



3GAN: A Three-GAN-based Approach for Image Inpainting Applied to the Reconstruction of Occluded Parts of Building Walls

Benedikt Kottler¹, Ludwig List¹, Dimitri Bulatov¹ ^a and Martin Weinmann² ^b

¹*Fraunhofer Institute for Optronics, System Technologies and Image Exploitation (IOSB),
Gutleuthausstrasse 1, 76275 Ettlingen, Germany*

²*Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT),
Englerstr. 7, 76131 Karlsruhe, Germany*


Keywords: Edges, Façades, GAN, Inpainting, Semantic Segmentation, Texture Synthesis.


Abstract: Realistic representation of building walls from images is an important aspect of scene understanding and has many applications. Often, images of buildings are the only input for texturing 3D models, and these images may be occluded by vegetation. One task of image inpainting is to remove these clutter objects. Since the disturbing objects can also be of a larger scale, modern deep learning techniques should be applied to replace them as realistically and context-aware as possible. To support an inpainting network, it is useful to include a-priori information. An example of a network that considers edge images is the two-stage GAN model denoted as EdgeConnect. This idea is taken up in this work and further developed to a three-stage GAN (3GAN) model for façade images by additionally incorporating semantic label images. By inpainting the label images, not only a clear geometric structure but also class information, like position and shape of windows and their typical color distribution, are provided to the model. This model is compared qualitatively and quantitatively with the conventional version of EdgeConnect and another well-known deep-learning-based approach on inpainting which is based on partial convolutions. This latter approach was outperformed by both GAN-based methods, both qualitatively and quantitatively. While the quantitative evaluation showed that the conventional EdgeConnect method performs minimally best, the proposed method yields a slightly better representation of specific façade elements.

1 INTRODUCTION

Images are probably the most widespread source of information nowadays. However, images cannot always be captured in the way the desired scene appears without irrelevant objects. Image inpainting is a discipline dedicated to removing these objects; the applications of image inpainting extend from restoring images (e.g. images contaminated with noise or scratches) via filling missing image parts (e.g. gaps resulting from text removal or object removal) to filling of image regions (e.g. areas resulting from image cropping or masking), see (Elharrouss et al., 2019). In particular, realistic representations of 3D city scenes, especially building walls from texture images, possibly combined with reconstruction results of different sensor data, are an essential aspect of scene understanding and have many applications (Shalunts et al., 2011; Bulatov et al., 2014; Zhang et al., 2020). Often it happens that the images of building façades are the

only input for texturing. As a consequence, proper occlusion analysis cannot be accomplished. This has the disadvantage that the façade images are often contaminated, especially by trees standing in front of the buildings. Clearly, these objects can be arbitrarily large, and the only feasible way to replace them in the image is to learn the backgrounds from a high number of training examples. In this paper, we wish to investigate to what extent the modern generative techniques of deep learning are capable of replacing such large occluding objects as realistically and context-aware as possible and, at the same time, of maintaining the distinction between rectangular façade objects and the background. The main innovation lies in combining a-priori information, allowing to support an inpainting network operating on such a complex and texture-rich entity as a façade with explicit semantic or geometrical descriptions of its elements. Inspired by the EdgeConnect approach (Nazeri et al., 2019) based on two generative adversarial networks (GANs), we propose a method that successively reconstructs an edge image, a label image, and a texture image using three

^a  <https://orcid.org/0000-0002-0560-2591>

^b  <https://orcid.org/0000-0002-8654-7546>

GANs (see Figure 1, top). This method will be denoted 3GAN approach throughout this paper.

We provide a literature review on image inpainting in Section 2 and identify at the end of this section the most promising method which relies on two GANs. In Section 3, we extend this method by yet another GAN that utilizes two-dimensional context information stored in a semantic segmentation image, expecting additional stability from this intermediate input. In Section 4, we present the results of the proposed method and compare them with those yielded by the predecessor and another algorithm. Finally, Section 5 summarizes the main findings of our work and outlines a few directions for future research.

