Assisting classical paintings restoration: Efficient paint loss detection and descriptor-based inpainting using shared pretraining

Laurens Meeus[a], Shaoguang Huang[a], Nina Žižakić[a], Xianghui Xie[b], Bart Devolder[c], Hélène Dubois[d], Maximiliaan Martens[e], and Aleksandra Pižurica[a]

[a]Group for Artificial Intelligence and Sparse Modelling, Ghent University, Belgium
[b]Faculty of Engineering Technology, KU Leuven, Belgium
[c]Princeton University Art Museum, NJ USA
[d]Royal Institute for Cultural Heritage, KIK/IRPA Brussels, Belgium
[e]Department of Art History, Musicology and Theatre Studies, Ghent University, Belgium

Keywords: Art investigation, multi-modal data, semantic segmentation, pretraining, transfer learning.

ABSTRACT

In the restoration process of classical paintings, one of the tasks is to map paint loss for documentation and analysing purposes. Because this is such a sizable and tedious job automatic techniques are highly on demand. The currently available tools allow only rough mapping of the paint loss areas while still requiring considerable manual work. We develop here a learning method for paint loss detection that makes use of multimodal image acquisitions and we apply it within the current restoration of the *Ghent Altarpiece*.

Our neural network architecture is inspired by a multiscale convolutional neural network known as U-Net. In our proposed model, the downsampling of the pooling layers is omitted to enforce translation invariance and the convolutional layers are replaced with dilated convolutions. The dilated convolutions lead to denser computations and improved classification accuracy. Moreover, the proposed method is designed such to make use of multimodal data, which are nowadays routinely acquired during the restoration of master paintings, and which allow more accurate detection of features of interest, including paint losses.

Our focus is on developing a robust approach with minimal user-interference. Adequate transfer learning is here crucial in order to extend the applicability of pre-trained models to the paintings that were not included in the training set, with only modest additional re-training. We introduce a pre-training strategy based on a multimodal, convolutional autoencoder and we fine-tune the model when applying it to other paintings. We evaluate the results by comparing the detected paint loss maps to manual expert annotations and also by running virtual inpainting based on the detected paint losses and comparing the virtually inpainted results with the actual physical restorations. The results indicate clearly the efficacy of the proposed method and its potential to assist in the art conservation and restoration processes.

1. INTRODUCTION

Digital painting analysis is a growing field of research, rapidly gaining interest from the image processing and machine learning communities. Typical tasks include artist identification,[1] forgery detection,[2,3] crack detection,[4] paint loss detection,[5] and virtual inpainting.[6,7]

Paint loss is often the result of flaking and drying of paint due to aging, although rough handling can also cause it. When restoring a painting, regions of paint loss are inpainted by the conservators. For this, it is needed to know and document the exact areas of paint loss. Currently, this documentation involves a lot of manual work since available software can only give a coarse estimation of the paint loss areas. The documentation process is

Further author information: (Send correspondence to L.M.)
L.M.: E-mail: laurens.meeus@ugent.be

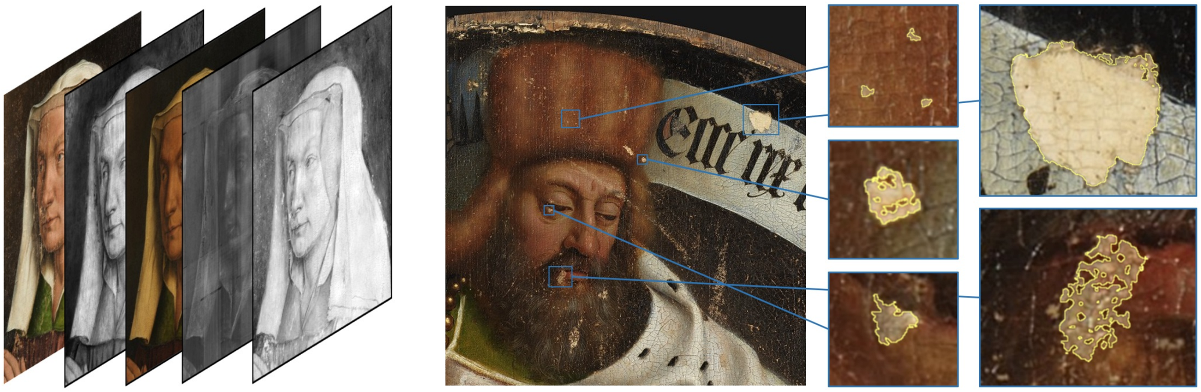


Figure 1: Multi modal image acquisitions and examples of paint loss in the macrophotgraphy during treatment. Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

therefore rather tedious, and would benefit from an improved automated mapping of paint loss regions. Restorers usually detect these regions visually, by inspecting the painting itself and sometimes also by looking in parallel at digital acquisitions in other modalities, such as infrared and X-ray images. These additional imaging modalities provide supplementary information and enable a better assessment of the features of interest, including lacunas and larger paint losses.

Research on automatic paint loss detection is limited, especially when using the multimodal images as input. Huang et al.[5] showed promising results using sparse representation classification (SRC), surpassing common machine learning approaches like linear regression classification and support vector machines in this task. However, this method is computationally intensive, which makes processing of larger images very challenging, especially in cases where user feedback is desirable. In our previous work[8] we proposed a novel neural network architecture for image segmentation that was optimized to employ the spatial context and multimodal data. Since this is a supervised deep learning approach, it requires a large amount of annotated data samples, which may be impractical. In addition, the annotations may not always be reliable, depending on the patience of the user, which further suggests the need to limit the necessary amount of labeling by the user.

In this paper, we propose a semi-supervised deep learning approach for paint loss detection, which does not rely heavily on large amounts of labels. We make use of transfer learning such that the pre-trained model can be reliably applied to new images with relatively few extra annotations.

