	False alar.	False miss.	Precision	F_1-m.
CNN [5]	**0.8481**	0.0777	**0.1519**	0.5989	0.7020
DFFN [33]	0.7488	0.0422	0.2512	0.7081	0.7279
BCTF [2]	0.7896	0.0535	0.2104	0.6686	0.7241
MCN	0.8161	0.0540	0.1839	0.6741	0.7383
MCNC	0.7673	**0.0375**	0.2327	**0.7365**	**0.7516**
mUNET	0.8034	0.0702	0.1966	0.6101	0.6935
mUNET+C	0.7437	0.0417	0.2563	0.7090	0.7259

Table 2: Comparison of crack detection methods on a second selected panel from the *Ghent Altarpiece*, where * denotes an extended dataset and $+C$ the use of a technique for suppressing excessive thickening of the crack boundaries [6].

Singing angels panel					
Method	Recall	False alar.	False miss.	Precision	F_1-m.
CNN [5]	0.6119	0.0999	0.3881	0.4680	0.5304
DFFN [33]	0.6242	0.0966	0.3758	0.4814	0.5436
BCTF [2]	0.6150	0.0905	0.3850	0.4941	0.5479
MCN	0.6340	0.0894	0.3660	0.5048	0.5621
MCNC	0.6083	**0.0681**	0.3917	**0.5622**	**0.5843**
mUNET	-	-	-	-	-
mUNET*	**0.6441**	0.1104	**0.3559**	0.4559	0.5339
mUNET+C*	0.6140	0.0916	0.3860	0.4905	0.5454

The analysis of the obtained results confirmed the assumption that incomplete marking of training samples can significantly reduce the performance of convolutional segmentation networks and in some cases lead to non-convergence. Among the advantages, we can note a high resistance to false positives, as well as the absence of a problem of excessive thickening of the actual boundaries of the detected cracks, under the condition of very high accuracy of the training data markup, which is found in patch-based methods. Patch-based methods, although they lead to excessive thickening of the boundaries, have balanced characteristics between classification accuracy and the complexity of training data preparation. Among the advantages of vector-based methods, we can note the simplicity of preparing a training set, high accuracy of describing cracks and their boundaries, as well as the ability to quickly apply the trained model to other pictures. Among the disadvantages of vector-based methods, we should note the low resistance to intermodal shifts, low resistance to noise-like texture objects, as well as in most cases the need to use hand-crafted descriptors.

3.2 Crack removal

The results of recent studies show that generative adversarial networks can reconstruct large lost areas of images better than methods that search for self-similar patches. Their key advantage is the correct recovery of the semantic information of the lost area, which in general cannot be achieved by conventional patch-based inpainting methods. The larger the missing area is, the more pronounced is the advantage of the GAN-based methods. However, patch-based methods are quite successful in crack removal. Therefore, we evaluate the quality of the restoration with adversarial networks in areas where patch-based methods are also effective. In particular, we compare the proposed GAN-based restoration method with an exemplar based method (EBM) [16] and a context-aware method based on Markov Random Fields (MRF) [17], both of which proved to be successful in virtual restoration of paintings [3]. The results of the restoration are shown in Figure 8.

The obtained results confirm the effectiveness of all methods in relation to the problem of crack removal. Nevertheless, the EBM method has a certain number of contextually incorrectly filled areas, which makes the resulting image look noisy and rough. The result of the context-aware image inpainting using MRF method looks much better: the restored areas appear consistent and without visually disturbing artefacts. However, some of the restored areas are not filled in exactly, some of the borders of the drawn objects are torn, and in some places the filling occurs with incorrect patches. The most visually appealing is the result of the proposed generating adversarial neural network. Since there are no color-inconsistent fillings on it, the filled objects are contextually correct. The disadvantages include some smoothness of the filled areas. As a conclusion, it can be noted that the generative network only removes cracks, while patch-based methods remove cracks and also paint semantically unrelated objects that did not exist before.

4. CONCLUSION

In this work, we investigate the problem of virtual restoration of paintings. To detect cracks in this work, we use an extended version of the segmenting autoencoder (U-Net). The analysis of the obtained results shows that most of the actual cracks are correctly detected, while very few false positives are introduced. An important advantage with respect to earlier crack detection methods based on deep learning is that there is no excessive, false thickening of the boundaries of the detected

