Partial Convolution Based Multimodal Autoencoder for Art Investigation

Xianghui Xie[1], Laurens Meeus[2], and Aleksandra Pižurica[2]

[1] Faculty of Engineering Technology, KU Leuven, Belgium
[2] Department of Telecommunications and Information Processing, TELIN-GAIM, Ghent University, Belgium

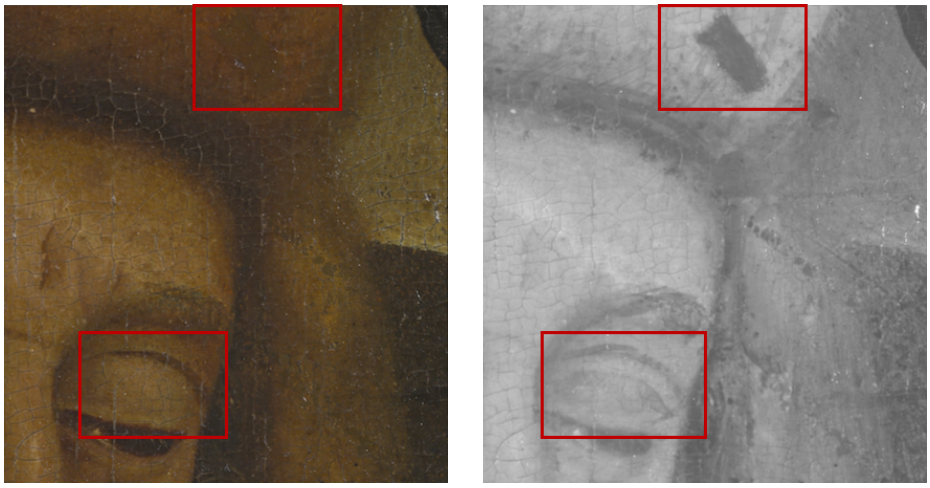
Abstract. Autoencoders have been widely used in applications with limited annotations to extract features in an unsupervised manner, preprocessing the data to be used in machine learning models. This is especially helpful in image processing for art investigation where annotated data is scarce and difficult to collect.
We introduce a structural similarity index based loss function to train the autoencoder for image data. By extending the recently developed partial convolution to partial deconvolution, we construct a fully partial convolutional autoencoder (FP-CAE) and adapt it to multimodal data, typically utilized in art invesigation. Experimental results on images of the *Ghent Altarpiece* show that our method significantly suppresses edge artifacts and improves the overall reconstruction performance. The proposed FP-CAE can be used for data preprocessing in craquelure detection and other art investigation tasks in future studies.

Keywords: Autoencoder, Partial convolution, Multimodal data

1 Introduction

Art investigation aims at developing and applying technologies to facilitate research and conservation of artworks. Some typical research topics are craquelure detection, paint loss detection, virtual reconstruction and so on. In recent years, deep learning has shown great potential in computer vision tasks, which attracts researchers to apply deep learning methods to art investigation. However, existing studies mainly utilize fully supervised learning that relies on a great number of annotation data, a requirement that is hard to come by in art investigation.

Using autoencoders as a data preprocesser for feature extraction, is very common in deep learning when only limited amount of annotated data is available. Autoencoders can be trained in an unsupervised manner so that it learns to extract the most important features from a particular dataset, e.g. paintings from the same artist. After unsupervised learning, the latent vector of the autoencoder can then be used as the input of models for the art investigation tasks. Since these models are applied on a compressed representation of the input, they can be of lower complexity and contain less parameters. Accordingly, less annotated



(a) (b)

Fig. 1: Different modalities of the panel *the Prophet Zachary*. (a) RGB and (b) Infrared reflectography image. Different types of degradations become visible in these images. Intermodal information through the variations in different modalities can be utilized. Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

data is needed to train these models to the same performance level. In this way, autoencoders can be a powerful tool for art investigation.

Besides photography, other sensors are commonly employed in art investigation in order to acquire more information about a particular object. Our methods are applied on images acquired in the ongoing restoration project of the *Ghent Altarpiece* [9] where at least five modalities are obtained: macro-photography before and during treatment (RGB, three color channels each), Infrared reflectography (IRR, single channel), X-ray (single channel) and ultraviolet fluorescence (UVF, three color channels). Researchers have used these multimodal data for craquelure [23] and paint loss detection [6,16], although are restricted in performance to the availability in annotations. Images from two different sensors of the painting *the Prophet Zachary* can be found in figure 1. In both images, craquelure and regions of paint loss are visible. A disparity in context can show overpainted regions, e.g. in the red rectangles. To achieve good data preprocessing for art investigation tasks, the autoencoder must be able to extract both inter- and intramodal features.

