Masonry Structure Analysis, Completion and Style Transfer Using a Deep Neural Network

Yahya Ibrahim[1,2][0000−0003−0131−985X], Péter Szulovszky[1,2][0000−0002−8792−9634], and Csaba Benedek[1,2][0000−0003−3203−0741]

[1] Faculty of Information Technology and Bionics, Péter Pázmány Catholic University
[2] Institute for Computer Science and Control (SZTAKI)

Abstract. In this paper we present a novel deep learning-based fully automatic method for masonry wall analysis in digital images. The proposed approach is able to automatically detect and virtually complete occluded or damaged wall regions, it segments brick and mortar areas leading to an accurate model of the wall structure, and it can also perform wall-to-wall style transfer as well. Our method involves numerous sequential phases. Initially, a U-Net-based network is used to segment the wall images into brick, mortar, and occluded/damaged regions. Thereafter the hidden wall regions are predicted by a two-stage adversarial inpainting model: first a schematic mortar-brick pattern is predicted, then the second network component adds color information to these areas, providing a realistic visual experience to the observer. Next, a watershed transform-based segmentation step produces accurate outlines of individual bricks in both the visible and the inpainted wall segments. Furthermore, we show that the second adversarial network can also be used for texture transfer: one can change the texture style of a given wall image, based on another wall image, and we can artificially color a schematic wall sketch map, based on the style of a sample wall image. Experiments revealed that the proposed method produces realistic results for various masonry wall types in terms of inpainting the occluded regions or style transfer.

Keywords: Masonry wall · Segmentation · Inpainting · Style transfer

1 Introduction

Ancient heritage sites have always been challenging to study, maintain, and model due to various destructed and altering regions produced by human interventions or natural forces over time, which should be carefully documented. Reassembling these destroyed sites necessitates a comprehensive understanding of the studied archaeological location's global basic structure.

This paper focuses on the investigation and examination of digital images taken from masonry walls, which are among the most fundamental and major components of ancient sites. Large historic walls frequently contain many ruined segments, while after the reconstruction process, the restored areas should follow the main construction patterns of the originally visible parts of the wall. When relying on archive wall images as a basis of reconstruction, it is also a common problem that some regions are hidden/occluded by covering items or other objects in the reference photos of the wall. Therefore the wall's original structural pattern is not visible in certain parts, thus it should be artificially synthesised during reconstruction or in visualization.

Archaeologists investigating masonry walls usually begin their analysis by separating and classifying the wall materials as either main components *(bricks)* or joints *(mortars)*. Next the observed wall structure should be understood and modelled, thereafter one can estimate the original layout of brick and mortar components in areas that have been damaged, discolored, or got covered by other items. Moreover, many historical documents of ancient walls are hand-drawn including only the structure information. When presenting a possible reconstruction view of these walls, archaeologists synthesize a texture style to these sketch drawings based on the style of other preserved ancient walls made from the same material.

In the recent years, we have observed a substantial increase of using machine learning and deep learning algorithms for cultural heritage applications, such as semantic segmentation for historical architectural elements [15] or classification of various segments of monuments [19].

In this paper, we propose a fully automatic end-to-end deep learning-based algorithm for masonry wall image analysis, virtual structure recovery and wall-to-wall style transfer. The proposed approach can be applied in different scenarios: First, given as input a wall image that is partially hidden by numerous irregular items, or contains some ruined/damaged regions, our algorithm detects these outlier areas, it predicts the brick-mortar pattern and wall color texture in these regions, and it extracts correct brick outlines, giving a meaningful structural representation for both the initially visible and the artificially inpainted sections of the wall (see Fig. 3).

The second scenario deals with style transfer and artificial coloring of schematic wall sketch maps. In particular, one can replace the coloring style (e.g. color and texture pattern of the bricks) of a wall image, with another wall's style, while maintaining the wall's original structural integrity. Given two images as inputs: a *content image* which is a color wall image or a binary image for the wall structure, and a *style image* which is a different wall image, the goal is to create a new image that incorporates both the structure of the *content image* and the texture style of the *style image* (see Fig. 4 and Fig. 5).

For both use-cases, our proposed algorithms are freely available for testing on our laboratory's public website[3].

Our contributions can be summarized as follows:

1. An end-to-end algorithm is proposed for inpainting and segmenting masonry wall images, as well as for style transfer from a given wall photo to another wall sample.

[3] http://imgproc.mplab.sztaki.hu/masonrydemo.