2 RELATED WORK

Image inpainting is an under-determined inverse problem that does not have a single well-defined solution because anything can theoretically appear behind the occluding object. Therefore, in order to address this problem, it is necessary to introduce a-priori information. Contrary to the early methods, these assumptions were formulated explicitly. Vanishing gradients, for example, allowed to formulate the (structural) inpainting problem as a partial differential equation (Chan and Shen, 2001). Assuming that the image patch to be inpainted has appeared somewhere else in the images incubates the so-called texture-based inpainting approaches (Criminisi et al., 2004). The machine learning and, in particular, deep-learning-based methods offer a systematic framework of considering many training examples to compute the network parameters and include the a-priori information in an implicit way. According to e.g. (Elharrouss et al., 2019), where more details and sources can be found, inpainting methods based on deep learning can be roughly subdivided into two groups: based on GANs (Isola et al., 2017; Yang et al., 2017; Nazeri et al., 2019; Shao et al., 2020) and on pixel-filling predictions (Pathak et al., 2016; Liu et al., 2018; Yu et al., 2019; Li et al., 2020; Pyo et al., 2020). Diving deeper into details, the Context-encoder method of (Pathak et al., 2016) is probably the earliest GAN-based method on inpainting and is based on an encoder-decoder architecture. Because of multiple pooling layers, fine details can hardly be reconstructed. To cope with high-resolution images, (Iizuka et al., 2017) adds dilation convolutional layers. Still, there are some noise patterns that have to be smoothed away by a post-processing routine, such as Poison blending. The work of (Yu et al., 2019) replaces partial convolutions (those affecting the in-

painting region only) with the so-called gated convolutions, where the gating parameters are learnable for each image channel and, optionally, for a user-provided sparse sketch of edges. Recently, the EdgeConnect method was developed (Nazeri et al., 2019), which aims to reconstruct such a good edges sketch automatically. There are two GANs, whereby the first GAN learns to complete the edge image of an RGB image. The edges serve as a-priori information for the second GAN, supposed to reconstruct the color image. Thus, the image structure in the edge image is captured with the first GAN, while the second GAN focuses on details of color image inpainting, such as the homogeneous color content of the regions enclosed by edges.

Finally, several works relying on semantic segmentation can be mentioned (Kottler et al., 2016; Kottler et al., 2020; Song et al., 2018; Liao et al., 2020; Huang et al., 2021). In the first two contributions, the semantic segmentation result is *undamaged* since it stems from an external source. The work of (Song et al., 2018) is a two-GANs-based network. The first GAN, called *Segmentation Prediction*, accomplishes inpainting of the segmentation image as an intermediate step. The second, called *Segmentation Guidance*, reconstructs the texture image. In this method, the segmentation result is a product of data processing and lacks the typical man-made features, such as rectangular structures, which can be observed in the results. In the work of (Liao et al., 2020), the corrupted image is initially completed in the feature space. Inpainting of segmentation and the texture image takes place alternately.

3 METHODOLOGY

3.1 Preliminaries: From EdgeConnect to 3GAN Approach

Edge images of color pictures often do not show a good balance between actual changes in structure and noise. More importantly, the binary edge images serving as input and output of the first GAN of EdgeConnect bear only a one-dimensional manifold on information. The reason is that the rasterized result of an edge detection algorithm is a binary image, where edge pixels have a thickness of one, and thus, the overwhelming majority of pixels in the images do not benefit from this information. If the binary image has formed a closed contour, for instance, around a window, then the second GAN will have learned that the pixels within this contour must