To assess the quality of this encoding with respect to the compression factor, the reconstruction performance is commonly analyzed. To improve the reconstruction performance, we have 3 main contributions: a fully partial convolutional autoencoder, a structural similarity (SSIM) index based loss function, and separating the inputs for a multimodal autoencoder. Firstly, we generalize

partial convolutions [14] and extend it to partial deconvolutions. As a result, we construct a novel fully partial convolutional autoencoder (FP-CAE) which significantly reduces edge artifacts on the reconstructed images. When training the autoencoder, we introduce a SSIM-based loss function to maximize the structural similarity between the input and reconstructed images. Finally, we investigate two strategies to improve extracting inter- and intramodal information from multimodal data.

The paper is organized as follows: We review briefly autoencoder designs, variation of CNNs, existing studies in art investigation, and multimodal data processing in section 2. Our generalization and extension of partial convolutions, the proposed model structure and loss function are explained in section 3. The experiments and results are discussed in section 4. Finally, section 5 draws some conclusion and encloses the paper.

2 Related work

In this section, we first review a few recent studies in art investigation as well as autoencoder and then discuss some variation of autoencoder structure. Finally, we briefly summarize some relevant work in multimodal data processing.

2.1 Art investigation and autoencoder

Some recent studies adopted deep learning methods in art investigation tasks. A U-net like structure was used in [16] to detect paint loss while Sizyakin et al. proposed to combine morphological filtering with CNN for crack detection [23]. Existing deep learning models used in art investigation are based on supervised learning, which is constrained by the limited annotations. Therefore, more research exploring unsupervised or semi-supervised learning such as using autoencoders is needed to improve these methods.

Autoencoder has been applied to fields such as medical image processing where annotations are limited [5]. When training an autoencoder, the mean squared error is typically employed as the loss function [4], [5]. However, Snell et al. have shown that SSIM based loss function can achieve better performance than MSE loss for images [24].

2.2 Autoencoder model structure

Various autoencoder structures exist for different applications. The variational autoencoder [12] is a stochastic autoencoder and is very popular in especially generative models [17], [19], [21]. Another category, the deterministic autoencoder, has been widely used in feature extraction and reconstruction. Some used stacked autoencoders to reduce the noise from input data [22].

In deep learning models, convolutional neural networks have been proved more effective compared to fully connected networks. Since convolution is implemented by sliding kernels along the input, one challenge researchers have to

deal with is preserving the border information when applying CNN. Carlo et al. proposed to use extra filters explicitly to learn the border information [7]. However, the total filters and parameters increase quickly when the kernel size increases. This limits its application with large kernel sizes and those that require fast computations. Another widely used technique to cope with border information in convolution is padding. Zero padding [13], reflection padding and duplication padding are the most common padding methods researchers use. All these methods introduce artificial values in the border, which does not necessarily correspond to the real value outside the border hence this leads to edge artifacts. Liu et al. proposed using partial convolution for image inpainting tasks [14]. In their method, appropriate scaling is applied to counter balance the varying amount of valid inputs. Since zero padding can be regarded as a special case of missing values by defining the input region to be non-holes and zero padded region to be holes, partial convolution based padding is used to reduce edge artifacts [15]. Their results suggest that partial convolution could indeed improve the segmentation accuracy on the edges.

Transposed convolution or deconvolution has been used as the basic building block for convolutional decoders [1], [10]. The most commonly used implementation of deconvolution first stretches the input feature by inserting zeros between each input unit and then applies the kernel to the stretched input with stride equal to 1 [3]. Since zero insertions are used in the deconvolution, checkerboard artifacts are easily introduced [18]. As an alternative, Ronneberger et al. used upsampling [20] to build the decoder. However, upsampling also introduces artificial values when applying interpolation to the input feature, which leads to other kind of artifacts.

2.3 Multimodal data processing

Multimodal data processing has attracted attention from researchers in recent years as more and more correlated data from different sensors are collected. Cadena et al. proposed separating depth sensor data, image and semantics in the input and combining encoded features in latent space to predict depth [2]. Jaques et al. investigated the possibility of combining data from text, number, location, time and survey for mood prediction [8]. Canonical correlation analysis based intra and inter modal information learning was introduced in [25] for RGB-D object recognition task. In art investigation, Meeus et al. stacked all modalities together for paint loss detection [16]. Given all the possible variables of data source, it is still an ongoing research topic of how to effectively combine different modalities to obtain correlated information nowadays.

3 Method

In this section, we start with illustrating how we extend the partial convolution and then explain the structure of our multimodal autoencoder as well as the proposed loss function.