As a case study, we use the panels of the *Ghent Altarpiece*,[9] a monumental polyptych made by the brothers *van Eyck* in the 15th century. The application of our paint loss detection is demonstrated on images taken during the conservation/restoration of this masterpiece. We evaluate the results in two ways: by comparing the detected paint loss maps to manual expert annotations and by running virtual inpainting based on the detected paint losses. The results of automatic virtual inpainting of the detected paint losses show close resemblance with the actual physical restoration, which indicates that paint losses have been accurately detected. Quantitative evaluation is conducted by comparing the detected paint loss areas with manual expert detection. Both quantitative and qualitative results show clearly the excellent performance of the proposed approach in comparison with the existing method and indicate its potential to assist in the actual conservation/restoration treatment of paintings.

2. PRELIMINARIES

In this Section, we review the main concepts behind the U-Net[10] architecture and convolutional autoencoders[11] that will serve as a basis for developing our model. In particular, we shall describe our previously reported translation invariant version of the U-Net architecture TI-U-Net[8] and the related concept of dilated convolutions.[12] In this paper, convolutional autoencoders are used to pretrain our architecture with the goal of limiting the necessary user interference. We shall also explain the basic concepts behind the chosen virtual inpainting approach that will be employed later on to evaluate the results of our paint loss detection method by simulating the actual painting restoration.

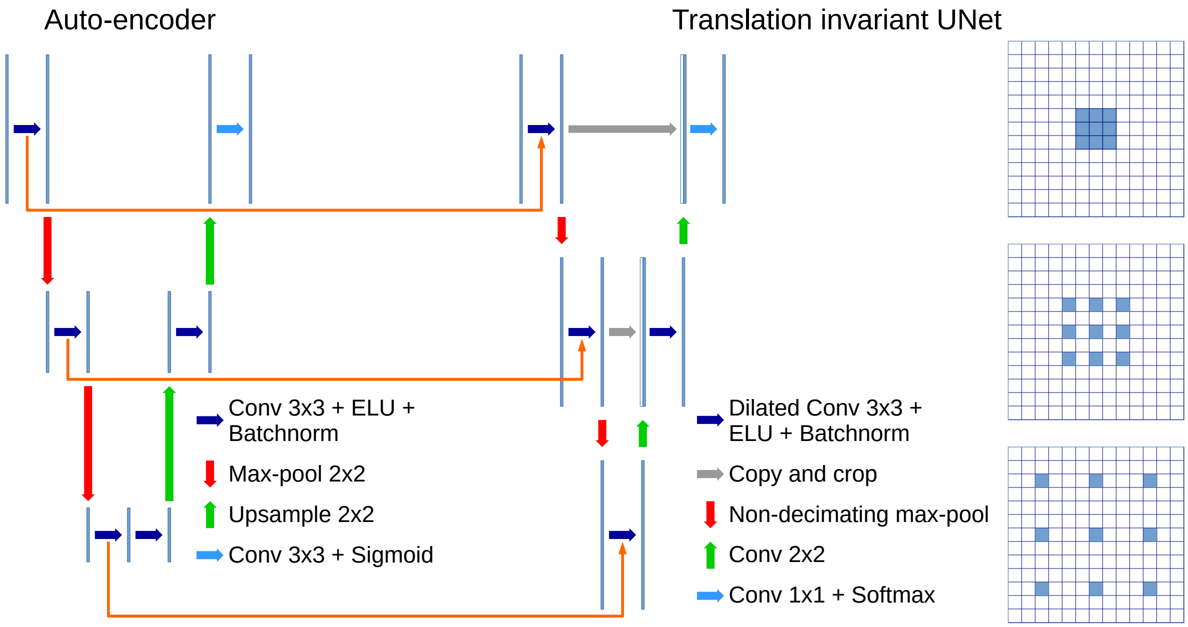


Figure 2: A schematic overview of the proposed model, showing how the weights of the encoder are transferred per layer from the autoencoder to the TI-UNet with dilated convolutions. Left: the autoencoder where the amount of filters per layer (k) is kept constant over the whole network. Right: the translation invariant UNet (TI-UNet), shown with same amount of filters per layer (k) and with the reach of the kernels of the dilated convolutions illustrated on the far right. The orange arrows indicate the transfer between the corresponding layers.

2.1 Translation invariant U-Net

A state-of-the-art deep learning model for image segmentation U-Net,[10] is a convolutional neural network whose architecture is similar to autoencoders[13] with additional layers called skip-connects. Skip-connects add shortcuts between the encoder and decoder such as to combine lower-level, higher-resolution information of the encoder with higher-level, lower-resolution information of the decoder. This way a big receptive field that captures more global contextual information is combined with local spatial information.

Our baseline architecture,[8] illustrated on the right-hand side of figure 2, efficiently uses spatial information to avoid overtraining on the limited amount of annotations. Inspired by U-Net,[10] our model consists of an encoder (left), a decoder (right) and skip-connects (center). Unlike U-Net, there are no decimating pooling layers and instead dilated convolutional layers[12] are used to maintain a big receptive field. In a dilated convolutional layer the weights of the kernel K (width $2M+1$) are spaced out by a factor k, the dilation rate:

$$z[u] = (x *_k K)[u] = \sum_{m=-M}^{M} x[u - k \cdot m]K[m]. \tag{1}$$

This way the produced output z at pixel u maintains the same resolution as the input x while the kernel virtually operates on a lower resolution grid, having a much bigger receptive field without increase in kernel size.

The encoder consists of 3×3 dilated convolutions layers, alternated every two layers with a max-pooling layer with stride set to one. This way the pooling layer solely acts as a low-pass filter. Additionally the operating width of each pooling layer increases according to $w = 2^{(k-1)} + 1$ to reduce more high frequency information according to the change in dilation rate.

