Boundary-aware Image Inpainting with Multiple Auxiliary Cues

Yohei Yamashita Kodai Shimosato Norimichi Ukita
Toyota Technological Institute
ukita@toyota-ti.ac.jp

Abstract

Image inpainting (a.k.a. image completion) allows us to remove unexpected foreground objects from an observed image and to restore the removed region with background pixels. The performance of image inpainting is improved by auxiliary cues such as edge boundaries and segmentation regions. As a new auxiliary cue, this paper focuses on a depth image that is estimated from an input RGB image by monocular depth estimation. In the depth image, boundaries between different objects (e.g., objects located in different distances) with similar pixel values might be available, while those boundaries are difficult to be detected by edge detection and segmentation. Our proposed method employs those boundaries in the edge and depth images as auxiliary cues. Experiments demonstrate that our proposed method augmented by the depth image outperforms its baseline quantitatively (i.e., 1.17dB and 0.74dB PSNR gains on the Paris-StreetView and Places datasets, respectively) and qualitatively.

(a) (b) (c) (d)

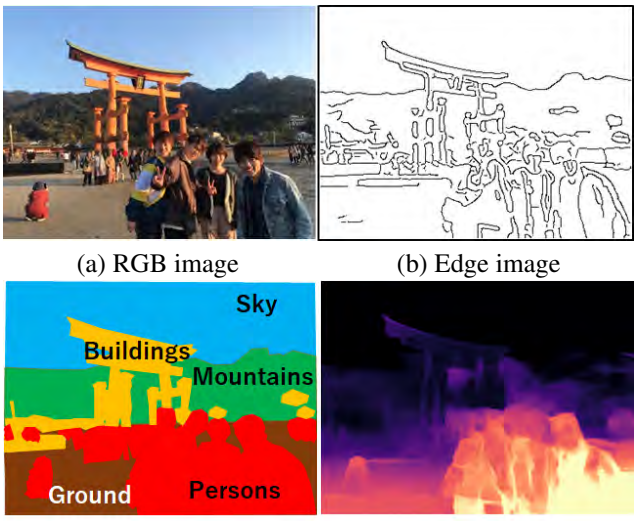
Figure 1. Comparison between the results of the baseline [40] and our proposed method. (a) Ground-truth RGB image, (b) masked RGB image where masked pixels are indicated by white pixels, (c) the result of the base method [40], and (d) the result of our method.

1. Introduction

Image inpainting restores unknown pixels in an image. In an example shown in Fig. 1, an original RGB image (a) is partially masked to make a masked image (b), which is an input image given to image inpainting. Image inpainting allows us to develop various real-world applications. For example, unexpected photobombs such as other tourists can be removed from sightseeing photos, noise such as glares can be removed, and so on. A basic approach for image inpainting [2, 32] is to predict the image structure within the masked region based on pixels surrounding the masked region. As demonstrated in [20, 30, 33–38, 40, 42, 43, 45, 52, 54, 55, 57, 58, 61, 62, 64], this approach is improved by deep convolutional neural networks trained with a huge number of sample images.

Image inpainting can be further improved by auxiliary cues such as edge boundaries [40] and segmentation regions [45]. The examples of edge and segmentation images, which are estimated from Fig. 2 (a), are shown in Fig. 2 (b) and Fig. 2 (c), respectively. Since these auxiliary images are simpler than its original RGB image, completing the masked region in each auxiliary image is easier than that in the RGB image. Therefore, these auxiliary images can be completed first, and then these completed auxiliary images support the image inpainting of the RGB image, as proposed in EdgeConnect [40]. In EdgeConnect, the edge image estimated by a local filter (i.e., Canny edge detector [4]) is used as an auxiliary image. However, the edge image often lacks boundaries between different objects with similar pixel values. These lacks result in unsuccessful support for RGB image inpainting.

This paper focuses on how to provide a more reliable auxiliary cue (or the one working complementarily with the edge image). As such an auxiliary cue, we propose to employ a depth image (Fig. 2 (d)). The depth image can be also estimated from the RGB image by monocular depth estimation. Its accuracy is improved also by convolutional



(a) RGB image (b) Edge image

(c) Segmentation image (d) Depth image

Figure 2. Auxiliary cues for image inpainting. In (b), pixels where pixel values are significantly changed among neighboring pixels are detected. In (c), pixels belonging to the same semantic region (e.g., sky, mountains, and persons) are clustered. In (d), regions closer to a camera are colored by brighter colors.

neural networks with wide receptive fields, as demonstrated in [1, 3, 9, 11, 15, 16, 18, 19, 39, 53]. In particular, it is expected that, in the depth image, boundaries between different objects (i.e., objects located in different distances) with similar pixel values, while those boundaries are difficult to be detected by edge detection.

