Deep Learning for Paint Loss Detection with a multiscale, translation invariant network

Laurens Meeus*, Shaoguang Huang*, Bart Devolder†, Hélène Dubois‡§,
Maximiliaan Martens§, and Aleksandra Pižurica*
*Group for Artificial Intelligence and Sparse Modelling, Ghent University - imec, Belgium
†Princeton University Art Museum, NJ USA
‡Royal Institute for Cultural Heritage, KIK/IRPA Brussels, Belgium
§Department of Art, Music and Theatre Sciences, Ghent University, Belgium
Corresponding author: L. Meeus (e-mail: laurens.meeus@ugent.be).

***Abstract*—We explore the potential of deep learning in digital painting analysis to facilitate condition reporting and to support restoration treatments. We address the problem of paint loss detection and develop a multiscale deep learning system with dilated convolutions that enables a large receptive field with limited training parameters to avoid overtraining. Our model handles efficiently multimodal data that are typically acquired in art investigation. As a case study we use multimodal data of the *Ghent Altarpiece*. Our results indicate huge potential of the proposed approach in terms of accuracy and also its fast execution, which allows interactivity and continuous learning.**

***Index Terms*—Art investigation, paint loss, multi-modal data, semantic segmentation, deep learning, transfer learning.**

I. Introduction

In art investigation typically high resolution images are acquired in different modalities, including visible, infrared, radiography, and others. For the study of old paintings these multimodal data provides extra information of the different layers of paint and materials used, allowing non-destructive analysis.

During the conservation and restoration of old paintings, one of the tasks consists of documenting and mapping the lacunas, the regions of paint loss. Lacunas are mostly a result of drying and flaking of paint through aging, although rough handling can also introduce losses. An example of visible paint loss is shown in Figure 1b. Currently, this documentation involves a lot of manual work since available software can only give a coarse estimation of the paint loss. This makes the process rather tedious. In order to improve the automated mapping, smarter image processing techniques are sought.

While paint loss is mostly visible in RGB images, it is not always differentiable from the background. Additional modalities can provide extra information to make a better assessment. Figure 1 shows an example of different modalities used in our work in a case study on the *Ghent Altarpiece*. Some of these images are available (in a compressed form) on the website *http://closertovaneyck.kikirpa.be/* and others were acquired at different stages of the restoration campaign [1].

Technical literature on paint loss detection is limited. Huang et al [2] reported promising results with sparse representation classification (SRC), surpassing common machine learning approaches like linear regression classification and support vector machines in this task. The current techniques make a classification mainly based on the spectral components while the receptive field is limited to a few pixels. Moreover these methods typically involve tuneable parameters that need to be adjusted for different paintings. The best performing methods reported so far, based on sparse coding [2], also tend to be slow when it comes to processing large paintings.