The dilation rate is doubled after each pooling layer, starting from the value 1 for the first layer. The amount of filters in each layer is kept constant. With respect to U-Net, there is no need to double the amount of filters f

after each pooling layer to make up for the loss of spatial information. Without taking into account the shrinking after each convolutional layer, the shape of each layer is now approximately $w \times w \times f$ versus $w/2^k \times w/2^k \times f \cdot 2^k$ for the respective U-Net layer, where k is the amount of previous pooling layers. This means the produced feature maps are denser in spatial information, and when training the network, the updates of the weights are averaged out over more pixels, improving stability. In summary, with every extra pooling and increase in dilation rate, the receptive field increases exponentially while the amount of trainable parameters only increases linearly.

The decoder mostly mirrors the encoder except the pooling layers are replaced with a 2×2 dilated convolutional layers. Instead of doubling the dilation rate, it is now again halved, starting with the 2×2 convolutional layers. As the resolution in each layer is the same, there is no need to upscale the previous feature map, however it is still useful to gradually reduce the dilation rate again to aggregate the information of neighbouring pixels.[14] The output of the 2×2 convolutional layers is concatenated with the input of the corresponding pooling layer of the encoder with the idea of combining high-level, low-pass with low-level high-pass features respectively. The last layer is a 1×1 convolutional layer with the number of filters corresponding to the number of classes. For our detection problem, the two output maps corresponds to the paint loss and background classes. Each layer uses an ELU[15] activation function, except for the last layer which uses softmax (σ):

$$\hat{y} = \sigma(\mathbf{z})_j = \frac{e^{z_j}}{e^{z_1} + e^{z_2}}. \tag{2}$$

This is to produce a normalised prediction $\hat{y}$ of the last output vector $\mathbf{z}$, corresponding to the likelihood that a pixel belongs to the classes paint loss and background.

Given that the whole architecture only uses dilated convolution layers and non-decimating pooling, an interesting property arises. Translation invariance means that translating an input x produces the same, although translated (T), version of the output z.

$$x(u) \mapsto z(u) \iff x(u + \mathbf{T}) \mapsto z(u + \mathbf{T}). \tag{3}$$

Unlike architectures with subsampling layers, each layer of this architecture is explicitly translation invariant, such that the whole system is translation invariant, which is a desired property of semantic segmentation models. Because no spatial information is lost after a pooling layer, translation invariance does not have to be learned by the model and the amount of filters (k) is kept constant in each layer.

2.2 Convolutional autoencoders

Autoencoders are neural networks consisting of an encoder e and decoder d part which are trained to reproduce the input x from a lower-dimensional latent representation $e(x)$. Formally,

$$\hat{x} = (d \circ e)(x) \approx x. \tag{4}$$

The goal of the encoder is to learn efficient data codings, typically compressing the input data to a lower dimensional manifold. The initial use was for dimensionality reduction techniques and learning features.[16]

With the appearance of convolutional neural networks, convolutional autoencoders made their debut.[11] As convolutional layers do not reduce the spatial dimensionality, pooling layers are used to reduce the spatial resolution. This loss in spatial information forces the different filters to capture the relevant spatial features into the different feature maps. What is crucial is that the total dimensionality gets reduced in the encoder in order to force the autoencoder to learn this lower dimensional representation. A schematic of a convolutional autoencoder is shown on the left side of figure 2, with the width and height of each bar depicting the amount of feature maps and resolution per layer respectively.

Working with multimodal data, the different image modalities are rescaled and registered with respect to each other in order to stack them as a 3D datacube, with the different channels representing the components from different imaging modalities. Training the autoencoder on the these modalities together leads to an encoding that combines both the intra- en inter-modal information which improves the reconstruction performance.[17] For that reason, we argue that this encoding can be extra helpful when used as initialisation for the paint loss detection model, which is what we propose in this paper.

2.3 Virtual restoration

In virtual inpainting, marked regions of an image are filled in a visually plausible way. Using the paint loss detection as mask to be virtually inpainted, allows us to simulate the whole restoration process done by the restorers, thus assisting them in the decision-making process. Conservators usually do not fill in craquelure (i.e. paint cracks) in the actual inpainting process, unless in cases where cracks show as larger areas of missing paint and can be characterized as paint loss. Thus it is relevant for the conservator that craquelure is not detected together with paint loss and the realistic digital inpainting needs to leave the craquelure untouched. We shall employ patch-based image inpainting[18] to virtually restore the images starting from the detected paint loss maps.

3. METHODS

Our primary contribution is a pretraining and weights-transfer scheme for the TI-U-net that allows the paint loss detection on the multimodal images to be performed with a minimal amount of labels.

For the detection of paint loss, annotations are typically very scarce since annotating manually high-resolution images at a pixel level is very tedious. The amount of annotations and the precision with which they are made depend largely on the patience of the user. Thus, to improve the usability of automatic paint loss detection, the performance has to be maximised with limited user-interference. In order to improve the quality of the prediction with limited data, we introduced earlier a translation invariant UNet (TI-UNet).[8] This TI-UNet combines dilated convolutions with the typical U-Net without any down- and upsampling layers. As there is no reduction in spatial resolution, the information through the layers becomes more dense, which improved the training capacity when working with restricted amount of data. To further improve the performance in the case of limited annotations, we apply an unsupervised pretraining step and extend this model such that it can continuously improve the classification accuracy by combining new annotations with preliminary prediction results.