The novel contributions of this paper are as follows:

- **Depth-aware two-step inpainting:** Image inpainting is supported by a depth image obtained by monocular depth estimation. Such a depth image provides us the boundaries of objects with similar pixel values. These boundaries can successfully help to predict object boundaries within a masked region for image inpainting, as proven in [40] with an edge image.
- **Auxiliary-cue fusion using gated convolutions:** Several object boundaries might be correctly estimated in only either of edge and depth images. In order to appropriately select these correct boundaries, our proposed method employs gated convolutions [57].

2. Related Work

2.1. Image Inpainting

A network for image inpainting is trained in a supervised manner using a reconstruction loss so that an input masked image is fed into the network and its output is equal to the ground-truth complete image. In addition to this reconstruction loss, additional loss functions are employed for improving the completion quality. Adversarial losses [17] are used for improving the fidelity in many methods [20, 21, 25, 27, 33, 35, 37, 43, 45, 52, 55, 57, 61, 64–66, 68]. The perceptual quality is progressed by style losses [20, 23, 27, 33] and perceptual losses [23, 27, 33, 38, 45, 47, 64, 68]. In [59], the reconstruction and perceptual losses are balanced in the frequency domain. Inappropriate local optima (e.g., checkboard artifacts [28]) are suppressed by total variation losses [23, 27, 33].

The inpainting quality is changed also depending on the type of convolution filters. Dilated convolutions are useful for efficiently incorporating wide contexts [25, 27, 37, 45, 52, 55, 57, 61]. Partial convolutions [36] are proposed for image inpainting with irregular masks [20, 33]. The partial convolutions are generalized with a learnable dynamic-feature-selection mechanism in gated convolutions [57]. The gated convolutions are used for image inpainting also in [27, 35, 55]. DSNet [51] also copes with difficulty in irregular masks by proposing two types of dynamic selection modules. Different from the aforementioned convolutions, Fourier convolutions [5] allow us to cover wider regions for inpainting [47]. Such wide receptive fields can be also covered by Transformers [8] as proposed in [60].

Instead of adjusting the receptive fields, large missing holes are filled by iterative hole filling using the confidence map of filled pixels [62] and by co-modulation of both conditional and stochastic style representations [65].

The effectiveness of an attention mechanism is also validated in image inpainting [24, 37, 48, 55, 61, 64, 66, 68] as well as in many other image recognition tasks; see survey papers (e.g., [41]). The attention maps can be used also for the iterative updates of image inpainting [34, 62].

As mentioned in Section 1, several types of auxiliary cues are utilized for image inpainting. Segmentation images are useful for selecting visual features for completion depending on semantic regions [33, 35, 45]. Edge images can be also used for image completion depending on object boundaries [40, 54]. For example, in EdgeConnect [40], a masked edge image as an auxiliary cue is first completed and then a RGB image is fed into an image-inpainting network that is supported by the completed edge image. EdgeConnect can work well if the completion of the auxiliary cue is easier than that of the RGB image. The goal of our proposed method is to further improve image painting by another auxiliary cue (i.e., depth images).

Different from RGB-D inpainting [10], our RGB inpainting method has to employ a depth image estimated by monocular depth estimation. While SLIDE [26] also employs monocular depth estimation, it focuses on the 3D photography task with the estimated depth image. On the other hand, our method focuses on how to utilize the depth cue for improving the RGB inpainting quality.