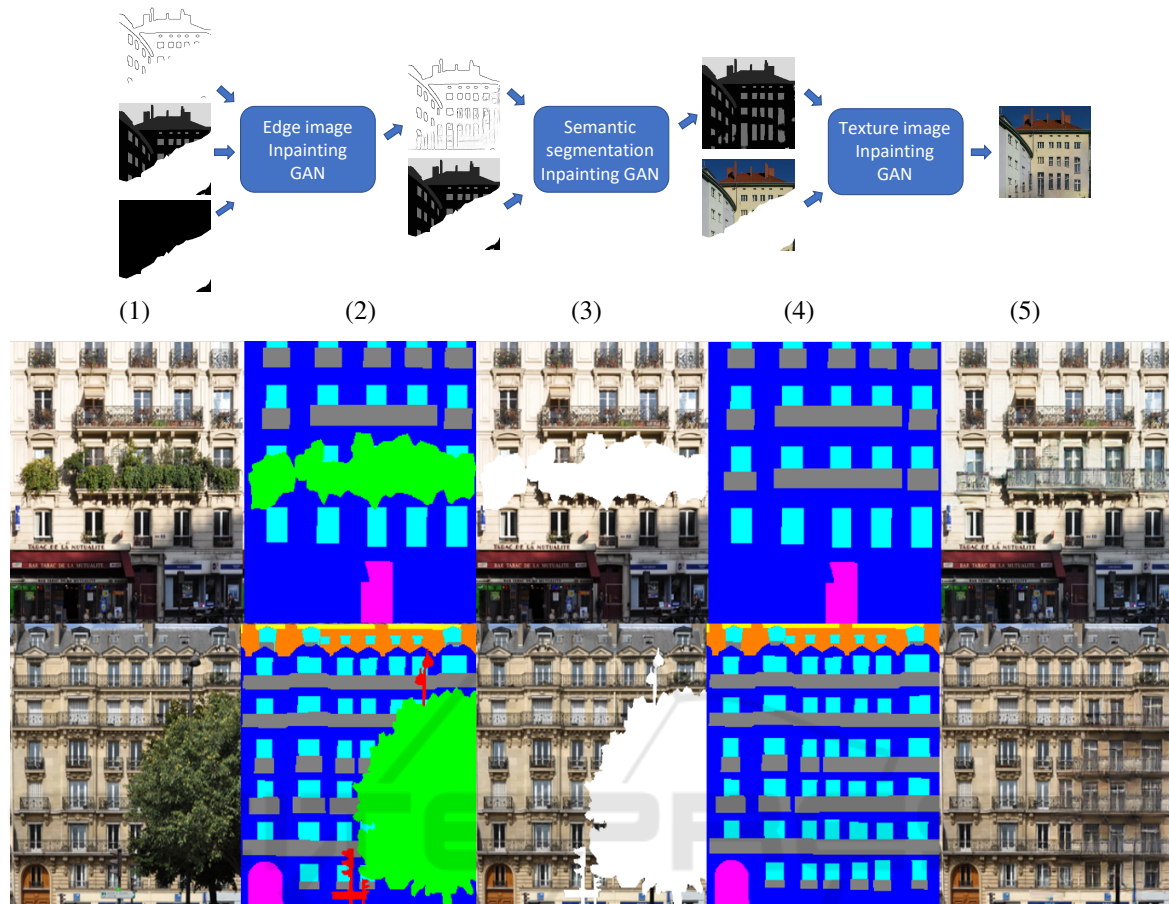


Figure 1: Top: Schematic overview of the proposed 3GAN approach. Bottom: Graphical overview. While the step from (1) to (2) is the result of semantic segmentation and from (2) to (3), foreground class selection, is trivial, we concentrate in this work on automatic inpainting of (2) resulting in (4), followed by inpainting of (1) resulting in (5) using GANs.

have an approximately homogeneous color representation at least locally, since otherwise there would have been further edges detected. However, to know these most representative colors, we should be able to copy the color information from one window to another. In other words, we must know which pixels of the image correspond to windows. There are two implications of this line of thoughts. The first implication is the relevance of the semantic segmentation result as additional input. It is a label image on the more abstracted level than the textured façade picture. It allows the GAN to learn typical representations of different classes, which would be trained separately within the same image or between the different images. Fortunately, there are many freely available datasets with labeled façade elements (Gadde et al., 2016) and many pipelines on semantic segmentation using both model-driven (Wenzel, 2016; Wenzel and Förstner, 2016) and example-driven (Fathalla and Vogiatzis, 2017; Schmitz and Mayer, 2016) methods. Using grammar-driven methods, sometimes even oc-

cluded windows can be recovered within incomplete grids of façade elements. We will assume the availability of accurate semantic segmentation for a fair number of façades in our dataset since they can be obtained either with one of these methods or interactively. The second implication is that the label image may be contaminated by the foreground objects as well. Inpainting such an image is expected to be easy because of a reduced value range to guess (for instance, 1 for wall, 2 for roof, 3 for window, 4 for door, and so on). Therefore, the proposed method will strive at inpainting the labeled image using the standard EdgeConnect method. Besides a small value range, the advantage of inpainting label images is that only sensible edges will be inpainted in the first step of the EdgeConnect algorithm and not those resulting from the data noise. Finally, the label image is used to refill the missed values in the original image.

Summarizing, we suppose in this work that the ground-based, high-resolution digital image of a façade is provided together with the binary mask of

the same size where pixels labeled as zeros and ones correspond to the background and occluding objects, respectively. The image itself is supposed to be a standard RGB pixel grid with integer values. Our last input is the semantic segmentation map which has, of course, the same size as the image, and a predefined labeling map for available semantic classes. The images, rasterized (Canny) edge maps, semantic labels, and masks are denoted, respectively, by capital letters J, C, S and M , if not specified differently. We refer to Subsection 4.1 of the results section for more information on available data, in particular, for training.

3.2 3GAN Approach

According to our findings from the previous section, the workflow of our method is illustrated in Figure 1, top. In the following three subsections, each of the three involved GANs will be presented.