3.1 Transfer learning: from autoencoder to dilated convolutional encoder

In order to be able to apply transfer learning on the encoder, the respective layers have to be compatible i.e. the kernels need to have the same shape. As visualised in figure 2, the encoder of the autoencoder has the same number of convolutional and (decimating) max-pooling layers. As the resolution is reduced after each pooling layer, the convolutional layers are not dilated, but still use the ELU activation function and are followed by a batch normalisation layer. The decoder is the mirrored version of the encoder, with the pooling layers replaced by upsampling layers to increase the resolution again to the original one. As the amount of filters per layer (k) is constant for the TI-UNet, the amount of filters is also constant for the autoencoder. The last layer of the autoencoder produces the same number of feature maps as the input and the activation is set to a sigmoid ($S(z) = 1/(1 + e^{-z})$) as the input is normalised to be in $[0, 1]$. It is optimized with respect to a binary cross entropy loss for each featuremap:

$$L(x, \hat{x}) = -\left(x \cdot \log(\hat{x}) + (1 - x) \cdot \log(1 - \hat{x})\right). \tag{5}$$

One of the main prerequisites to successfully training an autoencoder is to have a latent space that is smaller than the input space in order to force the model to learn a lower-dimensional representation of the input. If dilated convolutional layers would be used for the autoencoder, once the amount of filters (k) equals 9, the dimensionality does not have to be reduced and thus it would be possible that the identity transform is learned. This would lead to no new interesting information in the encoding. To avoid this, the autoencoder needs to use regular convolutional layers with downsampling.

We develop a method to make the encoder compatible for transfer learning to our TI-U-Net, motivated by the observations of Tschopp et al[19] who showed how convolutional neural networks with pooling layers can be converted to networks with a constant resolution by applying dilated convolutional layers. These networks have been shown to produce the identical outputs as their original. The intuitive explanation is that the dilated convolution will mimic the behaviour as if working on a lower resolution grid, matching the same operational resolution and receptive field of the respective regular convolutional layer of the autoencoder. As such, the weights can be safely transferred between the encoders without producing undesired outputs. Our transfer of the kernel parameters of the encoder is illustrated by the orange arrows in figure 2, this includes the weights of

Table 1: Summary of the average performance of the different methods. The IoU is used as measure of performance. The values are averaged over the the different cross-validation sets, after the model converged and only for those models with high enough number of kernels in order to be able to have the maximum performance. T and F stand for the encoder being trainable or fixed respectively. F-T is for the encoder first being trained with a fixed encoder, followed by a finetuning where the encoder is also trainable.

Methods		TI-UNet	T TI-UNet	F TI-UNet	F-T TI-UNet
IoU per fold	1	0.243	0.260	0.270	**0.278**
	2	0.137	**0.185**	0.124	0.174
	3	0.269	0.316	0.254	**0.337**
	4	0.080	**0.107**	0.081	0.089
	5	0.263	**0.271**	0.197	0.265
	6	0.126	**0.140**	0.113	0.138
Average IoU		0.186	**0.213**	0.173	**0.213**
Average accuracy		0.836	0.837	0.832	**0.838**

the batch normalisation layers. The first convolutional layer of both encoders is identical and is thus transferring the weights is trivial. After the first pooling layer, the convolutional layer of the encoder operates on half the original resolution. As such the dilation rate of the respective layer of TI-U-Net is set to 2 and thus this layer also operates, in essence, on a grid of half the resolution. After more pooling layers, the same reasoning can be made. As the resolution again halves, the dilation rate of the respective layer doubles and as such they stay compatible and their weights can transferred while still producing valid outputs.

When applying the same model to other panels, that were not included in the training, there is no need to start the training from scratch. Instead, the preliminary detection is shown to the user, which then can provide limited extra annotations for those regions he or she does not agree with. As we start from a better initialisation and these extra annotations provide a lot of information, the model can be fine tuned with limited user-interference, speeding up training time. This process can be repeated until the desired performance is reached.

4. RESULTS & DISCUSSION

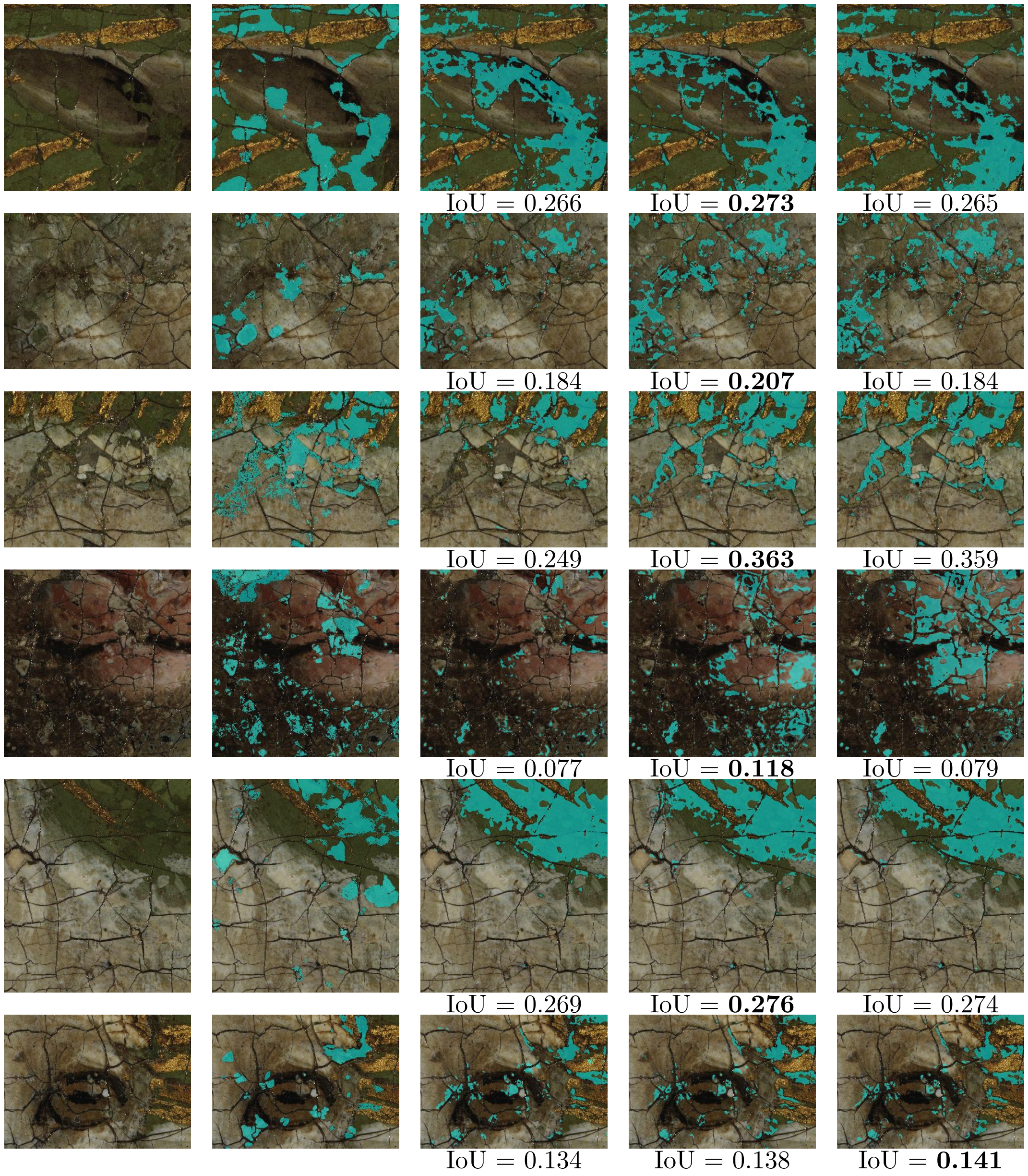
As a case study, we use image acquisitions of the panels of the *Ghent Altarpiece*, taken during the current restoration treatment. We evaluate the three variants of the proposed method described in section 3: the TI-UNet without any form of pretraining, transfer learning from an autoencoder and transfer learning a pretrained model to other paintings.