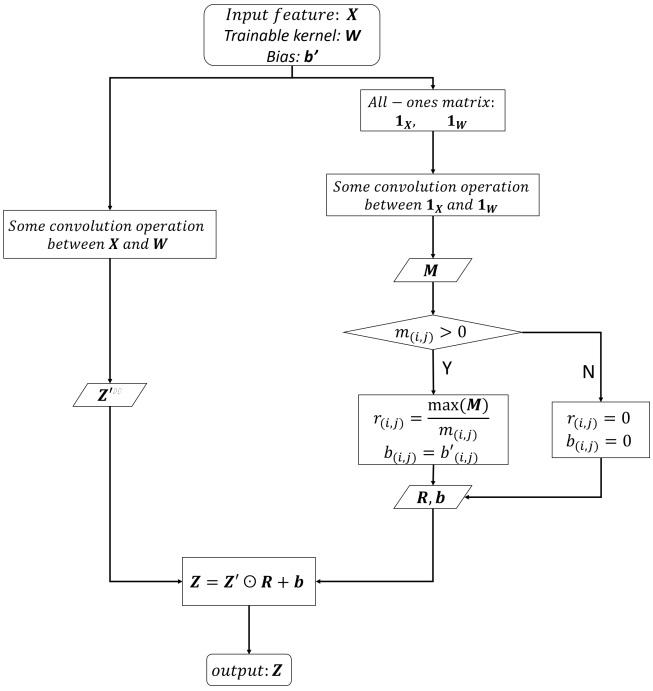


Fig. 2: The general flowchart for implementing partial convolution. The convolution operation can be 1D convolution, 2D convolution, transposed convolution etc.

3.1 Extending Partial convolution

General method for implementing partial convolution From the definition of partial convolution with zero padding [14], [15], we generalized a method for implementing partial convolution, see figure 2. Given the input feature $\mathbf{X}$, the trainable kernel $\mathbf{W}$ and bias $\mathbf{b}'$ (if not zero) of the current layer, two all-ones matrix 1_X and 1_W, having the same shape as $\mathbf{X}$ and $\mathbf{W}$ respectively, are generated. Some convolutional operation such as Conv1D, Conv2D or Conv2DTranspose is applied to $\mathbf{X}$ and $\mathbf{W}$, yielding $\mathbf{Z}'$. The same convolutional operation is also applied to $\mathbf{1}_X$ and $\mathbf{1}_W$, yielding a non-scaled mask $\mathbf{M}$. Instead of calculating the L_1 norm of all-ones matrix in [14], we take the maximum value of $\mathbf{M}$ as the numerator so that the minimum value of the scale factor is one. This way we enforce that the convolution result is not changed in the region where all elements are valid inputs. In the extreme case when all the elements where the region kernel applies are zeros, the scale factor and bias will be set to zero. Finally, $\mathbf{Z}'$ is multiplied element-wise with the scale factor $\mathbf{R}$. Bias and non-linearity can be applied after this multiplication.

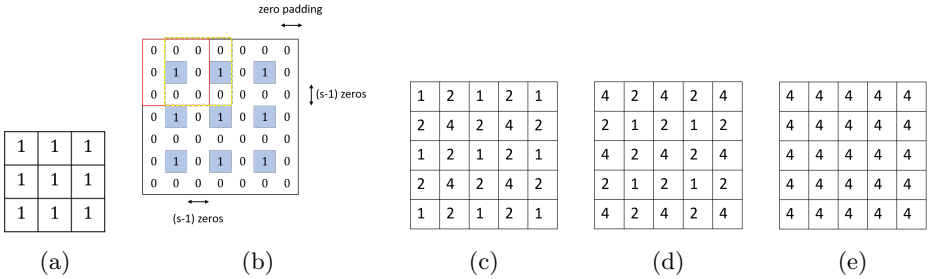


Fig. 3: Visualization of our partial deconvolution. (a). An input feature and filter matrix. (b). The stretched and zero-padded input feature. (c) Output of normal deconvolution. (d) The scale factor $r_{(i,j)}$. (e). Output of our partial deconvolution. Our partial deconvolution smooths the output of a normal deconvolution by multiplying with the appropriate scale factor based on the varying amount of valid inputs.

Partial deconvolution The zero insertion and padding used in the deconvolution can be regarded as missing input values, thus partial convolution can be applied. Let $\mathbf{X}$ be the input feature of current deconvolution layer and $\mathbf{1_X}$ be all-ones matrix with the same shape as $\mathbf{X}$. $\mathbf{X}^{ext}$ and $\mathbf{1}^{ext}$ is the stretched and zero padded result on X and $\mathbf{1}_X$ respectively. When a kernel $\mathbf{W}$ is applied to a local region of the input feature $\mathbf{X}_{(i,j)}$, the partial deconvolution result is:

$$z_{(i,j)} = \mathbf{W}^T(\mathbf{X}^{ext}_{(i,j)} \odot \mathbf{1}^{ext}_{(i,j)})r_{(i,j)} + b = \mathbf{W}^T\mathbf{X}^{ext}_{(i,j)}r_{(i,j)} + b. \tag{1}$$

The scale factor $r_{(i,j)}$ is defined as:

$$r_{(i,j)} = \frac{max(\mathbf{M})}{\mathbf{M}_{(i,j)}}, \tag{2}$$

where $\mathbf{M}$ is the deconvolution result of $\mathbf{1}^{ext}$ and the all-ones kernel $\mathbf{1}_W$, having the same shape as $\mathbf{W}$. The visualization of our partial deconvolution can be found in figure 3. In this example both the input feature $\mathbf{X}$ and the kernel $\mathbf{W}$ are 3×3 all-ones matrix. The input feature is first stretched to a 5×5 matrix and becomes 7×7 matrix after padding. The normal deconvolution result is shown in figure 3c while our partial deconvolution result is shown in figure 3e. By multiplying with scaling factor $r_{(i,j)}$, the appropriate adjustment is applied to different input regions with varying number of valid elements. Therefore, the partial deconvolution could smooth out the variation in output values and thus suppress edge artifacts.