3.2.1 Inpainting of the Edge Image

The first GAN resembles the one proposed by (Nazeri et al., 2019). The edges of the label image S are generated with the Canny edge detector (Canny, 1986), resulting in the edge image C . In order to complete C to \hat{C} , the generator G_1 obtains as input C , S , and M whereby S is parsed as the element-wise product $S \circ (1 - M)$ in order to suppress those regions of label images where the mask is true. The discriminator D_1 estimates whether an input edge image C is an original (training label 1) or an artificially generated (training label 0) one. Thus, the loss function

$$\min_{G_1} \max_{D_1} L_{G_1} = \min_{G_1} \left(\max_{D_1} (L_a) + \lambda_f L_f \right) \quad (1)$$

is used to estimate the parameters of G_1 and D_1 . Hereby, L_a and L_f are the adversarial and feature loss (see below), while $\lambda_f = 10$ is a regularization parameter whose value was chosen according to (Nazeri et al., 2019). The adversarial loss L_a is defined according to (Goodfellow et al., 2014):

$$L_a = E_{C,S} [\log D_1(C,S)] + E_S \log [1 - D_1(\hat{C}, S)]. \quad (2)$$

In this equation, E is the expectation value collected over all training data pairs (C, S) . A good discriminator D_1 outputs values for $D_1(C, S)$ close to 1 and values $D_1(\hat{C}, S)$ close to zero, because $\hat{C} = G_1(C, S, M)$ is an artificial image. Thus D_1 maximizes L_a in (1). A good generator, contrarily, has to produce \hat{C} for which D_1 is close to 1. The logarithm of $1 - D_1$ would yield a large negative number and decrease L_a in (1) strongly. At the beginning of the training process, when G_1 is not good enough to fool D_1 , the second

term in the sum of equation (2) penalizes L_a while with advancing training, the first term is supposed to avoid overfitting since it does not depend on G_1 . The feature matching loss

$$L_f(C, \hat{C}) = E \left[\sum_{i=1}^T \frac{1}{N_i} \left\| D_1^{(i)}(C) - D_1^{(i)}(\hat{C}) \right\|_1 \right] \quad (3)$$

compares the activation maps in the middle layers of the discriminator. Doing so stabilizes the training process by forcing the generator to produce results similar to real images, as we will explain in more detail in Section 3.2.3. In (3), T denotes the number of all convolutional layers of the discriminator, N_i denotes the number of elements in the i -th activation layer from the discriminator, and $D_1^{(i)}$ denotes the activation in the i -th layer of the discriminator.

3.2.2 Inpainting of the Label Image

Based on the input edge image C , we substitute only the occluded parts of it with the generated edges \hat{C} in the masking:

$$\hat{C} = C \circ (1 - M) + \hat{C} \circ M. \quad (4)$$

Within our second GAN, we wish to inpaint the semantic segmentation image S using \hat{C} . The second generator $\hat{S} = G_2(S, \hat{C}, M) = G_2(S \circ (1 - M), \hat{C}, M)$ operates on the set of ground truth images for semantic segmentation and makes predictions to be assessed by D_2 . The loss function is formed similarly to that from (1),

$$\min_{G_2} \max_{D_2} L_{G_2} = \min_{G_2} \left(\max_{D_2} (L_a) + \lambda_{\ell_1} L_{\ell_1} \right), \quad (5)$$

only that $D_2(S, C)$ now assesses S (differently from $D_1(C, S)$ in (2)) and that the second term is the L_1 loss $L_{\ell_1}(\hat{S}, S)$. This loss minimizes the sum of absolute differences between \hat{S} and S . For proper scaling, we normalize this loss by the mask size. Since the generated label image should be as close as possible to the original values of the input label image, the L_1 loss is a suitable tool for training the GAN. As proposed in (Nazeri et al., 2019), the choice for the regularization parameter is $\lambda_{\ell_1} = 10$.

3.2.3 Inpainting of the Texture Image

Analogously to the previous section and equation (4), we update S to become

$$\hat{S} = S \circ (1 - M) + \hat{S} \circ M \quad (6)$$

and use it to inpaint the texture image $\hat{J} = G_3(J, \hat{S}, M) = G_3(J \circ (1 - M), \hat{S}, M)$. The loss function

$$L_{G_3} = L_{a,3} + \lambda_{\ell_1} L_{\ell_1} + \lambda_{perc} L_{perc} + \lambda_{style} L_{style} \quad (7)$$