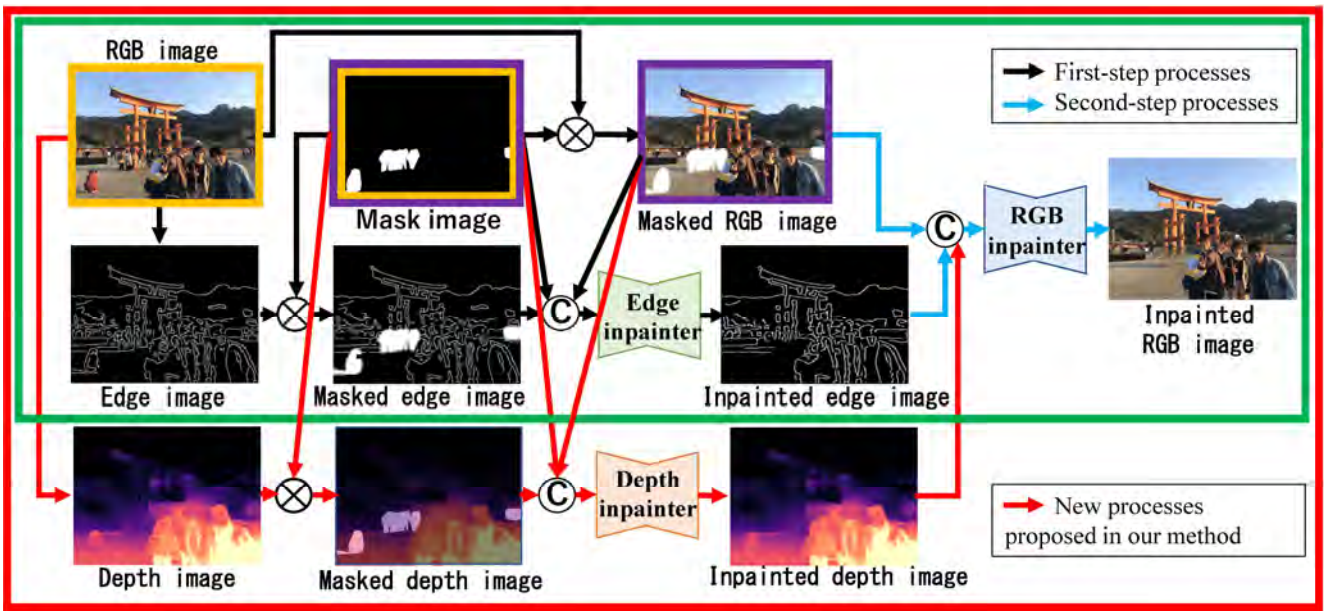


Figure 3. Pipelines of two-step image painting methods. The base method (i.e., EdgeConnect) [40] and our method are shown inside green and red rectangles, respectively. Black, blue, and red arrows indicate processes in the first step, the second step, and our method, respectively. "C" and "×" are concatenation and elementwise-mult operations, respectively. Images enclosed by orange and purple rectangles indicate input data in training and inference stages, respectively. Strictly speaking, instead of the masked RGB image, its masked grayscale image is fed into the edge inpainter and the depth inpainter. For simplicity, this masked grayscale image is omitted in both this figure and the main text.

2.2. Monocular Depth Estimation

Monocular depth estimation [39] is widely studied in computer vision. While its performance is significantly gained by convolutional networks (e.g., [9, 53]), it is further studied in terms of various aspects (e.g., self-supervised learning using sequential frames [3, 16, 18, 19], unsupervised learning using stereo views [3, 11, 15], and improved learning via transfer learning [1]).

3. Edge-aware Two-step Image Inpainting

While a depth image as an auxiliary cue can support various image inpainting approaches, our method is designed with a two-step approach proposed in EdgeConnect [40]. This is because it is easier to complete depth pixels as well as edge boundaries than RGB pixels with complex texture patterns.

The pipeline of EdgeConnect [40] is illustrated inside the green rectangle in Fig. 3. The first and second steps in EdgeConnect corresponding to edge inpainting and RGB inpainting are indicated by black and blue arrows, respectively. The first and second steps have networks for inpainting edge and RGB images, respectively. In what follows, this section briefly introduces the inference and training stages of EdgeConnect; see [40] for the detail.

3.1. Inference

In the inference stage, an input RGB image and its binary mask image (indicated by "RGB image" and "Mask image" in the Fig. 3, respectively) are given.

In the first step, from the input RGB image, its edge image is computed. These RGB and edge images are elementally-multiplied by the binary mask image to get their masked RGB and masked edge images, respectively. These mask, masked RGB, masked edge images are concatenated and fed into the edge inpainter network.

In the second step, the inpainted edge image is concatenated with the masked RGB image and its mask image. The concatenated images are fed into the RGB inpainting network. With the support of the inpainted edge image, RGB image inpainting is achieved.

3.2. Training

In the training stage, the input complete RGB image (denoted by I) and the mask image (denoted by M) are given as training data. M is the binary mask image in which its values are 1 and 0 for the missing region and for background, respectively.

Edge inpainter: Let E denote the edge image of I. In EdgeConnect, the Canny edge detector is used. I and E are masked by M. These masked images are denoted by I_M

and E_M. Then, E_M, M, and I_M are concatenated and fed into the edge inpainter in order to inpaint E_M. This edge inpainter is trained with the weighted sum of the following two loss functions, namely the hinge-variant of GAN loss ($\mathcal{L}_{EG}$) and the feature-matching loss ($\mathcal{L}_{EF}$):

$$\mathcal{L}_E = \lambda_{EG}\mathcal{L}_{EG} + \lambda_{EF}\mathcal{L}_{EF}, \quad (1)$$

where weight constants are $\lambda_{EG} = 1$ and $\lambda_{EF} = 10$.