4.1 Data set

We evaluate the proposed method in the context of the ongoing conservation of the Ghent Altarpiece by the Royal Institute of cultural Heritage (KIK-IRPA).[9] This monumental polyptych, depicting the *Adoration of the Mystic Lamb*, was painted by the brothers *Hubert* and *Jan van Eyck* and finished in 1432. We use the imaging modalities documented on the website *http://closertovaneyck.kikirpa.be*[20] as well as additional images that were acquired during the restoration treatment. We use the following modalities: Digital colour images (RGB) taken during treatment which are also used to annotate paint loss, and four imaging modalities acquired before treatment: visible, infrared (IR), infrared reflectography (IRR) and X-ray. For the different modalities, the painting was captured in sections with fixed lightning conditions, with resolutions in the range of 74 to 480 pixels/cm. For further processing these are rescaled and registered with respect to the visible modality during treatment, leading to a datacube with 9 components.

4.2 Paint loss detection without pretraining

To analyse the effect of pretraining the encoder, we report the results on a close-up of the face of the *Mystic Lamb* of the central panel, shown in figure 5a. To quantitatively evaluate the performance, six smaller regions are densely annotated and are used in a six-fold cross-validation scheme: one patch is kept as a test set, while

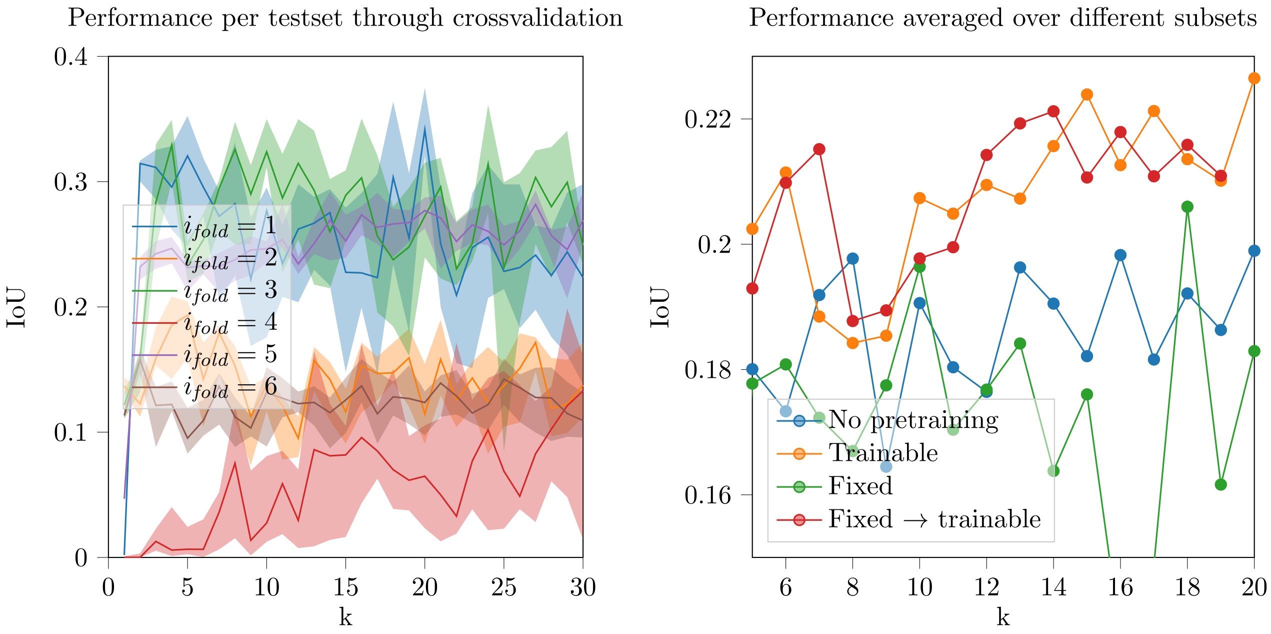


(a) During Treatment (b) Ground truth (c) TI-UNet (d) T TI-UNet (e) F-T TI-UNet

Figure 3: Predictions of the models on different testset splits. Image copyright for (a): Sint-Baafskathedraal Gent @ Lukasweb.be - Arts in Flanders vzw, photo KIK-IRPA.

the others are used to train the models. This is repeated for each patch and the intersection over union (IoU) is averaged as a reliable measure of performance. Each patch is roughly of size 300 × 300 pixels with on average 13% of the pixels belong to a region of paint loss. The visible modality during treatment with and without the annotated regions of paint loss is shown in figure 3 column a) and b) respectively.