has now four terms connected with regularization parameters $\lambda_{\ell_1} = 10$, $\lambda_{perc} = 1$ and $\lambda_{style} = 2500$ from (Nazeri et al., 2019). The first two are defined analogously to those in (1), with J instead of C and \hat{S} instead of S . The perceptual loss L_{perc} , originally proposed in (Gatys et al., 2015) and adopted from (Johnson et al., 2016; Nazeri et al., 2019) penalizes results that are not perceptually similar to labels by defining a distance measure between activated features of a pre-trained network. It resembles feature loss in (3), where $D^{(i)}$ s are replaced by the activation maps ϕ_i from layers `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` and `relu5_1` of the VGG-19 network pre-trained on the ImageNet dataset (Russakovsky et al., 2015). The reason not to use a model pre-trained on such a database in (3) is that the VGG-19 network has been trained for different purposes, and thus, the activation maps are not supposed to coincide. Finally, the style loss L_{style} from (Sajjadi et al., 2017) measures the difference between covariance matrices. The authors of EdgeConnect (Nazeri et al., 2019) have identified it as a helpful tool for eliminating checkerboard artifacts usually produced during the deconvolution and upsampling process in encoder-decoder networks. Given feature maps of the size $N_j \times H_j \times W_j$, the style loss is computed as follows:

$$L_{style} = E_j \left[\left\| G_j^\phi(\hat{J}) - G_j^\phi(J) \right\|_1 \right], \quad (8)$$

G_j^ϕ is the Gram-matrix of size $N_j \times N_j$ computed from activation maps ϕ_i from above.

3.3 Implementation Details

The generator of each of the presented GANs consists of an encoder, aiming to compute characteristic features on a reduced resolution, and a decoder, aiming to transform the features to the original size. The encoder starts with a convolutional block, concluded with a ReLU-based activation. Two convolutional blocks follow, each one concluded by a pooling layer. Finally, there are four residual blocks, each one as in (Johnson et al., 2016). The decoder possesses a mirror-symmetric structure to the encoder, containing four residual blocks, two blocks with transpose convolution and unpooling layer, as well as one convolutional block. This last convolutional block has one output channel for G_1 and G_2 and three for G_3 since the latter aims to restore a color image. The discriminator is based on a 70×70 patch GAN (Isola et al., 2017). It has four blocks of convolution layers with ReLU activation between them and the softmax layer at the end, such that the output of the discriminator is between 0 and 1. The two last layers

are fully connected. All layers in the different networks are instance-normalized (Ulyanov et al., 2017). For optimization, we used the Adam implementation in PyTorch with parameters $\beta_1 = 0$, $\beta_2 = 0.9$, see (Kingma and Ba, 2014). The network is trained using 256×256 images with a batch size of eleven, eight, and five for the edge, label, and texture GAN, respectively. Instance normalization is applied within each layer. Further important parameters are: learning rate 0.0001 and iteration number 600,000.

4 RESULTS

4.1 Dataset

Several investigations have already dealt with the labeling of building façades and provide datasets for this purpose (Gadde et al., 2016; Tyleček and Šára, 2013). Merging these different datasets is challenging because not all of them are rectified and also because they have different labeling requirements regarding quality and class selection. We used the data of the eTRIMS Image Database (Korč and Förstner, 2009), which contains labeled data from different European cities. The authors have also provided an annotation tool (Korč and Schneider, 2007), with which the data could be subsequently edited or new images annotated. After these steps, we have five building-induced classes (wall, roof, door, window, and railing) and three background classes (vegetation, sky, and a particular non-building class that includes roads, signs, cars, pedestrians, etc.). Due to mislabeling, classes may still be incorrectly assigned in individual images, since only the class labels of the original dataset were combined, but not every image was checked. Also, due to poor perspective and consequent excessive distortion during rectification, not all images can be used. In addition, there are few pixels that do not belong to any of these classes. They were therefore assigned to the corresponding neighbor classes using a nearest-neighbor algorithm.

Finally, we have to remove the typical clutter objects which are located in front of façades. Vegetation takes the largest part in this process and thus, it makes sense that the AI learns inpainting with vegetation-like masks. LabelMe (Russell et al., 2008) is a free online annotation tool that makes it easy to label images and make them available to the public. Also, many images from cities and streets are stored in their database. We searched the dataset for images with vegetation-like classes and parsed them for mask creation. Six uniformly sized 10% intervals are formed between 0 and 60 percentage points. The total number

of images was 903 while subdivision into 75, 15, and 10 percents was used for training, validation, and testing, respectively. The usual data augmentation module containing rescaling, flipping along the vertical direction, contrast, and brightness adjustment has been performed as well, aiming at a better generalization of the model.

4.2 Quantitative Evaluation

Using data presented in Section 4.1, we compared the proposed approach with the EdgeConnect (Nazeri et al., 2019), since we wish to assess to what extent these maps may improve the results. Moreover, an implementation of the method of (Liu et al., 2018), developed by Nvidia and available online, has been modified for our purposes and trained on our data. Using the same evaluation metrics as the authors of (Nazeri et al., 2019), namely SSIM (structural similarity index), FID (Fréchet inception distance (Heusel et al., 2017)), PSNR (peak signal-to-noise ratio), and the absolute differences of the pixel values from the original and generated image (MAE), we guarantee the fairness of the comparison.