3.2 Multimodal autoencoder

We proposed two autoencoder architectures to cope with the multimodal data: stacked input and separated input autoencoder. The main difference of these two structure is the strategy to combine different modalities in the input.

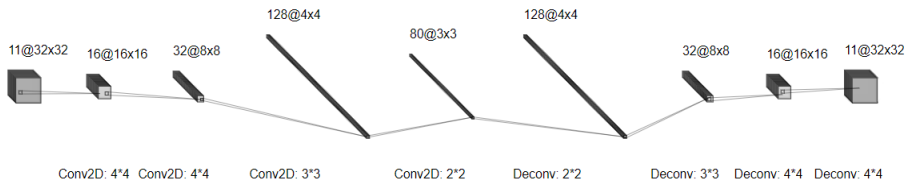


Fig. 4: Model structure of stacked input autoencoder. All image modalities are stacked together in the input layer so the input channel depth is 11.

Our model structure for stacked input autoencoder is a fully convolutional neural network, see figure 4. Images from different modalities are stacked together as a single input for the autoencoder. To reduce edge artifacts, different versions of the model are tested by replacing the convolution and deconvolution layers with partial convolution, deconvolution, or upsampling layer. For this model, the input shape is $32 \times 32 \times 11$ while the latent vector shape is $3 \times 3 \times 80$, thus the data compression ratio is 15.6.

For the separated input autoencoder, each modality has an encoder to extract important intra-modal features, illustrated in figure 4. The encoded features are combined either by an addition or a concatenation layer. Then these combined features are given to a convolutional layer to learn the inter-modal information. Finally, the learned inter-modal features are distributed to decoders for each modality to reconstruct the multimodal images. The encoder and decoder used for multi-modal autoencoder have the same structure as the stacked input autoencoder, only the input channel depth changes. The output of the convolution layer for the inter-modal features is used for later art investigation tasks. Therefore, the dimension of latent vector is again $3 \times 3 \times 80$ and compression ratio stays 15.6.

3.3 Loss function

SSIM is widely used as a metric to compare the similarity between two images. The single scale SSIM consists of three components: luminance (L), contrast (C) and structure (S). With μ and σ^2 the average and variance operator respectively, they are defined as $L(x,y) = \frac{(2\mu_x\mu_y+C_1)}{(\mu_x^2+\mu_y^2+C_1)}$, $C(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}$, $S(x,y) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}$. The SSIM score is calculated by combining these three functions:

$$SSIM(x,y) = L(x,y)^\alpha C(x,y)^\beta S(x,y)^\gamma. \tag{3}$$

As usually $\alpha = \beta = \gamma$ and $C_3 = \frac{C_2}{2}$, the SSIM can be rewritten as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

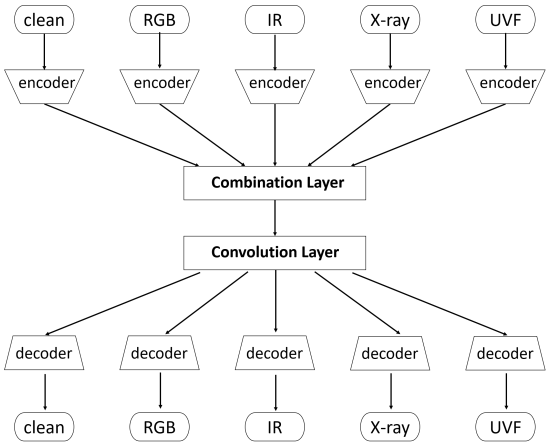


Fig. 5: Separated input autoencoder model structure. The different modalities are separated. Intra-modal information is learned by the encoder and decoder while inter-modal information is extracted by the convolution layer. The combination layer could be either an addition or a concatenation layer.

Based on the definition, the SSIM score ranges from -1 to 1 and only when two images are identical the score can be equal to one. For the proposed loss function, the logarithm is applied on a shifted and rescaled SSIM, in order to punish a low SSIM more.

$$Loss = -log(\frac{SSIM + 1}{2}) \tag{5}$$

The more similar two images are, the smaller the loss is. Given the properties of the logarithm function, when SSIM is small, the loss value and and gradient is high, thus pushes bigger model update steps. When SSIM is closer to one, the gradients become smaller and the model optimizes the parameters in a more stable way. Since the SSIM is only defined for grey scale images, we apply grey scale to images that have three color channels such as RGB and UVF images before calculating its SSIM. For the multimodal autoencoder, the final loss is the mean loss on each modality.

4 Results and Discussion

All our models are applied on multimodal acquisitions of two panels from the *Ghent Altarpiece*: *John the Evangelist* and *the Prophet Zachary* [9]. The five modalities mentioned in section 1 are used, totalling 11 color channels. For each painting, we first divide the full image into two roughly equal parts. One part is used as the training data while the other part is used for testing. Then the full image is further cropped into small squares to match the input dimension

of our model. Horizontal, vertical and diagonal flip are randomly applied to the patches. This way, around 2.6 million images and 2 million images are available for training and testing respectively.