$\mathcal{L}_{EG}$ is defined as follows:

$$\mathcal{L}_{EG} = -D_E(E_I, I), \quad (2)$$

where E_I denotes the inpainted edge image. The discriminator D_E evaluates whether or not E_I is realistic as the edge image of I. D_E is trained by the following loss:

$$\begin{aligned}\mathcal{L}_{D_E} = &\max(0, 1 - D_E(E_{GT}, I)) \\ &+ \max(0, 1 + D_E(E_I, I))\end{aligned} \quad (3)$$

The feature-matching loss $\mathcal{L}_{EF}$ compares the activation maps in the intermediate layers of D_E. As proposed in [12, 28], feature-level matching between E_I and E allows us to perceptually make E_I look as much like E as possible.

$$\mathcal{L}_{EF} = \sum_i \frac{1}{N^i} \parallel D_E^i(E) - D_E^i(E_I) \parallel_1, \quad (4)$$

where N^i and D_E^i denote the number of elements and the activation in the i-th activation layer of D_E, respectively.

RGB inpainter: E_I and I_M are concatenated and fed into the RGB inpainter in order to inpaint I_M, as shown in "Inpainted RGB image" in Fig. 3. This RGB inpainter is trained with the weighted sum of the following four loss functions, namely the hinge-variant of GAN loss ($\mathcal{L}_{IG}$), the feature-matching loss ($\mathcal{L}_{IF}$), the style loss ($\mathcal{L}_{IS}$), and the reconstruction loss ($\mathcal{L}_{IR}$):

$$\mathcal{L}_I = \lambda_{IG}\mathcal{L}_{IG} + \lambda_{IF}\mathcal{L}_{IF} + \lambda_{IS}\mathcal{L}_{IS} + \lambda_{IR}\mathcal{L}_{IR}, \quad (5)$$

where $\lambda_{IG} = \lambda_{IF} = 0.1$, $\lambda_{IS} = 250$, and $\lambda_{IR} = 1$.

As with $\mathcal{L}_{EG}$ and $\mathcal{L}_{EF}$, $\mathcal{L}_{IG}$ and $\mathcal{L}_{IF}$ are defined as the hinge-variant of GAN loss and the feature-matching loss, respectively.

The activations used in $\mathcal{L}_{IF}$ are also employed for $\mathcal{L}_{IS}$ as follows:

$$\mathcal{L}_{IS} = \sum_i \parallel G_\phi^i(I) - G_\phi^i(I_I) \parallel_1, \quad (6)$$

where G_ϕ^i denotes a Gram matrix constructed from activations ϕ^i [13]. $\mathcal{L}_{IS}$ relieves checkerboard artifacts [44].

The reconstruction loss, $\mathcal{L}_{IR}$, directly evaluates the difference between I and I_I:

$$\mathcal{L}_{IR} = \frac{1}{N_M} \parallel I - I_I \parallel_1, \quad (7)$$

where N_M denotes the number of masked pixels in M.

4. Image Inpainting with Adaptive Fusion of Multi Auxiliary Maps

This section describes our proposed method. While our method is based on EdgeConnect [40] in terms of a two-step inpainting scheme, our method (1) utilizes an additional auxiliary cue (i.e., depth map) that works complementarily with an edge image and (2) adaptively fuses these two auxiliary cues by gated convolutions.

4.1. Training of the Depth Inpainter

The input complete RGB image and the mask image are given as training data in the training stage of our proposed method as with EdgeConnect. The goal is to optimize the three networks (i.e., "Depth inpainter" in addition to "Edge inpainter" and "RGB inpainter" in Fig. 3). In addition to all processes in EdgeConnect enclosed by the green rectangle in Fig. 3, additional processes indicated by red arrows are proposed for our method.

In the first step, from the complete RGB image, its depth image is estimated. While any monocular depth estimation is applicable, our method employs Dense Depth [1]. After this estimated depth image is elementally-multiplied with the mask image, the masked depth image is concatenated with the masked RGB image and the mask image, and fed into the depth inpainting network. This depth inpainter is trained with the weighted sum of the following two loss functions, namely the hinge-variant of GAN loss ($\mathcal{L}_{DG}$) and the feature-matching loss ($\mathcal{L}_{DF}$):

$$\mathcal{L}_D = \lambda_{DG}\mathcal{L}_{DG} + \lambda_{DF}\mathcal{L}_{DF}, \quad (8)$$

where λ_{DG} and λ_{DF} are weight constants.