The metrics are calculated and recorded in Table 1 for the six mask size intervals individually and over all mask sizes (0-60%). The numbers marked in bold represent the best results of the respective row. For the majority of configurations, the EdgeConnect method performs best. Our 3GAN approach provides minimally better results for SSIM with larger masks and equal or slightly worse results in almost all other cases. On the positive side, we note that both GAN-based approaches outperform the one based on partial convolutions by far. Searching for the main reasons explaining a mostly better performance of the EdgeConnect method, we must state that the insufficiencies in the semantic segmentation may negatively affect the inpainting results. Very occasionally, the test images exhibit a small, inexplicable black spot, which slightly worsens the values of all metrics.

4.3 Qualitative Evaluations

In Figures 2, 3, and also 1, bottom, we show examples of façade image inpainting with the 3GAN approach and competing methods. It becomes evident that our method can close even big gaps plausibly. Without using any kind of grammar, a semantic segmentation image can be filled. From there, texture portions are generated class-wise without overfitting since the forms and colors of e.g. windows are not the same as in other entities of the same image. Since the entities are not “plagiarized”, the observer usually does not

Table 1: Quantitative comparison of three different approaches. By P.C. and E.C., we denote the Partial Convolution and EdgeConnect methods as proposed in (Liu et al., 2018) and (Nazeri et al., 2019), respectively.

Metric	Mask	P.C.	E.C.	Ours
II	0-10%	0.03	0.02	0.02
	10-20%	0.05	0.02	0.03
	20-30%	0.06	0.03	0.04
	30-40%	0.09	0.04	0.05
	40-50%	0.11	0.06	0.06
	50-60%	0.14	0.08	0.08
	0-60%	0.08	0.04	0.05
SSIM	0-10%	0.95	0.99	0.98
	10-20%	0.87	0.95	0.95
	20-30%	0.81	0.90	0.91
	30-40%	0.70	0.83	0.84
	40-50%	0.56	0.74	0.76
	50-60%	0.47	0.66	0.67
	0-60%	0.73	0.85	0.85
PSNR	0-10%	25.50	31.08	30.20
	10-20%	20.76	25.64	25.07
	20-30%	18.99	23.09	22.87
	30-40%	16.67	21.71	21.20
	40-50%	14.95	18.58	18.39
	50-60%	13.83	17.77	17.31
	0-60%	18.45	22.98	22.51
FID	0-10%	24.09	5.75	8.09
	10-20%	47.66	16.66	20.28
	20-30%	68.29	28.95	30.31
	30-40%	109.64	49.76	51.89
	40-50%	144.67	66.86	68.20
	50-60%	170.90	82.02	74.93
	0-60%	50.86	20.42	22.70

immediately notice that the images are fake but probably needs a more profound second glance. We show both the complete image \hat{J} processed by our method and also the one in which only the inpainted part has been replaced, to explore the effect of “vanilla” convolutions (Yu et al., 2019) on \hat{J} . It turned out that for a negligibly small percentage of analyzed images, J and \hat{J} indeed looked slightly different.

While comparing the results to those achieved by EdgeConnect, we notice that the label image for the proposed 3GAN approach describes a clearer structure of the image. Thus, objects appear clearer than in EdgeConnect, where some texture artifacts are noticeable, probably, resulting from a noisy edge image. The CNN-based approach based on partial convolutions gives simple results with a rather random setting of windows. The network cannot distinguish between different classes. A reasonably good color gradient is generated; however, a strong blue cast negatively affects the results. For all three methods, a morphological dilation around the foreground pixels allows