For each test set the performance is evaluated for a range of kernels per layer k, illustrated in figure 4a. From the detections generated after the model converged, the median IoU is given bounded by its 25th and 75th percentile. For comparison to our proposed method with different transfer learning schemes, TI-UNet without any form of preprocessing is used. The different predictions are shown in figure 3c with their respective



(a) (b)

Figure 4: Performance in terms of amount of filters per layer. (a) IoU distribution after the model converged, in terms of amount of filters per layer k. (b) Average IoU per pretraining method.

intersection over union score. To test the influence of the number of filters per layer k, the average score over the different test sets is plotted in figure 4b. We notice that from a certain value of k, the performance stabilizes. This average value is summarized in table 1. In addition the accuracy is provided, but it should be noted that this is less informative given the presence of the imbalance in classes.

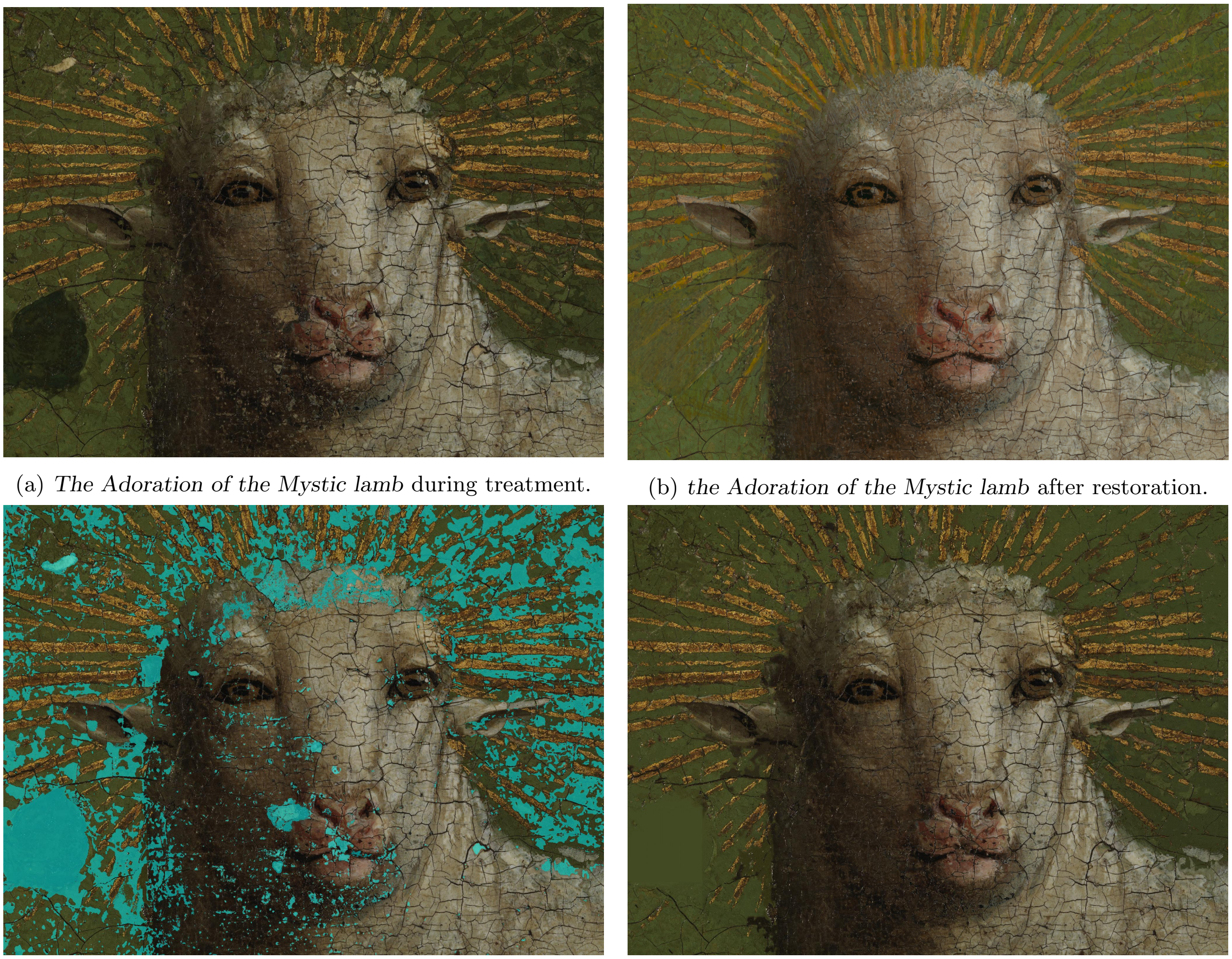
4.3 Unsupervised pretraining of the encoder

To compare the effects of a pretrained encoder, different models are trained through K-fold cross-validation according to section 4.2 for a range of numbers of features per layer k. The results comparing our baseline method and the pretrained encoder are plotted in figure 4. The performance of the trained TI-UNet is significantly improved if the encoder is initialised by the encoder of a pretrained autoencoder (T TI-UNet). From $k > 12$, the performance becomes relatively stable and from the aggregated results in table 1, the IoU is increased from .186 tot .213. The visual results in figure 3 confirmm that the the paint loss is better detected.