We started our experiments by testing performance of stacked input autoencoders, i.e. all modalities are stacked before being given to the model. The kernel size, stride, input and output dimensions of the different stacked input autoencoders are the same as illustrated in figure 4. The convolution layers might be replaced by partial convolution, upsampling or deconvolution layer depending on the model configuration. As baseline, we first train an autoencoder whose layers are normal convolution and deconvolution layers (*normal AE*). For the second model, the convolutional layers in encoder are replaced with partial convolution layers while the decoder remains the same (*PEN + NDE*). The third model has the same encoder as the second model and the deconvolution layers are replaced with upsampling and partial convolution layers (*PEN + UPDE*). Nearest interpolation is used for the upsampling method. The last model is constructed by replacing all normal convolution and deconvolution layers with partial convolution and deconvolution layers, which becomes our fully partial convolutional autoencoder (*FP-CAE*). Adam optimizer [11] was used to optimize parameters. The learning rate for the baseline model was set to 6e−4 without decay. However, with partial convolution layers we found that higher learning rate is desired in order to achieve good performance. The learning rate for the other three models is 12e−4 with 3e−5 decay. All models are trained until convergence.

The stacked input fully partial convolution autoencoder was used as basic unit to construct the separated input autoencoder. We first combined the different encoded features by concatenating them together and then applied a convolution layer (*Concatenation FP-CAE*), as illustrated in figure 5. In the second multimodal autoencoder, the combination layer is an addition layer(*Addition FP-CAE*). The learning rate for both models is 8e−4 with 4e−5 decay.

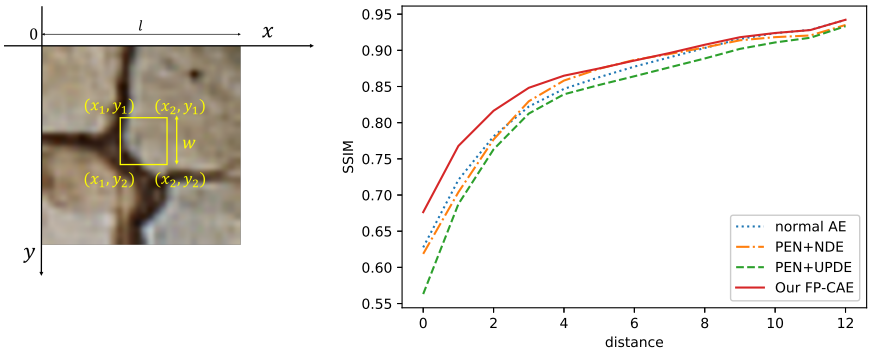
4.1 Stacked input autoencoder

The average testing SSIM score of different models is shown in table 1. It can be seen that our FP-CAE is the best among four models while the model with the commonly used upsampling layers performs the worst. The visual comparison of some test samples can be found in figure 7. The most severe edge artifacts occur in the normal AE. Replacing only normal convolution with partial convolution (*PEN + NDE*) reduces some artifacts but the overall performance drops. (*PEN + UPDE*) reduces most visual artifacts but the reconstruction performance drops a lot and corner artifacts become dominant. When replacing all the normal layers with their partial substitute (*FP-CAE*), the reconstruction performance slightly increases and most artifacts are suppressed.

Although the visible reduction of edge artifacts, the numerical difference between our FP-CAE and normal autoencoder is relatively small. This is because the edges only account for a small proportion for the full image. Improving only the edges and keeping most of the interior unchanged does not lead to significant improvement of the overall SSIM score. In order to evaluate the actual

Table 1: The average SSIM for stacked input autoencoders. Our FP-CAE is better than all the other models. The improvement of SSIM in our model comes from the suppression of edge artifacts.

Model	Normal AE	PEN + NDE	PEN + UPDE	Our FP-CAE
SSIM	0.9366	0.9349	0.9271	**0.9377**



(a) (b)

Fig. 6: Evaluating the effect of suppressing edge artifacts. (a) The definition of distance. (b) Visualization of local SSIM with cropping window size 8 with respect to distance from edge.

improvement of the partial deconvolution on the edges, we calculate the SSIM score in local regions and plotted the SSIM score with respect to the distance to the edge of the patch. The distance is the Manhattan distance: Suppose the width and height of the image is l, the small window size applied to crop the image is w. With x and y the spatial coordinates of a pixel according to an image patch, the coordinates of the four corners on the cropped image are (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2). The distance of this cropped image to the edge is defined as:

$$d = min(x_1, l - x_2) + min(y_1, l - y_2) \tag{6}$$

The smaller the distance is, the closer the cropped image to the four corners. For the locations with the same distance, the SSIM is averaged.