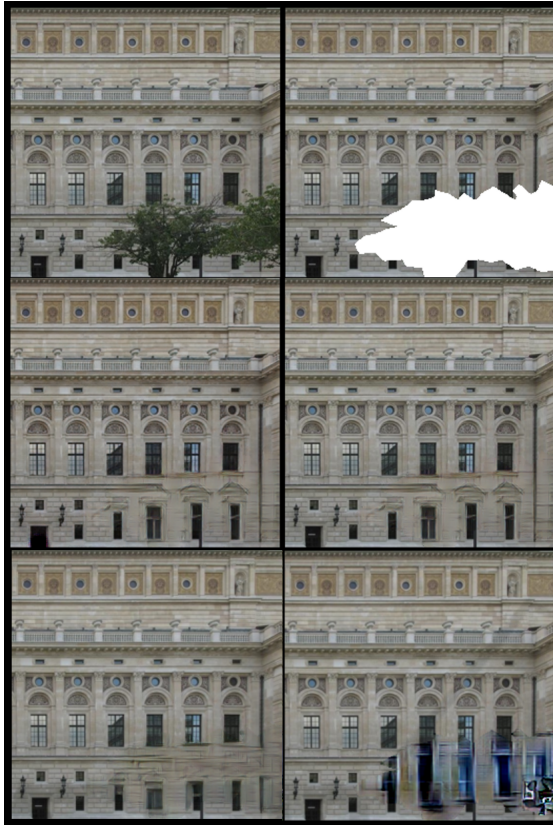


Figure 2: Inpainting example 1: Top row: RGB façade image (left) and mask specified by white color (right). Second row: result of the proposed 3GAN approach, direct output on the left and inpainted mask parsed into the original image on the right. Note the classical window design in the ground floor. Bottom row: Inpainting with EdgeConnect (left) and CNN-based approach (Liu et al., 2018) (right).

excluding the most disturbing shadows. The parts of the façade that shine through the foliage are very necessary for patch-based methods, like that of (Criminisi et al., 2004), but can be omitted for all CNN-based methods. To remove larger shadows, shadow class computation would be required, analogously to trees, but for example, the shadow course in the inpainted image of Figure 1, middle row, does not disturb.

5 CONCLUSION AND OUTLOOK

We presented a new method for inpainting of foreground objects, such as trees and road signs, in façade images. The approach is oriented on the EdgeConnect method but based on three GANs: the first for the inpainting of edge images, the second for the inpainting of (semantic) segmentation images, coded as label maps, and the third for the inpainting of tex-

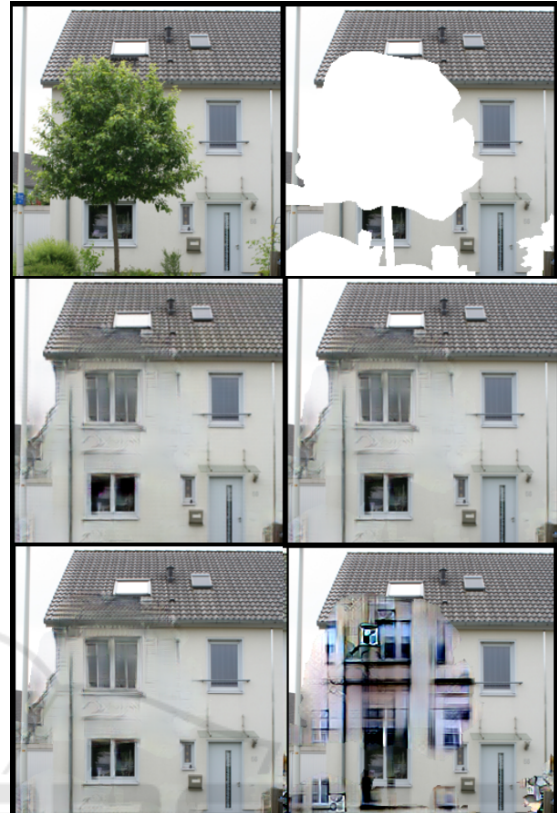


Figure 3: Inpainting example 2; ordering of images according to Figure 2.

ture images.

Comparison of the proposed method with EdgeConnect yields a symbolic draw: qualitatively slightly more advantageous and quantitatively slightly inferior. Objects such as windows are set in a qualitatively better, more accurate, and geometrically correct way. The conceptual advantage of EdgeConnect over the proposed method is that it does not need a label image. The sequence of three GANs is responsible for error propagation, and since a GAN is based on Deep Learning and thus represents a blackbox, it is very difficult to correct those errors in the inpainted label image that from the semantic segmentation or manual annotation. This is a problem because the third GAN is based on this image. Still, the two GAN-based methods are far superior to the CNN-based method both quantitatively and qualitatively, which shows the importance of the information provided by edges and classes in the reconstruction of façades. In addition to edges, façade elements contain many other properties that are useful for automatic methods: rectangular structures, symmetries, etc. If in the past these properties were exploited using production networks (Michaelsen et al., 2012), in the future, they can also

be learned in upcoming modifications of the EdgeConnect and 3GAN approach. Our further intentions for future work include the implementation of semantic segmentation of façades to provide a complete pipeline from sensor data to cleaned façades, as well as generalization of the 3GAN approach to a wider class of problems.

ACKNOWLEDGEMENTS

We express our deep gratitude to Dr. Susanne Wenzel for providing us the eTRIMS dataset (Korč and Förstner, 2009) and the labeling tool. We thank the authors of (Liu et al., 2018) and (Nazeri et al., 2019) who put their code online.