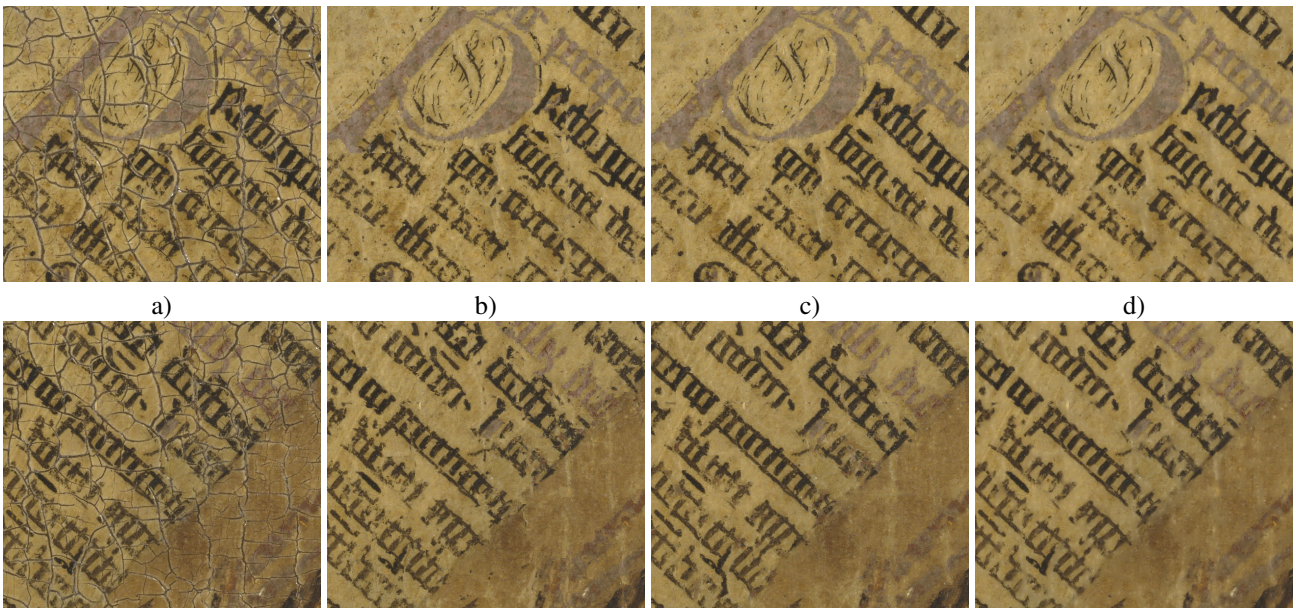


Figure 8: Example of removing detected cracks on two parts of the panel *Annunciation virgin Mary* a,e) Parts of the original painting, b,f) The results of EBM inpainting method, c,g) The results of context-aware MRF method, d,h) The proposed GAN technique.

cracks. This way, we overcome an important limitation of earlier deep learning based crack detection methods, including our previous work and most of the other current state-of-the-art methods in this field. However, this requires a training set with high accuracy of data markup. If such a high-quality training data set is not available it is still necessary to use techniques to refine the boundaries of the detected cracks. The second stage of our work was to investigate the possibility of using a generative adversarial neural network to remove the detected cracks. Due to the fact that a large set of training data is required for successful training of such a network, which is not always possible to have in practice, we propose to form such a data set using the remaining undamaged parts of the image. The obtained results confirm the high efficiency of the designed architecture of the GAN-based network and the proposed training method. The results of inpainting appear visually consistent and better than the results of patch-based methods that were earlier used to restore digitized paintings.

Acknowledgments

The Scientific Research was funded by Educational Organizations in 2020–2022 Project under Grant NoFSFS-2020-0031

REFERENCES

[1] Gupta, A., Khandelwal, V., Gupta, A., and Thammasat, M. C. S., "Image processing methods for the restoration of digitized paintings," *International Journal of Science and Technology* **13**(3), 66–72 (2008).

[2] Cornelis, B., Yang, Y., Vogelstein, J. T., Dooms, A., Daubechies, I., and Dunson, D. B., "Bayesian crack detection in ultra high resolution multimodal images of paintings," *IEEE, 18th International Conference on Digital Signal Processing* (2013).

[3] Pižurica, A., Platiša, L., Ružić, T., Cornelis, B., Dooms, A., Martens, M., Dubois, H., Devolder, B., Mey, M. D., and Daubechies, I., "Digital image processing of the Ghent altarpiece: supporting the painting's study and conservation treatment," *IEEE Signal Processing Magazine* **32**, 112–122 (2015).

[4] Yang, Y. and Dunson, D. B., "Bayesian conditional tensor factorizations for high-dimensional classification," *Journal of the American Statistical Association* **111**(512), 1–32 (2013).

[5] Lei, Z., Fan, Y., Yimin, D., and Ying, J. Z., "Road crack detection using deep convolutional neural network," *IEEE International Conference on Image Processing (ICIP)* , 3708–3712 (2016).

[6] Sizyakin, R., Cornelis, B., Meeus, L., Dubois, H., Martens, M., Voronin, V., and Pižurica, A., "Crack detection in paintings using convolutional neural networks," *IEEE Access* **8**, 74535–74552 (2020).