When the encoder is transferred, the weights of the decoder do not yet contain any useful information. Training the model could risk part of the initialisation to be unlearned. In order to avoid that, we repeat the training but now first fix the weights of the encoders (F TI-UNet). As the autoencoder is trained to compress all relevant information into a lower dimensional space, it should still be possible to learn to detect paint loss on these pretrained features. When the models are fully converged, all the weights are again set to trainable and a last training step is applied with the goal of finetuning the model (F-T TI-UNet).

As seen in table 1 and figure 4 the model with a fixed encoder (F TI-UNet) performs worse with respect to the standard TI-UNet: An average Jaccard index of 0.173 versus 0.186 respectively. While it shows the encoder does contain most information useful to detect paintloss, there is no access to all relevant features. When training further after setting all layers trainable, it significantly outperforms the TI-UNet, however there is significant difference in performance as both T TI-UNet and F-T TI-UNet have an average Jaccard index of around 0.213. This is advantageous as it means there is no need to first train the decoder alone, and a single training phase is needed to achieve the best generalisation performance using a pretrained encoder.

Another way to evaluate the quality of paint loss detection is to use it as input to virtual inpainting and to evaluate the inpainted result visually. Generating a virtual restoration can serve as a simulation for the actual



(a) *The Adoration of the Mystic lamb* during treatment.

(b) *the Adoration of the Mystic lamb* after restoration.

(c) Paint loss detection.

(d) Virtual inpainting.

Figure 5: Results on the central part of the panel *the Adoration of the Mystic lamb*. (a) and (b) are the images taken during treatment and after physical restoration, respectively, (c) shows the the detected paint loss with the T TI-U-Net in blue and (d) shows a virtual restoration based on the paint loss detection. Image copyright for (b) and (c): Sint-Baafskathedraal Gent @ Lukasweb.be – Arts in Flanders vzw, photo KIK-IRPA

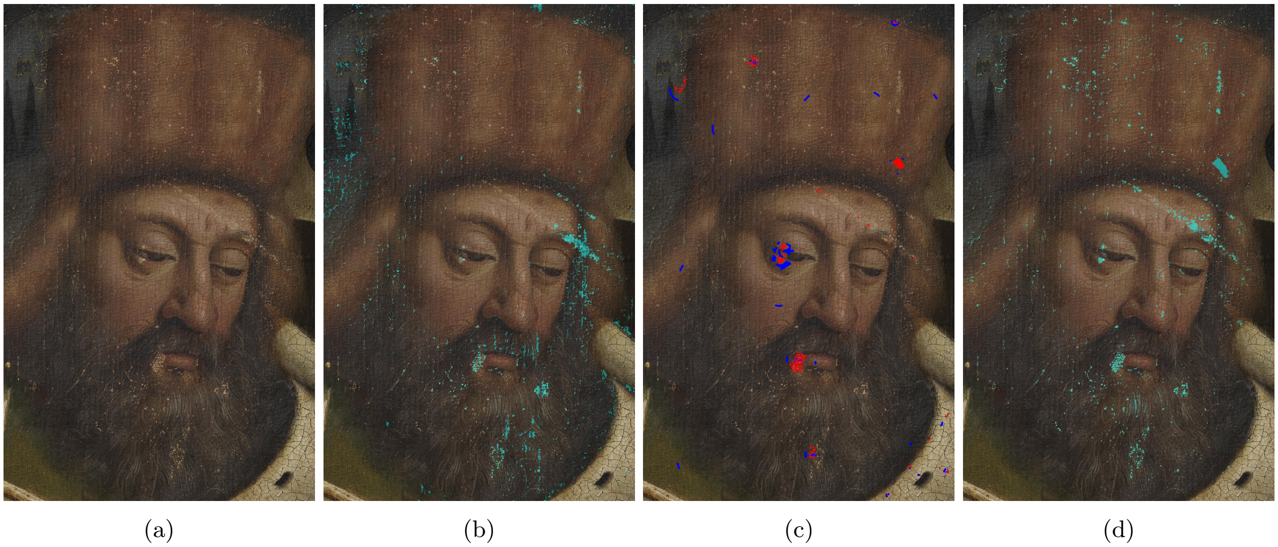


Figure 6: Fast paint loss learning by continuous training from a pretrained model with added annotations. (a) close-up of the *Prophet Zachary* during treatment; (b) detected paint loss based on the network trained on a different painting; (c) added minimal annotations where red marks paint loss and blue marks regions which are not to be restored; (d) paint loss detection after retraining the network. Image copyright for (a): Sint-Baafskathedraal Gent @ Lukasweb.be – Arts in Flanders vzw, photo KIK-IRPA

one and further assist in decision making processes. In figure 5 we show the respective paint loss detection map used as mask to generate the virtual inpainting.[18] It can be seen that the virtual inpainting resembles the actual inpainting rather well. The automatic virtual inpainting filled in reasonably well most of the paint loss regions that have been inpainted by the conservators in the actual restoration process as well. This shows the effectiveness of the paint loss detection and its potential in assisting the actual conservation/restoration treatment.

4.4 Generalisation for other paintings

Applying the model on another painting directly can produce inadequate results. The detected paint loss can however be used as a guide to add annotations to train the model further and achieve a better detection.[21] A good pretraining alleviates the need for a lot of extra annotations. Provided less input is required, we expect the overall quality of the annotations to improve.

In figure 6b we show the detection after applying our previously trained model directly on the panel of *the Prophet Zachary*. While most paint loss regions are at least partially detected, there are substantial amount of false positives while other regions are completely undetected. Based on this prediction, extra annotations were added as shown in figure 6c, mostly guided by the regions mispredicted by the model. This results in a limited extra training to further improve the accuracy and generalisation performance of the model. Our fine-tuned model indeed gives a much better detection, accurately marking regions previously undetected as shown in figure 6d, generated in only a matter of seconds.