From the graph in figure 6 it can be seen that our fully partial convolutional autoencoder always outperforms the other models, i.e. our FP-CAE not only reduces edge artifacts, the overall performances increases too. As the difference between *PEN + NDE* and *normal AE* is very small, we conclude that the biggest performance increase is due to our proposed deconvolution layers. The reconstruction of *PEN + UPDE* in four corners ($d = 0$) is the worst among all models, which is consistent with the visualization in figure 7. This result clearly

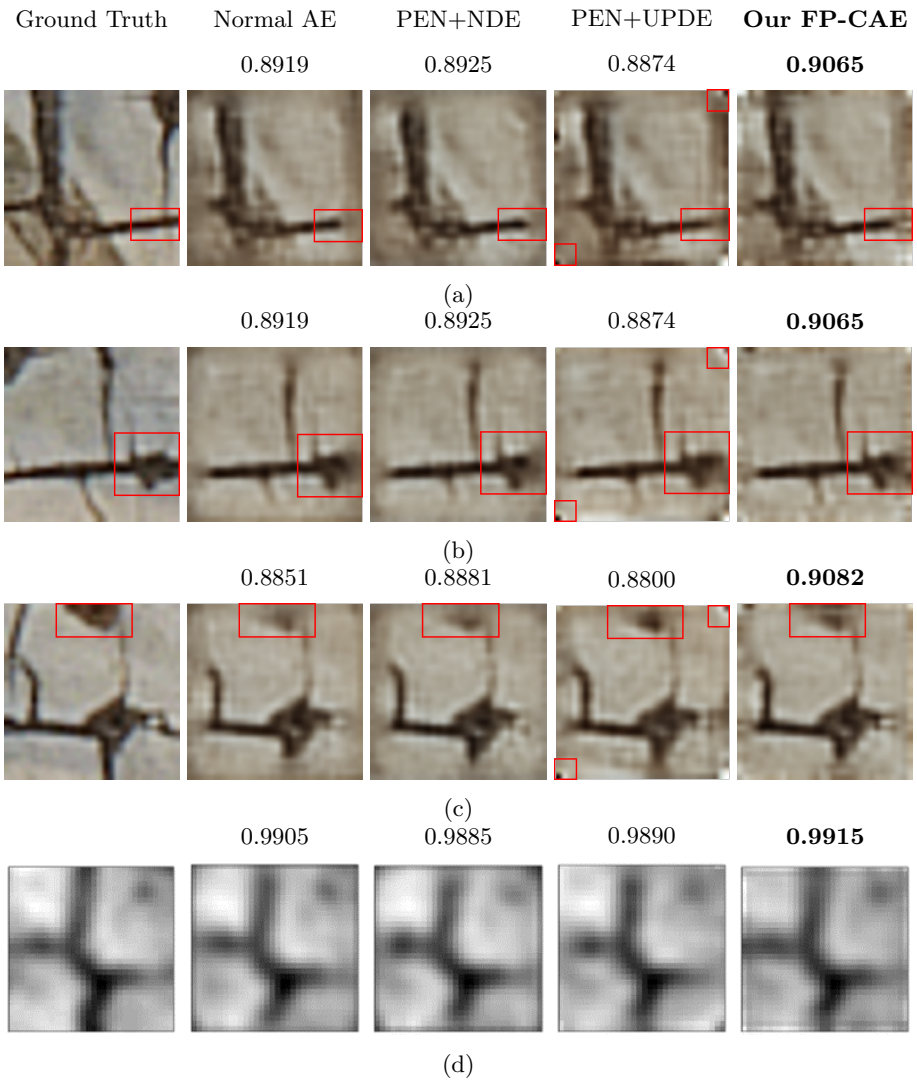


Fig. 7: Visualization of some test images. The best reconstruction SSIM score is in black. The first column is the ground truth. Images in the second to fifth column are reconstruction from different models. Partial convolution layer (third column) does not help improve edge artifacts while up sampling layer (fourth column) causes severe artifacts on the corner and reduce the overall reconstruction quality. The partial deconvolution layers in our fully partial autoencoder (last column) improve reconstruction on edges hence slightly increase the overall SSIM. Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

Table 2: The average SSIM for separated input autoencoders. Both separate input models are better than the stacked input model but the difference between two separate models is very small.

Model	Stacked input FP-CAE	Concatenation FP-CAE	Addition FP-CAE
SSIM	0.9377	0.9469	0.9450

proves that partial deconvolution improves reconstruction performance on the edges.

4.2 Separated input autoencoder

The average SSIM for the seperated multimodal input models are shown in table 2. Compared the SSIM score with stacked input FP-CAE, both separate autoencoders show significant improvement. Some visualization of testing samples can be found in figure 8. The visualization also suggests a better reconstruction on the edges. However, the difference between concatenation based combination and addition based combination is very small. Concatenation FP-CAE only shows 0.36% improvement with respect to the addition FP-CAE. Given that the concatenation model has more parameters (996,043) than the addition model (893,643), we can not conclude that one method outperforms the other one. More studies will be needed to further investigate different combination strategies.