REFERENCES

- Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., and Wernerus, P. (2014). Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:157–170.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.
- Chan, T. F. and Shen, J. (2001). Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449.
- Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212.
- Elharrouss, O., Almaadeed, N., Al-Maadeed, S., and Akbari, Y. (2019). Image inpainting: A review. *Neural Processing Letters*, 51:2007–2028.
- Fathalla, R. and Vogiatzis, G. (2017). A deep learning pipeline for semantic facade segmentation. In *Proc. British Machine Vision Conference*, page 120.1—120.13.
- Gadde, R., Marlet, R., and Paragios, N. (2016). Learning grammars for architecture-specific facade parsing. *International Journal of Computer Vision*, 117(3):290–316.
- Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *arXiv preprint arXiv:1706.08500*.
- Huang, Z., Qin, C., Liu, R., Weng, Z., and Zhu, Y. (2021). Semantic-aware context aggregation for image inpainting. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2465–2469. IEEE.
- Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):1–14.
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134.
- Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Proc. IEEE European Conference on Computer Vision (ECCV)*, pages 694–711. Springer.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Korč, F. and Förstner, W. (2009). eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn.
- Korč, F. and Schneider, D. (2007). Annotation tool. Technical Report TR-IGG-P-2007-01, Dept. of Photogrammetry, University of Bonn.
- Kottler, B., Bulatov, D., and Schilling, H. (2016). Improving semantic orthophotos by a fast method based on harmonic inpainting. In *Proc. 9th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*, pages 1–5. IEEE.
- Kottler, B., Bulatov, D., and Zhang, X. (2020). Context-aware patch-based method for facade inpainting. In *VISIGRAPP (1: GRAPP)*, pages 210–218.
- Li, J., Wang, N., Zhang, L., Du, B., and Tao, D. (2020). Recurrent feature reasoning for image inpainting. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7760–7768.
- Liao, L., Xiao, J., Wang, Z., Lin, C.-W., and Satoh, S. (2020). Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *Proc. 16th European Conference on Computer Vision, Part XXVII 16*, pages 683–700. Springer.
- Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *Proc. IEEE European Conference on Computer Vision (ECCV)*, pages 85–100.
- Michaelsen, E., Iwaszczuk, D., Sirmacek, B., Hoegner, L., and Stilla, U. (2012). Gestalt grouping on facade textures from IR image sequences: Comparing different production systems. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B3):303–308.
- Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., and Ebrahimi, M. (2019). EdgeConnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*.

- Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544.
- Pyo, J., Rocha, Y. G., Ghosh, A., Lee, K., In, G., and Kuc, T. (2020). Object removal and inpainting from image using combined GANs. In *Proc. 20th International Conference on Control, Automation and Systems (IC-CAS)*, pages 1116–1119.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173.
- Sajjadi, M. S., Schölkopf, B., and Hirsch, M. (2017). Enhancenet: Single image super-resolution through automated texture synthesis. In *Proc. International Conference on Computer Vision (ICCV)*, pages 4491–4500.
- Schmitz, M. and Mayer, H. (2016). A convolutional network for semantic facade segmentation and interpretation. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:709–716.
- Shalunts, G., Haxhimusa, Y., and Sablatnig, R. (2011). Architectural style classification of building facade windows. In *International Symposium on Visual Computing*, pages 280–289. Springer.
- Shao, H., Wang, Y., Fu, Y., and Yin, Z. (2020). Generative image inpainting via edge structure and color aware fusion. *Signal Processing: Image Communication*, 87-115929:1–9.
- Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., and Kuo, C.-C. J. (2018). Spg-net: Segmentation prediction and guidance network for image inpainting. In *Proc. British Machine Vision Conference*, volume 97, pages 1–14.
- Tyleček, R. and Šára, R. (2013). Spatial pattern templates for recognition of objects with regular structure. In *Proc. German Conference on Pattern Recognition (GCPR)*, pages 364–374, Saarbrücken, Germany.
- Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6924–6932.
- Wenzel, S. (2016). *High-level facade image interpretation using marked point processes*. PhD thesis, University of Bonn.
- Wenzel, S. and Förstner, W. (2016). Facade interpretation using a marked point process. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:363–370.
- Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., and Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6721–6729.
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 4471–4480.
- Zhang, R., Li, W., Wang, P., Guan, C., Fang, J., Song, Y., Yu, J., Chen, B., Xu, W., and Yang, R. (2020). Autoremover: Automatic object removal for autonomous driving videos. *Proc. AAAI Conference on Artificial Intelligence*, 34(07):12853–12861.