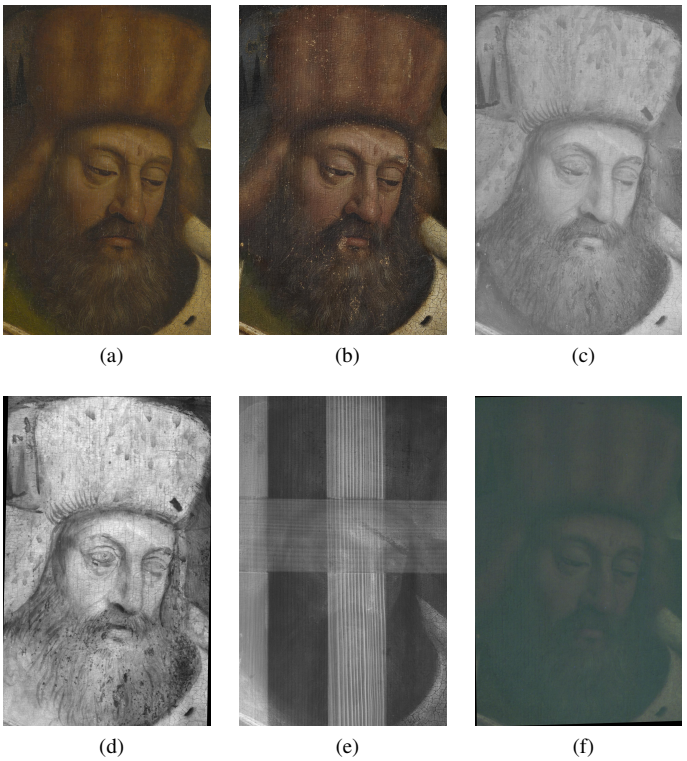


Fig. 1. Different modalities of the panel *the Prophet Zachary*, acquired during the restoration treatment of the *Ghent Altarpiece*. (a) and (b) are RGB acquisistions before and during treatment respectively, (c) and (d) are infrared and infrared reflectography, (e) X-ray, and (f) RGB image from ultraviolet fluorescence. Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

The state-of-the-art in supervised image segmentation is dominated by convolutional deep neural networks [3], [4], [5], [6], [7], [8], [9]. We build our approach on a recent convolutional neural network architecture U-Net [4]. U-Net is an encoder-decoder network, similar to convolutional auto-encoders [10], with skip connects [11] to combine local information of the encoder with global information captured in the lower resolution layers of the decoder. Given the big range in size of paint loss regions the multi-scale approach is especially useful.

In contrast to most of these other segmentation applications, we have to deal with scarce and unreliable annotations on a

pixel level. As data in art investigation is typically very costly to obtain and thus limited, the complexity of the model should be modest to avoid overtraining. This limits the amount of layers and filters per layer and as a consequence the receptive field is also limited. This will also guide the design of our architecture. Moreover, we develop our system such to handle efficiently multi-modal data. As the size of paint losses can range from a few to hundreds of pixels, the algorithm combines spectral information with a large enough spatial support to exploit well the relevant spatial information.

We will validate our method on the panels of the *Ghent Altarpiece* [1], a monumental triptych made by the brothers van Eyck in the 15^{th} century. We also propose a transfer learning scheme to efficiently apply the system to other paintings with very limited user-input.

II. Method

In this section we build a new method for pixel-level target labeling in multimodal data motivated by recent developments in deep learning. The proposed architecture, illustrated in Figure 2, efficiently uses spatial information to avoid overtraining on the limited amount of annotations. Inspired by U-Net [4], 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. A dilated convolution is convolutional of which the weights of the kernel W are spaced out by a factor k, the dilation rate:

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

This way the produced output z 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.

Starting from one, the dilation rate is doubled after each pooling 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 [8]. 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 1×1 convolutional layer with as many filters as there are classes. With paint loss and background, this is two output maps. We set the activation of all layers to ELU [13], except for the last one, where we use softmax. This is to produce a normalised vector $\hat{y}$, corresponding to the likelihood that a pixel belongs to the classes paint loss and background:

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

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, version of the prediction y.

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

Unlike architectures with decimating layers, each layer of the proposed architecture is explicitly translation invariant, such that the whole system is translation invariant, which is a desired property of pixel labeling algorithms. Because no spatial information is lost after a pooling layer, translation invariance does not have to be learned and the amount of filters is kept constant in each layer.