5 Conclusion

We showed that autoencoders can be a powerful tool for feature extraction as a data preprocessing step in art investigation tasks where annotations are typically very limited. To achieve good feature extraction, the reconstruction performance of the autoencoder is maximized. In this study, we generalized implementation of the partial convolution operations and extended it to partial deconvolution, which becomes the basic building block for our fully partial convolutional autoencoder (FP-CAE). In partial convolution and deconvolution, appropriate scale factor is applied to the normal convolution output to counter balance varying number of valid inputs, thus it can smooth the output and reduce artifacts. Results suggest that our partial deconvolution layers in the decoder significantly reduce the artifacts on the edges while avoiding deteriorating inner regions. This way, the reconstruction performance of our FP-CAE outperforms, both visually and numerically, other autoencoders models with normal layers. During training, we introduced an SSIM based loss function, which is effective to maximize the similarity in structure between the original and reconstructed images. Finally, we showed that the reconstruction performance of autoencoder can be further improved by separating the different modalities in the encoder and decoder and

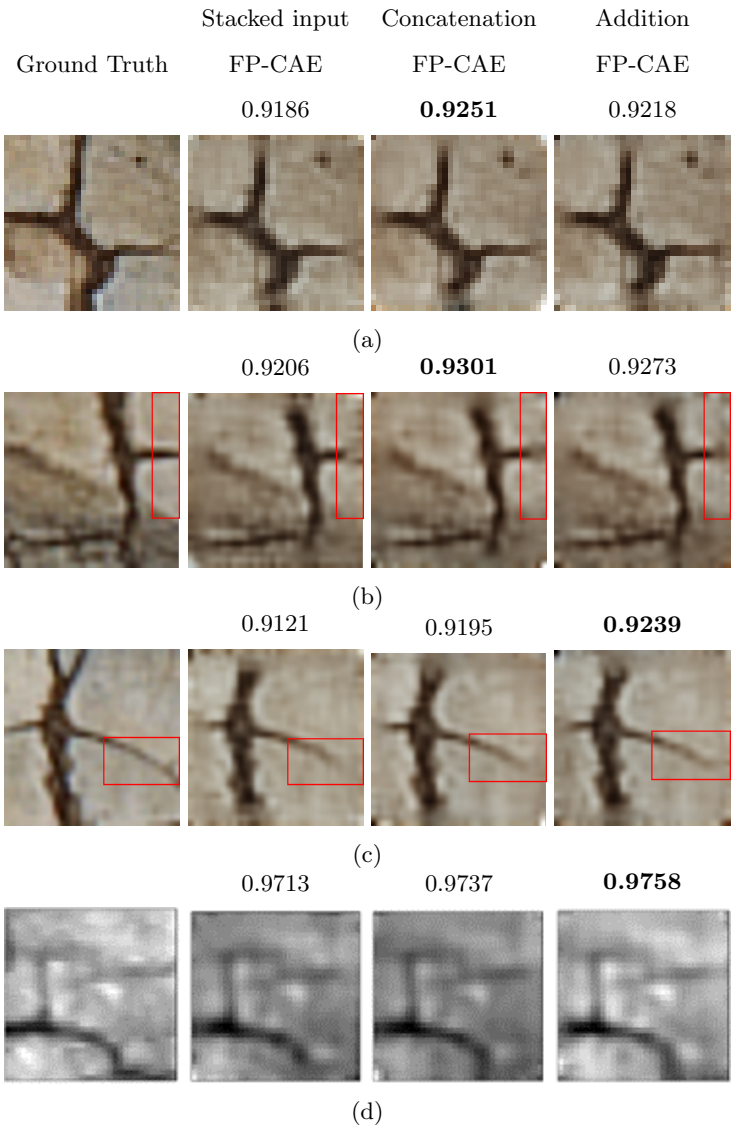


Fig. 8: Visualization of test images from separate input autoencoders, the best is in bold. The SSIM score of both separate input models is better the stacked input model and reconstruction on the edges is improved. The difference between two combination strategies is very small and none of them can always outperforms the other. Image copyright: Ghent, Kathedrale Kerkfabriek, Lukasweb; photo courtesy of KIK-IRPA, Brussels.

combining them in latent space. Results indicate that the performance difference between concatenating and summing the latent vectors is small. More studies

are needed to compare various combination strategies. In future studies the proposed autoencoder FP-CAE can be used in craquelure detection, inpainting, overpainting detection or other art investigation tasks.