5. CONCLUSION

In this paper we designed a deep learning model for paint loss detection in paintings that can be used in a reliable and robust manner without excessive manual labeling by the user for each new painting. We aimed at limiting the need for user interaction by enabling pre-training (on similar paintings) and applying the model to new paintings with relatively small amount of new annotations. By using transfer learning from a multimodal autoencoder to our proposed translation invariant UNet, we were able to significantly improve the classification performance with respect to omitting the pretraining step. While only the encoder was initialised, our results show it is not necessary to first train the decoder of our model, and instead the whole model can be trained at once without any loss of performance. The results were evaluated both objectively, by calculating the overlap measures between manual expert detections and our automatic results, and visually, by assessing virtual inpainting based on the detected paint loss maps. The objective evaluation in terms of the detection accuracy showed clear improvements over our previous deep learning approach in this task. Visual results of automatic virtual inpainting of the detected paint losses show close resemblance with the actual physical restoration, which indicates that paint losses have been accurately detected. Based on this, we can conclude that the proposed approach shows good potential to assist in the actual conservation/restoration treatments.

Acknowledgements

This research was supported by the Artificial Intelligence Research Flanders program funded by the Flemish Government.

REFERENCES

[1] Abry, P., Roux, S. G., Wendt, H., Messier, P., Klein, A. G., Tremblay, N., Borgnat, P., Jaffard, S., Vedel, B., Coddington, J., et al., "Multiscale anisotropic texture analysis and classification of photographic prints: Art scholarship meets image processing algorithms," *IEEE Signal Processing Magazine* **32**(4), 18–27 (2015).

[2] Polatkan, G., Jafarpour, S., Brasoveanu, A., Hughes, S., and Daubechies, I., "Detection of forgery in paintings using supervised learning," in [*2009 16th IEEE International Conference on Image Processing (ICIP)*], 2921–2924, IEEE (2009).

[3] Platisa, L., Cornelis, B., Ruzic, T., Pizurica, A., Dooms, A., Martens, M., De Mey, M., and Daubechies, I., "Spatiogram features to characterize pearls and beads and other small ball-shaped objects in paintings," in [*Vision and material : interaction between art and science in Jan Van Eyck's time*], De Mey, M., Martens, M., and Stroo, C., eds., *Speciale Uitgaven* **6**, 315–329, KVAB PRESS (2012).

[4] Sizyakin, R., Cornelis, B., Meeus, L., Martens, M., Voronin, V., and Pizurica, A., "A deep learning approach to crack detection in panel paintings," in [*Image Processing for Art Investigation (IP4AI)*], 40–42 (2018).

[5] Huang, S., Liao, W., Zhang, H., and Pizurica, A., "Paint loss detection in old paintings by sparse representation classification," in [*Proceedings of the third "international Traveling Workshop on Interactions between Sparse models and Technology" (iTWIST'16)*], 62–64 (2016).

[6] Ružić, T., Cornelis, B., Platiša, L., Pižurica, A., Dooms, A., Philips, W., Martens, M., De Mey, M., and Daubechies, I., "Virtual restoration of the ghent altarpiece using crack detection and inpainting," in [*International Conference on Advanced Concepts for Intelligent Vision Systems*], 417–428, Springer (2011).

[7] Pizurica, A., Platisa, L., Ruzic, T., Cornelis, B., Dooms, A., Martens, M., Dubois, H., Devolder, B., De Mey, M., and Daubechies, I., "Digital image processing of the ghent altarpiece: Supporting the painting's study and conservation treatment," *IEEE Signal Processing Magazine* **32**(4), 112–122 (2015).

[8] Meeus, L., Huang, S., Devolder, B., Dubois, H., Martens, M., and Pizurica, A., "Deep learning for paint loss detection with a multiscale, translation invariant network," in [*International Symposium on Image and Signal Processing and Analysis, ISPA*], **2019-Septe**, 158–162 (2019).

[9] KIK/IRPA, "Belgian art links and tools." http://balat.kikirpa.be/ (2018).

[10] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," in [*International Conference on Medical image computing and computer-assisted intervention*], 234–241, Springer (2015).

[11] Kulkarni, T. D., Whitney, W., Kohli, P., and Tenenbaum, J. B., "Deep Convolutional Inverse Graphics Network," *Advances in Neural Information Processing Systems* **2015-Janua**, 2539–2547 (3 2015).

[12] Yu, F. and Koltun, V., "Multi-Scale Context Aggregation by Dilated Convolutions," in [*ICLR*], (1 2016).

[13] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *science* **313**(5786), 504–507 (2006).

[14] Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., and Hikosaka, S., "Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery," *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018* **2018-Janua**, 1442–1450 (2018).

[15] Clevert, D.-A., Unterthiner, T., and Hochreiter, S., "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in [*ICLR*], (1 2016).

[16] LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T., "A theoretical framework for back-propagation," in [*Proceedings of the 1988 connectionist models summer school*], **1**, 21–28, CMU, Pittsburgh, Pa: Morgan Kaufmann (1988).

[17] Xie, X., Meeus, L., and Pižurica, A., "Partial convolution based multimodal autoencoder for ART investigation," *CEUR Workshop Proceedings* **2491**, 1–15 (2019).

[18] Ruzic, T. and Pizurica, A., "Context-Aware Patch-Based Image Inpainting Using Markov Random Field Modeling," *IEEE Transactions on Image Processing* **24**, 444–456 (1 2015).

[19] Tschopp, F., Martel, J. N. P., Turaga, S. C., Cook, M., and Funke, J., "Efficient convolutional neural networks for pixelwise classification on heterogeneous hardware systems," in [*2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*], **2016-June**, 1225–1228, IEEE (4 2016).

[20] KIK/IRPA, "The ghent altarpiece restored." http://closertovaneyck.kikirpa.be/ghentaltarpiece (2020).

[21] Pan, S. J. and Yang, Q., "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering* **22**, 1345–1359 (10 2010).