2. We provide a comprehensive qualitative evaluation to demonstrate the benefits of our approach in masonry wall inpainting and wall-to-wall style transfer.
3. We present a new web-based application, which can be used by the research community for widely testing and validating our proposed approach.

The initial steps of the presented approach have been described in our prior papers [4–6], which methods we extend here with further steps and real world application examples. More specifically, [4] focused purely on the brick segmentation step, and [6] described an initial occlusion detection and inpainting model that was verified mostly on synthetic data samples. The previously mentioned steps have been first used together and evaluated on real data in [5]. In the present paper, we introduce an extended model, additional experiments and novel use case options for wall image analysis with the proposed approach. In particular, we focus on new style transfer and sketch coloring applications that can widely help architects and archaeologists in many applications such as modeling or restoration.

The article is organized as follows: Sec. 2 reviews recent works related to state-of-the-art wall image delineation, image inpainting and style transfer algorithms; Sec. 3 demonstrates how we produce and augment our dataset; Sec. 4 explains our proposed model in detail; Sec. 5 shows the experimental results; Sec. 6 summarizes our results.

2 Related Work

In this section, we highlight three essential challenges linked to the discussed research topic, and provide a brief summary of recent image segmentation methods utilized for masonry wall delineation. Following that, we give an overview on the state-of-the-art in image inpainting techniques, focusing on their suitability for the task of completing the wall images. Finally, we discuss recent research on style transfer methods.

Several wall image delineation techniques are presented in the literature for various wall types. Many of them are working on 2D images of walls, relying on various image features and modeling approaches, such as a combination of object-oriented and pixel-based image processing technology [3], a color-based automated algorithm based on an improved marker-controlled watershed transform [16], a Hough transform-based delineation algorithm [14], a machine learning-based algorithm [9], and a deep neural-based network for stone-by-stone segmentation [7]. As for using 3D point clouds instead of images, [20] used a Continuous Wavelet Transform (CWT) applied on a 2.5D depth map to obtain the outlines of the bricks. Alternatively, [1] used different 3D local and global features for the wall segmentation. Nevertheless, none of the approaches mentioned have been tested on a wide range of wall image structures. On the contrary, in our research project we focused on involving a wide range of wall types and texture models.

Image inpainting is the process of filling in damaged, noisy, or missing areas of an image to create a completed image. Large datasets are often required to train and learn the hidden feature representation of the images, especially in the recent deep learning models [13, 21, 22]. GMCNN [21] is a generative multi-column convolutional neural network model which deals with global texture features and local details simultaneously, containing a multi-output model [22] which produces numerous image inpainting solutions with two GAN-based parallel paths. The recent EdgeConnect [13] method employs a two-stage adversarial network that fills in the gaps of an edge map created in the first stage. This algorithm is based either on the Canny detector or on the Holistically-Nested Edge Detection algorithm (HED) for initial edge information. However, neither of these two edge generators can provide semantic structural information about the masonry walls because their performance is sensitive to the internal edges inside the brick or mortar regions, resulting in a number of false edges that are unrelated to the wall structure pattern. In contrast, our solution uses in the first step a deep neural network to separate the wall components (brick, and mortar), allowing for a better observation of the entire wall structure in the following steps.

Style transfer approaches combine two images (a *content image* and a *style image*) so that the resulting output image keeps the content image's fundamental elements while appearing to be *painted* in the style of the reference *style image*. Li and Wand proposed a patch-based non-parametric Markov Random Field (MRF) approach [10], which has been proved to be effective in preserving coherent textures in complex images, however it worked less efficiently with non-textured styles. [2] combines independent network components to learn the corresponding content and style information, and uses a *StyleBank* layer which comprises mid-level convolutional filters to individually learn different styles, where style is coupled to a set of parameters. This algorithm provides the advantage of learning a new style as well as a flexible control over style fusion. However, it has a number of weaknesses, including the lack of details in the result images. Li et al. [11] attempt to use a sequence of feature transformations to transfer arbitrary styles without the need for style learning. However, when the used feature vector has a large dimension, the expensive matrix calculation in whitening and colouring modifications becomes a limitation. In contrast to the existing state-of-the-art algorithms, our solution uses a preliminary phase to extract the *content wall image*'s dominant structure, then we add the target wall image style to the model in a direct manner, as it will be detailed later in Sec. 4.2.