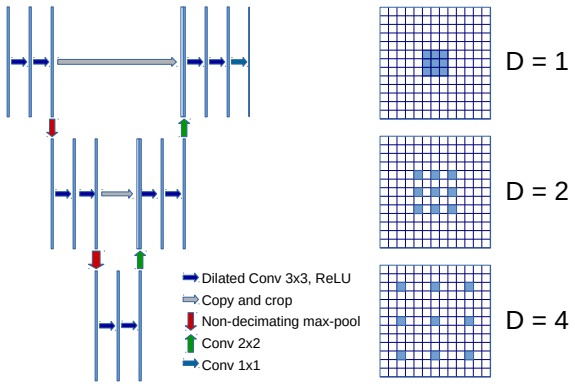
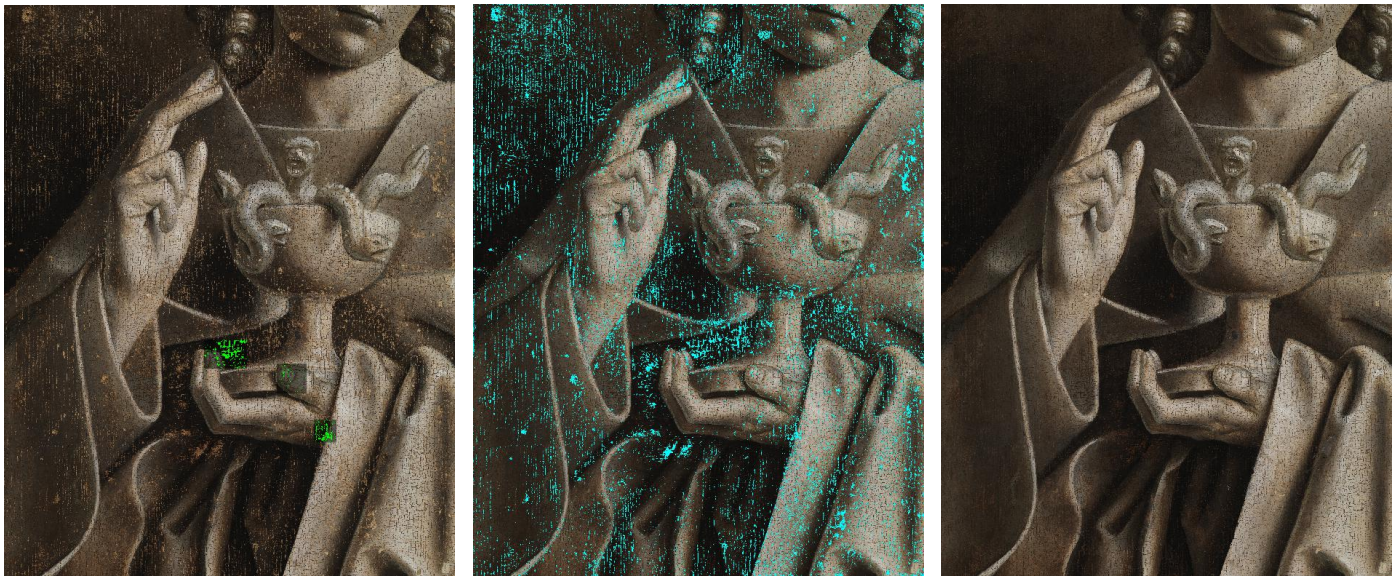


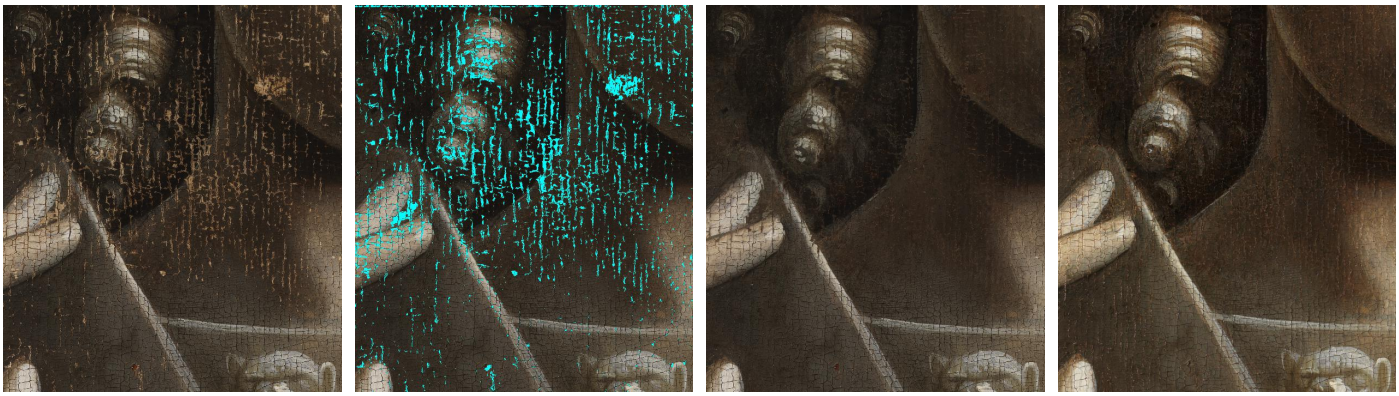
Fig. 2. Schematic of the proposed architecture with two pooling layers. Left: The complete network with different layer types. Wider rectangles denote larger amount of filters. Each convolutional layer applies the same amount of filters and operates on the same resolution. Right: the dilated kernel and dilation rates for the convolutional layers on each row.