[7] Cha, Y.-J., Choi, W., and Büyüköztürk, O., "Deep learning-based crack damage detection using convolutional neural networks," *Computer - Aided Civil and Infrastructure Engineering* **32**, 361–378 (2017).

[8] Li, Y., Li, H., and Wang, H., "Pixel-wise crack detection using deep local pattern predictor for robot application," *MDPI and ACS Style* (2018).

[9] Kim, B. and Cho, S., "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique," *MDPI and ACS Style* (2018).

[10] Sizyakin, R., Cornelis, B., Meeus, L., V. V., and Pižurica, A., "A two-stream neural network architecture for the detection and analysis of cracks in panel paintings," *ISO&P, Optic, Photonics and Digital Technologies for Imaging Applications VI* , 1–9 (2020).

[11] Badrinarayanan, V., Kendall, A., and Cipolla, R., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(12), 2481–2495 (2017).

[12] Ronneberger, O., Fischer, P., and Brox, T., "U-net: convolutional networks for biomedical image segmentation," *Springer,Medical Image Computing and Computer-Assisted Intervention, MICCAI* **9351** (2015).

[13] Sizyakin, R., Voronin, V., Gapon, N., and Pižurica, A., "A deep learning approach to crack detection on road surfaces," *ISO&P, Artificial Intelligence and Machine Learning in Defense Applications II* , 1–7 (2020).

[14] Meeus, L., Huang, S., Zizakic, N., Xie, X., Devolder, B., Dubois, H., Martens, M., and Pizurica, A., "Assisting classical paintings restoration: efficient paint loss detection and descriptor-based inpainting using shared pretraining," *SPIE, Optics, Photonics and Digital Technologies for Imaging Applications VI* , 1–13 (2020).

[15] Bertalmío, M., Bertozzi, A., and Sapiro, G., "Navier-stokes, fluid dynamics, and image and video inpainting," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* **1**, I–I (2001).

[16] Criminisi, A., Perez, P., and Toyama, K., "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing* , 1200–1212 (2004).

[17] Ružic, T. and Pižurica, A., "Context-aware patch-based image inpainting using markov random field modeling," *IEEE Transactions on Image Processing* **24**(1), 444–456 (2015).

[18] Voronin, V., Marchuk, V., Sizyakin, R., Gapon, N., Pismenskova, M., and Tokareva, S., "Automatic image cracks detection and removal on mobile devices," *Mobile Multimedia/Image Processing, Security, and Applications* (2016).

[19] Ham, C., Raj, A., Cartillier, V., and Essa, I., "Variational image inpainting," *Third workshop on Bayesian Deep Learning (NIPS)* , 1–6 (2018).

[20] Iizuka, S., Simo-Serra, E., and Ishikawa, H., "Globally and locally consistent image completion," *ACM Transactions on Graphics* **36**(4) (2017).

[21] Sizyakin, R., Voronin, V., Gapon, N., Zelensky, A., and Pižurica, A., "A deep learning-based approach for defect detection and removing on archival photos," *Electronic Imaging, Society for Imaging Science and Technology,* **10**, 1–7 (2020).

[22] Jiahui, Y., Zhe, L., Jimei, Y., Xiaohui, S., Xin, L., and Thomas, S. H., "Generative image inpainting with contextual attention," *CoRR* (2018).

[23] Dataset., "Image gallery: Closer to Van Eyck, rediscovering the Ghent Altarpiece", *http://closertovaneyck.kikirpa.be/* (2011).

[24] Sörensen T., A., "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Kongelige Danske Videnskabernes Selskab* **5**(4), 1–34 (1948).

[25] Dice Lee, R., "Measures of the amount of ecologic association between species," *Ecology* **26**(3), 297–302 (1945).

[26] Clevert, D., Unterthiner, T., and Hochreiter, S., "Fast and accurate deep network learning by exponential linear units (ELUs)," *ICLR: International Conference on Learning Representations* (2016).

[27] Krizhevsky, A., Sutskever, I., and Hinton, G., "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems* , 1097–1105 (2012).

[28] Glorot, X., Bordes, A., and Bengio, Y., "Deep sparse rectifier neural networks," *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR* **15**, 315–323 (2011).

[29] Maas, A. L., "Rectifier nonlinearities improve neural network acoustic models," *International Conference on Machine Learning (ICML)* (2013).

[30] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on International Conference on Machine Learning* , 448–456 (2015).

[31] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *ICLR: International Conference on Learning Representations* (2015).

[32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in Neural Information Processing Systems* , 2672–2680 (2014).

[33] Song, W., Li, S., Fang, L., and Lu, T., "Hyperspectral image classification with deep feature fusion network," *IEEE Transactions on Geoscience and Remote Sensing* **56**, 3173–3184 (2018).