3 Dataset Generation

In our project, we created a unique dataset[4] with images of facades and masonry walls from both old and new buildings for training and evaluating the proposed algorithm. Among the collected images, there are examples for three forms of rubble masonry (Random, Square, and Dry), as well as two types of ashlar (Fine, Rough).

[4] The wall images and the manually annotated brick-mortar delineation masks used in the paper are available: http://mplab.sztaki.hu/geocomp/masonryWallAnalysis.

Our dataset consists of 532 different 512×512 wall images separated into a training set (310 images) and a test set (222 images). The training dataset is made up of occlusion-free wall images with a manually segmented binary brick-mortar mask I_{wall_ftr} for each image. Synthetic objects are used to create occlusions for training phase during the data augmentation process which various perturbation steps, including (a) horizontal flip, (b) vertical flip, (c) both vertical and horizontal flips, (d) adding Gaussian noise (e) randomly increasing the average brightness, (f) randomly decreasing the average brightness, and (g) adding random shadows.

4 Proposed Approach

The proposed method's major use-cases are twofold (i) a comprehensive *wall analysis* usage which includes extracting individual brick instances from masonry wall images automatically, detecting occluded/damaged wall segments and filling in their regions with a realistic brick-mortar pattern prediction, and (ii) *a style transfer* usage between different wall images. The following sections present the implementations of these two functions in details.

4.1 Wall Analysis

Our complete masonry wall analysis workflow is presented in Fig. 1. The input of the method should be a masonry wall image I_{in} that can be partially hidden by one or several foreground items, or may contain damaged regions. Our method generates a color image of the reconstructed wall I_{G2_out}, as well as it provides the contours of segmented individual bricks I_{out}. The proposed algorithm is divided into three stages.

Pre-Processing Stage: The initial stage has two objectives: obtaining the delineation structure of the wall by separating the bricks from the mortar regions, and detecting the masks of potentially occluded or damaged wall parts that should be inpainted in the subsequent steps. A U-Net [18] network preforms this segmentation task, yielding a three-class segmented image $I_{wall\_ftr\_oclud}$ as its output, with class labels representing brick (white), mortar (black), and occluded (gray) parts. Then we construct two binary auxiliary images from the output I_{u_out}: a binary mask I_{mask} of the predicted occluded pixels, and a mask of the anticipated bricks I_{ftr_mskd} in the observed wall sections, where both the previously detected occluded and brick pixels have white labels, whilst the mortar pixels are black. The Adam optimizer is used to train the network over a joint loss that includes a cross-entropy loss and a feature-matching loss.

Inpainting Stage: There are two primary sub-steps in this stage: First, based on the mortar pattern detected in the I_{ftr_mskd} mask, the Hidden Feature Generator completes the mortar pixels in the area covered by the white parts of the

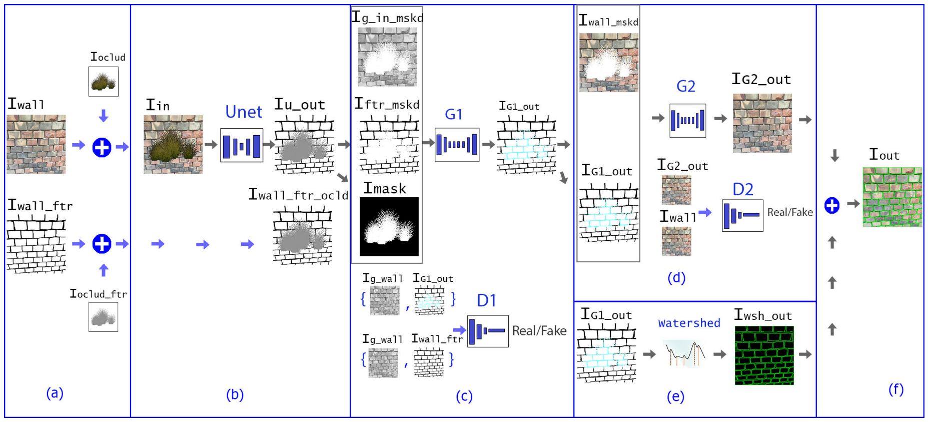


Fig. 1: Dataflow of our algorithm. (**a**) Data augmentation by adding synthetic occlusion (**b**) Pre-processing Stage (U-Net network). (**c**) Inpainting Stage - Hidden Feature Generator (GAN-based network $G1$).(**d**) Inpainting Stage - Image Completion (GAN-based network $G2$). (**e**) Segmentation Stage (Watershed Transform). (**f**) Brick segmentation map superimposed to the inpainted image output ($\mathrm{I}_{\mathrm{out}}$).

predicted $\mathrm{I}_{\mathrm{mask}}$ image. Second, the Image Completion step predicts the RGB color values of the pixels marked as occluded in the $\mathrm{I}_{\mathrm{mask}}$ image, depending on the color information in the non-occluded regions of the $\mathrm{I}_{\mathrm{wall\_mskd}}$, and the structural wall features extracted from the $\mathrm{I}_{\mathrm{G1\_out}}$ map in the occluded regions. The inpainting stage uses two separate generative adversarial networks (GANs), each of them has generator and discriminator components [8, 13].