III. Results

Three experiments are conducted to validate paint loss detection with the proposed system with efficient use of annotations. Firstly the paint loss detection is tested on a single painting based on dense annotations Secondly the quality of the map is checked by using the damage map for virtual inpainting. Lastly the ability to generalise is tested by applying the model directly to another panel and updating it with a limited set of extra annotations. The results are shown on images of the ongoing restoration of the *Ghent Altarpiece*



(a) (b) (c)



(d) (e) (f) (g)

Fig. 3. Paint loss detection map and virtual restoration on the panel *John the Evangelist*. The annotations of paint loss are visible in green in (a). The paint loss prediction map (b) is used as mask for virtual restoration (c). A close-up of the shoulder (d) shows the maintained details of the prediction (e). The paint loss prediction (f) is comparable to the physical restoration (g). Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

A. *Paint loss detection*

Training the model for paint loss detection requires annotations from an expert. Both examples of paint loss and background are needed. For the *Ghent Altarpiece* we have a total of $807,740$ annotated pixels of which 8.3% is paint loss. This amount is increased by a factor 8 after data augmentation by rotations of $90°$ and flips. An example of dense annotations on the panel *John the Evangelist* is visible in Figure 3a. Per smaller regions, all the paint loss is annotated in green while the rest is background. For the input data, the different image modalities are registered and stacked to a single image with 12 spectral components. The ground truth map is converted per pixel to a one-hot encoding. The model is trained with cross-entropy cost on crops around the annotated pixels of both the input and output. Taking into account the receptive field, the input and output patches are set to size 146×146 and 100×100 respectively. While one patch contains mostly similar context and thus possibly less variety, due to shared computations, it only adds a limited increase in computation time. Moreover, more data in a single training batch smoothens the gradient updates, making the training more stable.

While relatively limited with respect to the size of the whole panel, training on this set of samples is sufficient to generate an accurate map of paint loss for the whole panel as seen in Figure 3b. Taking a more detailed look, Figure 3e shows the detection on a close-up of the shoulder. The model appears to be not only accurate in the vicinity of annotations but it also generalises well over the whole panel.

In terms of speed, with a GeForce GTX1070, training converges after approximately one hour. Because in processing neighbouring pixels a lot of computations can be shared, our algorithm processes a relatively large image region jointly with limited increase in computation time compared to processing a single pixel. Because of memory limitations, it is not possible to run the inference on the whole panel at once, so instead overlapping patches of size 446×446 are processed individually, and the output patches of size 400×400 are stitched back together for the whole detected paint loss map. To give an idea of processing speed, the results of Figure 3b ($44.9\,\mathrm{MP}$ image) are generated in mere seconds. Including pre- and post-processing, the whole process takes less than one minute. This process efficiently allows interactive use and indicates an excellent potential for practical applicability.

B. Virtual inpainting

Besides documentation purposes, the damage map is an essential input for virtual inpainting. This virtual restoration can be useful to provide a quick prediction of how the final restoration may look like. To be of relevance to art restorers, the damage map should not include craquelure as when the paintings are physically restored, typically the craquelure stays untouched. An accurate paint loss map is a necessary prerequisite for satisfactory virtual inpainting and thus we can use virtual inpainting to assess the quality of our system.

Figure 3c shows the virtually restored panel of *John the Evangelist* by applying the method of [14] on our detected paint loss map. An enlarged detail in Figure 3f shows how with respect to Figure 3e most of the paint loss is not visible anymore. Compared to Figure 3g, visually the fully automatic processing result resembles well the physical restoration, especially since the craquelure is well maintained. This shows that our detection of paint paint loss regions coincides well with the regions that were manually detected and restored by the experts.