References

1. Bigdeli, S.A., Zwicker, M.: Image Restoration using Autoencoding Priors. arXiv:1703.09964 [cs] (Mar 2017), http://arxiv.org/abs/1703.09964, arXiv: 1703.09964
2. Cadena, C., Dick, A., D. Reid, I.: Multi-modal Auto-Encoders as Joint Estimators for Robotics Scene Understanding. In: Robotics: Science and Systems XII. Robotics: Science and Systems Foundation (2016). https://doi.org/10.15607/RSS.2016.XII.041, http://www.roboticsproceedings.org/rss12/p41.pdf
3. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv:1603.07285 [cs, stat] (Mar 2016), http://arxiv.org/abs/1603.07285, arXiv: 1603.07285
4. Feng, F., Wang, X., Li, R.: Cross-modal Retrieval with Correspondence Autoencoder. In: Proceedings of the 22Nd ACM International Conference on Multimedia. pp. 7–16. MM '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2647868.2654902, http://doi.acm.org/10.1145/2647868.2654902, event-place: Orlando, Florida, USA
5. Gondara, L.: Medical image denoising using convolutional denoising autoencoders. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW) pp. 241–246 (Dec 2016). https://doi.org/10.1109/ICDMW.2016.0041, http://arxiv.org/abs/1608.04667, arXiv: 1608.04667
6. Huang, S., Meeus, L., Cornelis, B., Devolder, B., Martens, M., Pizurica, A.: Paint loss detection via kernel sparse representation. In: Image Processing for Art Investigation (IP4AI) : proceedings. pp. 24–26 (2018), https://ip4ai.ugent.be/
7. Innamorati, C., Ritschel, T., Weyrich, T., Mitra, N.J.: Learning on the Edge: Explicit Boundary Handling in CNNs. arXiv:1805.03106 [cs] (May 2018), http://arxiv.org/abs/1805.03106, arXiv: 1805.03106
8. Jaques, N., Taylor, S., Sano, A., Picard, R.: Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In: 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII). pp. 202–208 (Oct 2017). https://doi.org/10.1109/ACII.2017.8273601
9. KIK/IRPA: Closer to van eyck: The ghent altarpiece. http://closertovaneyck.kikirpa.be/ghentaltarpiece/#home/ (2019)
10. Kim, J., Song, S., Yu, S.: Denoising auto-encoder based image enhancement for high resolution sonar image. In: 2017 IEEE Underwater Technology (UT). pp. 1–5 (Feb 2017). https://doi.org/10.1109/UT.2017.7890316
11. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs] (Dec 2014), http://arxiv.org/abs/1412.6980, arXiv: 1412.6980
12. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. arXiv:1312.6114 [cs, stat] (Dec 2013), http://arxiv.org/abs/1312.6114, arXiv: 1312.6114
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)

14. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image Inpainting for Irregular Holes Using Partial Convolutions. arXiv:1804.07723 [cs] (Apr 2018), http://arxiv.org/abs/1804.07723, arXiv: 1804.07723
15. Liu, G., Shih, K.J., Wang, T.C., Reda, F.A., Sapra, K., Yu, Z., Tao, A., Catanzaro, B.: Partial Convolution based Padding. arXiv:1811.11718 [cs] (Nov 2018), http://arxiv.org/abs/1811.11718, arXiv: 1811.11718
16. Meeus, L., Huang, S., Devolder, B., Martens, M., Pizurica, A.: Deep learning for paint loss detection: A case study on the ghent altarpiece. In: Image Processing for Art Investigation (IP4AI). pp. 30–32 (2018), https://www.ip4ai.ugent.be/IP4AI2018_proceedings.pdf
17. Mescheder, L., Nowozin, S., Geiger, A.: Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. pp. 2391–2400. ICML'17, JMLR.org (2017), http://dl.acm.org/citation.cfm?id=3305890.3305928, event-place: Sydney, NSW, Australia
18. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, http://distill.pub/2016/deconv-checkerboard
19. Razavi, A., Oord, A.v.d., Vinyals, O.: Generating Diverse High-Fidelity Images with VQ-VAE-2. arXiv:1906.00446 [cs, stat] (Jun 2019), http://arxiv.org/abs/1906.00446, arXiv: 1906.00446
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 [cs] (May 2015), http://arxiv.org/abs/1505.04597, arXiv: 1505.04597
21. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational Approaches for Auto-Encoding Generative Adversarial Networks. arXiv:1706.04987 [cs, stat] (Jun 2017), http://arxiv.org/abs/1706.04987, arXiv: 1706.04987
22. Shin, H., Orton, M.R., Collins, D.J., Doran, S.J., Leach, M.O.: Stacked Autoencoders for Unsupervised Feature Learning and Multiple Organ Detection in a Pilot Study Using 4d Patient Data. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1930–1943 (Aug 2013). https://doi.org/10.1109/TPAMI.2012.277
23. Sizyakin, R., Cornelis, B., Meeus, L., Martens, M., Voronin, V., Pižurica, A.: A deep learning approach to crack detection in panel paintings p. 3 (2018)
24. Snell, J., Ridgeway, K., Liao, R., Roads, B.D., Mozer, M.C., Zemel, R.S.: Learning to Generate Images with Perceptual Similarity Metrics. arXiv:1511.06409 [cs] (Nov 2015), http://arxiv.org/abs/1511.06409, arXiv: 1511.06409
25. Wang, A., Lu, J., Cai, J., Cham, T., Wang, G.: Large-Margin Multi-Modal Deep Learning for RGB-D Object Recognition. IEEE Transactions on Multimedia **17**(11), 1887–1898 (Nov 2015). https://doi.org/10.1109/TMM.2015.2476655