Hidden Feature Generator: Three images are used as inputs of the first generator (G_1): the output images of the two pre-processing stages ($\mathrm{I}_{\mathrm{mask}}$, $\mathrm{I}_{\mathrm{ftr\_mskd}}$), and $\mathrm{I}_{\mathrm{g\_in\_mskd}}$, which is the masked grayscale version of the input image. The output $\mathrm{I}_{\mathrm{G1\_out}}$ represents the predicted wall structure, completing $\mathrm{I}_{\mathrm{ftr\_mskd}}$ with brick outlines under the occlusion mask regions.

The adversarial loss and the feature-matching loss are used to train the network. The adversarial loss is modeled as a two-player zero-sum game between the Generator G_1 and Discriminator D_1 networks, so that the generator attempts to minimize the loss and the discriminator attempts to maximize it.

Image Completion: The input of the second generator G_2 combines both the masked wall image ($\mathrm{I}_{\mathrm{wall\_mskd}}$) and the output of the first generator ($\mathrm{I}_{\mathrm{G1\_out}}$). The aim of the G_2 generator is creating an output image $\mathrm{I}_{\mathrm{G2\_out}}$ which is filled with predicted color information in the masked regions (see Figure 1d). The inputs of the discriminator D_2 are $\mathrm{I}_{\mathrm{G2\_out}}$ and $\mathrm{I}_{\mathrm{wall}}$, while its goal is to predict whether the color pattern is true or not. The network is trained over a joint loss that consists of L_1 loss, adversarial loss, perceptual loss, and style loss.

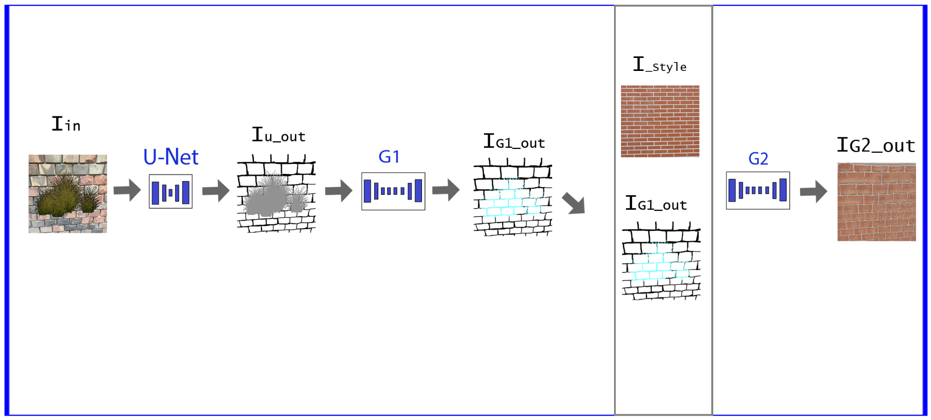


Fig. 2: Dataflow of the Style Transfer application

Segmentation Stag: This stage of the proposed algorithm extracts the accurate outlines of the brick instances, using the previously obtained Hidden Feature Generator delineation map $\mathrm{I}_{\mathrm{G1\_out}}$. Although the separation maps are notably precise, they may be noisy at brick boundaries, and we can observe that some bricks are touching, making Connected Component Analysis CCA-based separation susceptible to errors. Therefore, we use a marker-based watershed [17] segmentation for robust brick separation to overcome these issues.

First, we calculate the obtained delineation map's ($\mathrm{I}_{\mathrm{G1\_out}}$) inverse distance transform (IDT) image, then we apply the H–minima transform, which suppresses all minima under a given H-value since the IDT image may have numerous false local minima, we start flooding using the H–minima regions as a starting points (internal marker of the watershed algorithm), and take the inverse of $\mathrm{I}_{\mathrm{G1\_out}}$ (i.e., all mortar or non-wall pixels) as an external marker map, whose pixels are not compatible with any bricks. Finally, the obtained brick shapes can be displayed over the Image Completion output (see $\mathrm{I}_{\mathrm{out}}$).