C. Generalisation to other paintings

Ideally after training, it is desired to have an intelligent system that can detect the paint loss on other paintings as well. Because of the diversity in i.a. content, art style, age, and amount of abrasion, a trained model may not be sufficiently accurate on arbitrary new data.

In Figure 4b and 4f, we applied our previously trained model directly on parts of the panel *John the Baptist*. The results show that the model can differentiate some paint loss from background, however it does not generalise well enough as large regions are under- or overdetected.

We apply transfer learning [15] to reduce the amount of user-input. Instead of requesting a new set of dense annotations, extra annotations are limited as illustrated in Figure 4c. The previously trained model is then fine tuned on these new annotations with a reduced amount of training steps. As the annotating and training is so limited, we effectively create a very fast training scheme.

The new results in Figure 4d show how the model improved to get an accurate detection. Even in other regions of the same painting, there is no clear indication of strongly misclassified regions and shows again a strong generalisation capability over the whole panel. Instead of having to train the network from scratch for around one hour, the pretrained model converges on the new data after around 5 minutes of training.

We conclude the model shows possibility to generalise well to other paintings where limited annotations are needed to improve the network to adjust to the given panel, making it possible to segment large resolution acquisitions of a whole panel, including annotation and training time, in an acceptable time, making it a strong candidate for practical use.

IV. Conclusion

In this paper we showed how deep learning can assist for the detection of paint loss during the restoration of paintings. The results on the ongoing restoration of the *Ghent Altarpiece* show that our system is both accurate and fast. The proposed deep learning architecture was optimized to employ a big receptive field and multimodal data. The paint loss is accurately detected over the whole panel, with relatively few annotations. These results are appreciated by the art restorers. The virtual restoration based on our paint loss detection showed close resemblance to the physical restoration. This means the detection is in agreement with the physical restored regions. For other paintings the system is applicable after updating the pre-trained model further on a limited set of extra annotations. Whilst preserving accuracy, this led to a huge speedup of the whole process from hours to minutes, indicating a good potential for practical use.

References

[1] KIK/IRPA, "Belgian art links and tools," 2018. [Online]. Available: http://balat.kikirpa.be/

[2] S. Huang, W. Liao, H. Zhang, and A. Pizurica, "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)*, 2016, pp. 62–64. [Online]. Available: http://arxiv.org/pdf/1609.04167v1.pdf

[3] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 4 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2572683

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv*, 5 2015. [Online]. Available: https://www.doi.org/10.1007/978-3-319-24574-4_28

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 12 2017. [Online]. Available: https://www.doi.org/10.1109/TPAMI.2016.2644615

[6] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2699184

[7] D. Minh Nguyen, W. Ding, A. Munteanu, N. Deligiannis, and Y. Liu, "Hourglass-ShapeNetwork Based Semantic Segmentation for High Resolution Aerial Imagery," *Remote Sensing*, vol. 9, no. 6, p. 522, 2017. [Online]. Available: https://www.doi.org/10.3390/rs9060522

[8] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "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*, vol. 2018-Janua, pp. 1442–1450, 2018. [Online]. Available: https://www.doi.org/10.1109/WACV.2018.00162

[9] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv preprint*, pp. 1–23, 2017. [Online]. Available: http://arxiv.org/abs/1704.06857

[10] G. E. Hinton and R. S. Zemel., "Autoencoders, Minimum Description Length and Helmholtz free Energy," *Advances in Neural Information Processing Systems (NIPS)*, vol. 3, no. 3, 1994. [Online]. Available: https://doi.org/10.1021/jp906511z

[11] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-46976-8_19

[12] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv*, 11 2015. [Online]. Available: http://arxiv.org/abs/1511.07122

[13] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *CoRR*, 11 2015. [Online]. Available: http://arxiv.org/abs/1511.07289

[14] T. Ruzic and A. Pizurica, "Context-Aware Patch-Based Image Inpainting Using Markov Random Field Modeling," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 444–456, 1 2015. [Online]. Available: http://doi.org/10.1109/TIP.2014.2372479

[15] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 10 2010. [Online]. Available: https://doi.org/10.1109/TKDE.2009.191