4.2 Style Transfer

The goal of the style transfer component is to fill or modify the wall's texture style based on an image of another wall or wall segment. The workflow of this procedure is presented in Fig. 2. The style of the input image I_{in} is changed to match the style of another wall image sample called the *style image* $\mathrm{I}_{\mathrm{style}}$. Archaeologists can use this algorithm to modify some degraded segments of the studied wall that have become soiled or lost its original color over time due to environmental factors, relying in various style features extracted from the intact wall regions.

The proposed style transfer procedure uses the same G_1 and G_2 network components, that were previously introduced in the wall analysis section (Sec. 4.1). Here the *image completion* network G_2 has two inputs: the first one is the G_1

generator's output $\mathrm{I}_{\mathrm{G1\_out}}$, which is a brick-mortar separation map that includes the predicted complete wall structure for the processed wall image I_{in} (including both the originally visible and occluded wall components). The second input is the style image $\mathrm{I}_{\mathrm{style}}$, which is represented by an occlusion-free colored wall image with a different texture style. The network extracts a representation for the texture style of $\mathrm{I}_{\mathrm{style}}$ and transfers it to the predicted brick-mortar separation map $\mathrm{I}_{\mathrm{G1\_out}}$, resulting in a new image $\mathrm{I}_{\mathrm{G2\_out}}$, that is a color image reflecting the structure of I_{in} and the style of $\mathrm{I}_{\mathrm{style}}$.

Note that apart from changing the style of existing wall images (Fig. 4), we can also directly feed in a binary brick-mortar map to the network for applications where we initially have a binary sketch of the wall only, and intend to paint it using the style from another wall image (see Fig. 5).

To use the same trained network as in the previous application and to avoid any additional training for the G_2 generator, which is trained to paint exclusively the *occluded* (i.e. masked) region of its input image (see Sec.4.1 and Fig. 1), we add here virtual masks to the style images. Two binary virtual masks - which are inverted variants of each other - ensure that the entire target image will be filled with the texture of the style image, so that the style image is fed twice to the network, each time with one of the masks.

5 Experiments

In this section, we analyze the performance of the proposed approach on the test data.

We begin with the discussion of the *wall analysis* application, where the results of applying our algorithm in real-world scenarios are presented. Fig. 3 shows some qualitative results where the first row presents wall images with occluding objects (human-made items or plants), the second row shows the output of the pre-processing stage (U-net network) presented as a three-color images (gray: the occluding objects, black: the mortar, white: the bricks), the third row illustrates the G_1 output where the predicted mortar lines are in blue, the fourth row shows the results of the color inpainting algorithm, and the last row presents the results of the brick segmentation step where brick instances are separated by a green lines.

The results confirm that our technique is effective in recognizing the occluded wall regions in the images: we can see that the U-net network can accurately extract occluded image parts and nearly all occluding objects are properly detected. The results of the "Hidden Feature Generator" phase indicate our algorithm's efficiency in predicting the mortar lines in occluded regions. By examining the inpainted color images we can confirm that our algorithm manages to create realistic wall patterns in the occluded areas. Moreover, despite the wide variety of brick component sizes in the images, and image quality issues such as the poor contrast of the second sample, the output of the brick segmentation stage has a high quality.

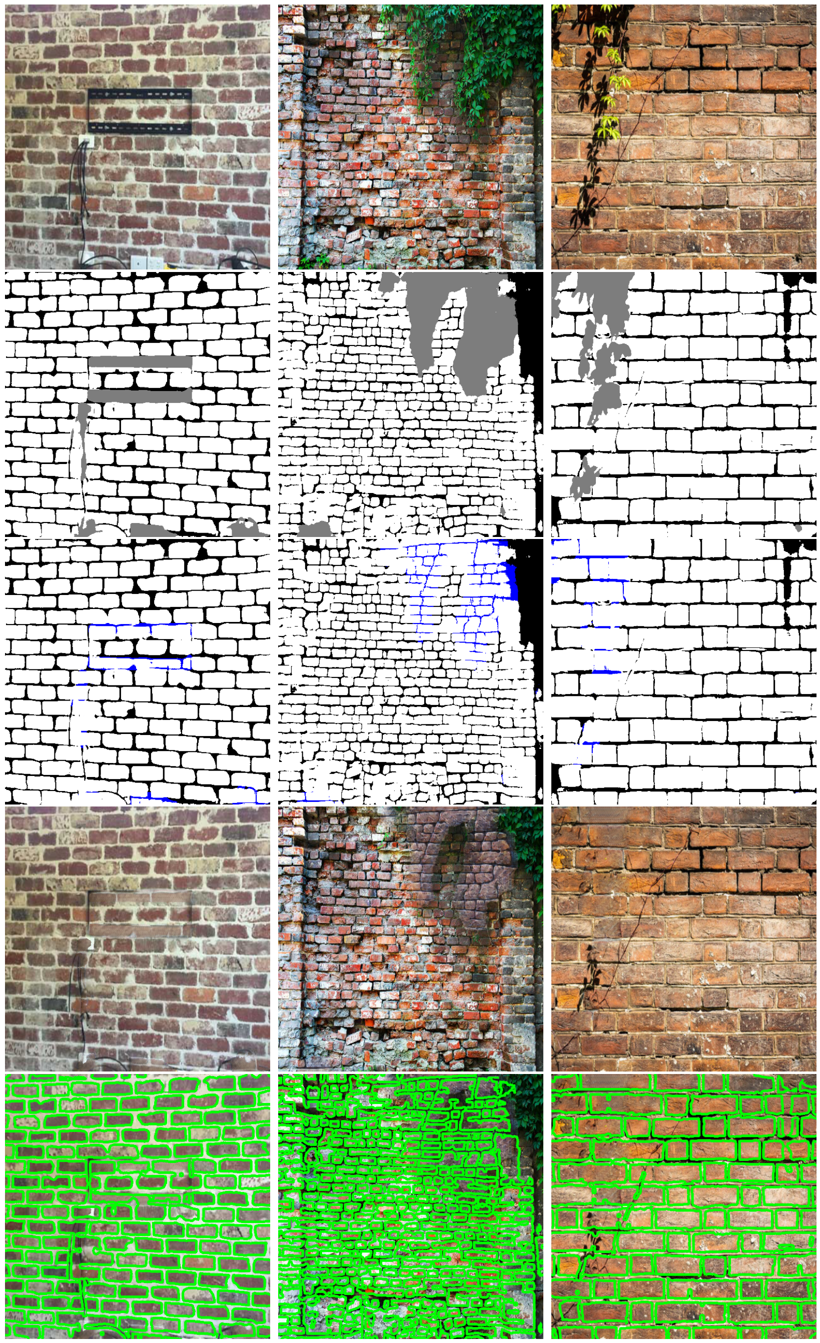


Fig. 3: Results for wall analysis application step by step for three real-world samples, first row: input images, second row: Pre-Processing Stage output, third row: G_1 output, fourth row: G_2 output, last row: Segmentation Stage output.

Input images

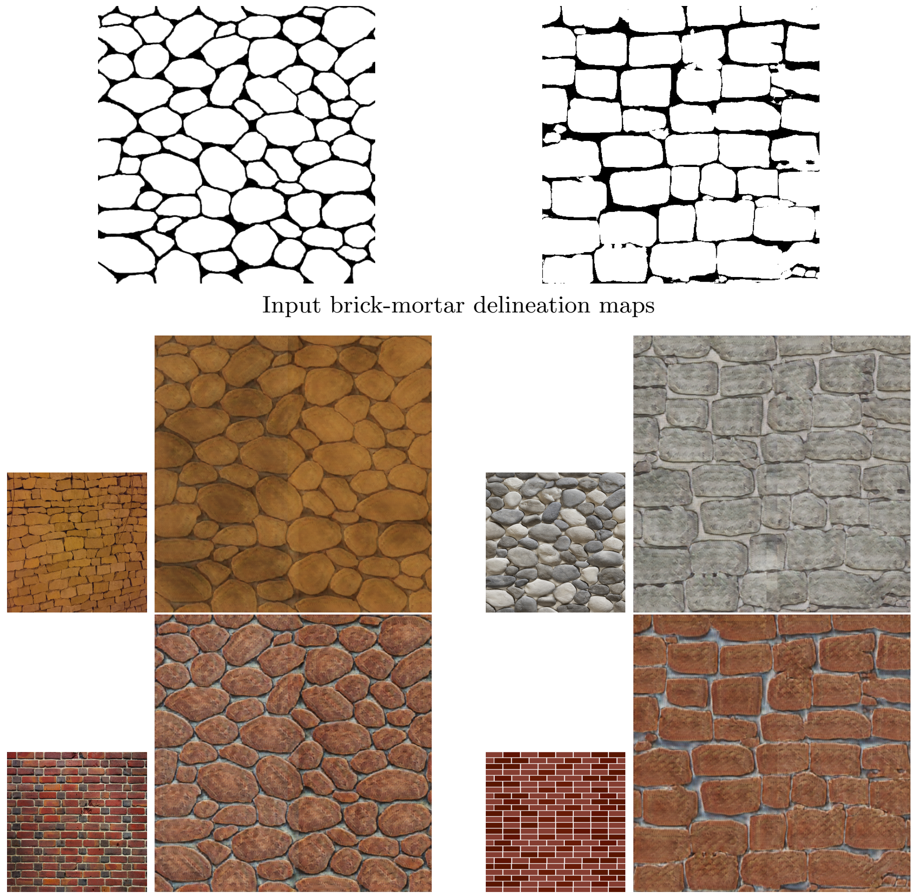
(a) Style (b) Our output (c) Style (d) Our output