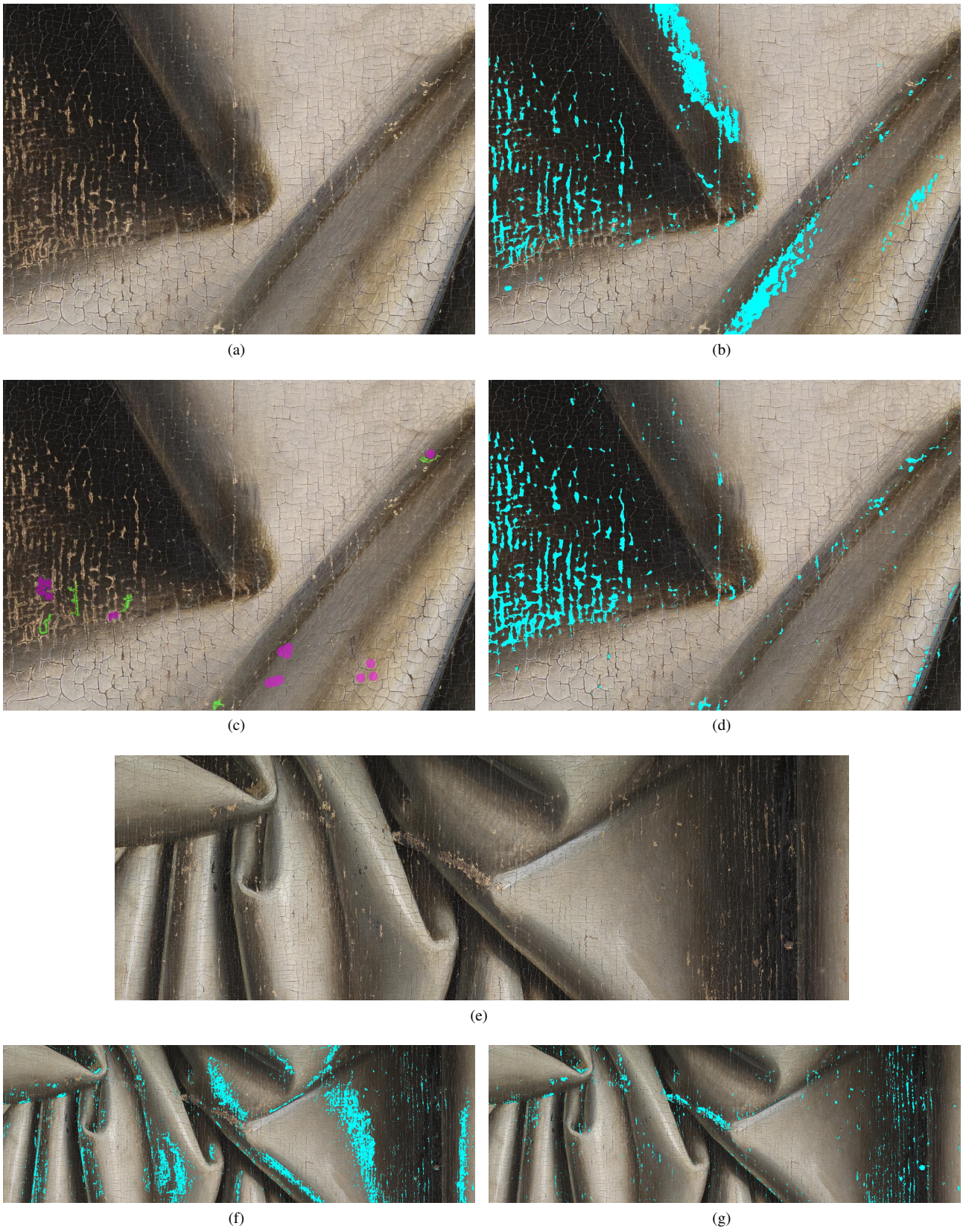


Fig. 4. Detection results on the panel *John the Baptist*. Figure (a)-(d) is of a close-up on a central part of the panel while (e)-(f) is from a larger, separate region. It shows the panel during treatment (a), (e); the detected paint loss map by directly applying a pretrained model (b), (f); limited set of extra annotation (c); and improved detected paint loss map after continued learning (d), (g). Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.