Fig. 4: Wall to wall style transfer samples, first row shows the input images, second and third rows represent the style image and our output side by side.

In Fig. 4, we show results for the *style transfer* application, where we present two different *content image* samples in the first row, and each of them is transformed in the process using two different *style images*. (The style images and the corresponding style transfer results are displayed side by side in the second and third rows.)

Next, We present the results of transforming a binary brick-mortar map to a wall image. Fig. 5 displays two brick-mortar sketch maps, which can be drawn even by hand. Each of them is transformed to different colored wall images using two distinct *style images*, and the results are shown in the last two rows.

We can observe in both Fig 4 and 5 how the algorithm manged to paint the style texture of the style image onto the brick-mortar pattern of the processed image or the handrawn sketch map, so that the mortar regions efficiently match



Input brick-mortar delineation maps

(a) Style (b) Our output (c) Style (d) Our output

Fig. 5: brick-mortar map to a wall style transfer samples.

the color of the mortar, and the appearances of the brick regions match the brick texture of the style image. However there are a few error cases when some neighboring bricks are merged into a large brick object in the image as shown in the center part of Fig. 4(d).

Since in image inpainting tasks there are usually many alternative solutions, there is no clear numerical metric for evaluating image inpainting [12] and style transfer results. Therefore, similarly to our user survey regarding the image inpainting model in [5], we conducted a user survey to assess our proposed style transfer approach. Using 15 different style images and 15 content images (9 color wall images and 6 brick-mortar maps), we generated 15 stylized images by our network. We showed the outputs to 28 test subjects displaying the images side-by-side with the style images, and asked them to vote whether they find the result of our style transfer network visually appropriate or not. 18 out of

28 participants said that more than 66% of the seen images were transformed appropriately; 10 out of 15 images received more than 66% votes that it has been transferred appropriately. By summarizing all votes on all images, 70.47% (296/420) images were considered being transferred correctly.

Based on the above-mentioned study results, we can conclude that our algorithm provides high quality inpainting outputs for real-world wall images, when we need to expect that real occluding objects are present. We also get efficient segmentation results both in the originally visible and in the inpainted segments of the walls. Furthermore, the algorithm is capable of transferring the style between different wall structures and styles resulting a realistic image outputs.

6 Conclusion

This research presented a new approach for wall image processing, analysis and style transfer from one wall to another. Our network has two different ways of utilization: in the first one, the algorithm recognizes occluded or damaged wall portions and inpaints the corrupted segment with a realistic wall pattern. Moreover, for the inpainted wall image, the technique returns an instance level brick segmentation output. The second implementation able to transfer the texture and color style from an image to the another wall structure. Our approach is divided into three steps. A U-Net-based module is used in the initial preprocessing stage to separate the brick, mortar, and occluded areas of the input image. This preliminary segmentation is fed into the inpainting stage, which is comprises two GAN-based networks: the first one is responsible for the wall structure compilation, while the second is in charge of color image inpainting. The final stage employs the watershed algorithm, which ensures that the entire wall is accurately segmented. The algorithm's second application involves altering the second GAN's inputs to fit two different wall images. Numerous qualitative experiments have shown that the proposed approach is significantly robust against various artifacts in real-world applications regarding the specified challenges.

Acknowledgement

The authors acknowledge the support of the Hungarian National Research, Development and Innovation Office via the project TKP2021-NVA-01, and the Artificial Intelligence National Laboratory (MILAB) program.

References

1. Bosché, F., Valero, E., Forster, A., Wilson, L., Leslie, A.: Evaluation of historic masonry substrates: towards greater objectivity and efficiency (06 2016). https://doi.org/10.4324/9781315628011-8

2. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2770–2779 (2017). https://doi.org/10.1109/CVPR.2017.296
3. Hemmleb, M., Weritz A, F., Schiemenz B, A., Grote C, A., Maierhofer, C.: Multispectral data acquisition and processing techniques for damage detection on building surfaces. In: ISPRS Commission V Symposium. pp. 1–6 (1 2006)
4. Ibrahim, Y., Nagy, B., Benedek, C.: CNN-based watershed marker extraction for brick segmentation in masonry walls. In: Int'l Conf. Image Analysis and Recognition. pp. 332–344 (2019)
5. Ibrahim, Y., Nagy, B., Benedek, C.: Deep learning-based masonry wall image analysis. Remote Sensing **12**(23) (2020). https://doi.org/10.3390/rs12233918, https://www.mdpi.com/2072-4292/12/23/3918
6. Ibrahim, Y., Nagy, B., Benedek, C.: A gan-based blind inpainting method for masonry wall images. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 3178–3185 (2021). https://doi.org/10.1109/ICPR48806.2021.9413009
7. Idjaton, K., Desquesnes, X., Treuillet, S., Brunetaud, X.: Stone-by-stone segmentation for monitoring large historical monuments using deep neural networks. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (eds.) Pattern Recognition. ICPR International Workshops and Challenges. pp. 235–248. Springer International Publishing, Cham (2021)
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (2016)
9. Kajatin, R., Nalpantidis, L.: Image segmentation of bricks in masonry wall using a fusion of machine learning algorithms. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (eds.) Pattern Recognition. ICPR International Workshops and Challenges. pp. 446–461. Springer International Publishing, Cham (2021)
10. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 702–716. Springer International Publishing, Cham (2016)
11. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/49182f81e6a13cf5eaa496d51fea6406-Paper.pdf
12. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: European Conference on Computer Vision (ECCV) (2018)
13. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. In: Int'l Conf. on Computer Vision Workshop (ICCVW). pp. 3265–3274 (2019)
14. Oses, N., Dornaika, F., Moujahid, A.: Image-based delineation and classification of built heritage masonry. Remote Sensing **6**(3), 1863–1889 (Feb 2014). https://doi.org/10.3390/rs6031863, http://dx.doi.org/10.3390/rs6031863
15. Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E., Lingua, A.M.: Point cloud semantic segmentation using a deep

learning framework for cultural heritage. Remote Sensing **12**(6), 1005 (Mar 2020). https://doi.org/10.3390/rs12061005, http://dx.doi.org/10.3390/rs12061005
16. Riveiro, B., Conde, B., Gonzalez, H., Arias, P., Caamaño, J.: Automatic creation of structural models from point cloud data: the case of masonry structures. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences **II-3/W5**, 3–9 (8 2015). https://doi.org/10.5194/isprsannals-II-3-W5-3-2015
17. Roerdink, J.B., Meijster, A.: The Watershed transform: Definitions, algorithms and parallelization strategies. Fundam. Inf. **41**(1,2), 187–228 (Apr 2000), http://dl.acm.org/citation.cfm?id=2372488.2372495
18. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. LNCS, vol. 9351, pp. 234–241 (2015)
19. Teruggi, S., Grilli, E., Russo, M., Fassi, F., Remondino, F.: A hierarchical machine learning approach for multi-level and multi-resolution 3d point cloud classification. Remote Sensing **12**(16), 2598 (Aug 2020). https://doi.org/10.3390/rs12162598, https://doi.org/10.3390/rs12162598
20. Valero, E., Bosché, F., Forster, A., Hyslop, E.: Historic digital survey: Reality capture and automatic data processing for the interpretation and analysis of historic architectural rubble masonry. In: Aguilar, R., Torrealva, D., Moreira, S., Pando, M.A., Ramos, L.F. (eds.) Structural Analysis of Historical Constructions. pp. 388–396. Springer International Publishing, Cham (2019)
21. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. CoRR **abs/1810.08771** (2018), http://arxiv.org/abs/1810.08771
22. Zheng, C., Cham, T., Cai, J.: Pluralistic image completion. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1438–1447 (2019)