$\mathcal{L}_{DG}$ and $\mathcal{L}_{DF}$ are equal to $\mathcal{L}_{EG}$ in Eq. (2) and $\mathcal{L}_{EF}$ in Eq. (4) except that the depth image is used in $\mathcal{L}_{DG}$ and $\mathcal{L}_{DF}$ instead of the edge image.

4.2. Training of the RGB Inpainter with Adaptive Fusion of Multi Auxiliary Maps

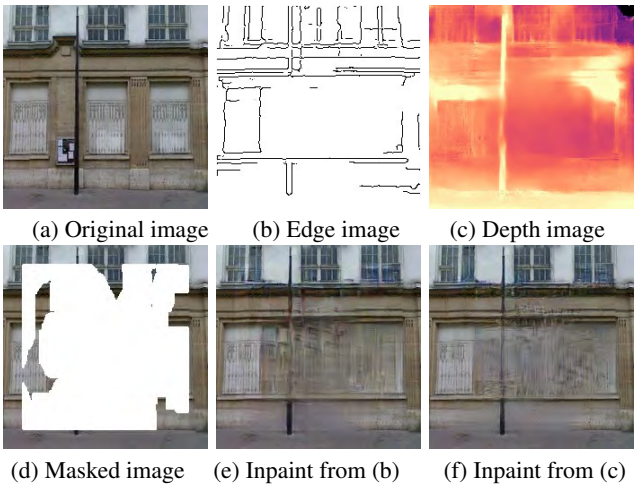
The RGB inpainter of our proposed method is trained with the weighted sum of the following four loss functions, namely the hinge-variant of GAN loss ($\hat{\mathcal{L}}_{IG}$), the feature-matching loss ($\hat{\mathcal{L}}_{IF}$), the style loss ($\hat{\mathcal{L}}_{IS}$), and the reconstruction loss ($\hat{\mathcal{L}}_{IR}$).

$$\hat{\mathcal{L}}_I = \hat{\lambda}_{IG}\hat{\mathcal{L}}_{IG} + \hat{\lambda}_{IF}\hat{\mathcal{L}}_{IF} + \hat{\lambda}_{IS}\hat{\mathcal{L}}_{IS} + \hat{\lambda}_{IR}\hat{\mathcal{L}}_{IR}, \quad (9)$$

where weight constants are $\hat{\lambda}_{IG} = \hat{\lambda}_{IF} = 0.1$, $\hat{\lambda}_{IS} = 250$, and $\hat{\lambda}_{IR} = 1$. As with $\hat{\mathcal{L}}_{DG}$ and $\hat{\mathcal{L}}_{DF}$, $\hat{\mathcal{L}}_{IG}$ and $\hat{\mathcal{L}}_{IF}$ are defined as follows:

$$\hat{\mathcal{L}}_{IG} = -\hat{D}_I(I_I, D_C), \quad (10)$$

where I_I and D_C denote the inpainted RGB image and composite depth image $D_C = D \odot (1 - M) + D_I \odot M$



(a) Original image (b) Edge image (c) Depth image

(d) Masked image (e) Inpaint from (b) (f) Inpaint from (c)

Figure 4. Difference between (b) edge and (c) depth images.

where M denotes the binary mask image, respectively. The discriminator $\hat{D}_I$ evaluates whether or not I_I is realistic as the RGB image of D_C. $\hat{D}_I$ is trained by Eq. (3) so that D_E, E_{GT}, and I in Eq. (3) are substituted by $\hat{D}_I$, I_{GT}, and D_C, respectively.

$$\hat{\mathcal{L}}_{IF} = \sum_i \frac{1}{N_\phi^i} \parallel \phi^i(I) - \phi^i(I_I) \parallel_1, \quad (11)$$

where N_ϕ^i and ϕ^i denote the number of elements and the activation in the i-th activation layer of VGG-19 pretrained with ImageNet, respectively. The activations used in $\hat{\mathcal{L}}_{IF}$ are also employed for $\hat{\mathcal{L}}_{IS}$ as follows:

$$\hat{\mathcal{L}}_{IS} = \sum_i \parallel G_\phi^i(I) - G_\phi^i(I_I) \parallel_1 \quad (12)$$

The reconstruction loss, $\hat{\mathcal{L}}_{IR}$, is equal to Eq. (7) as follows:

$$\hat{\mathcal{L}}_{IR} = \frac{1}{N_M} \parallel I - I_I \parallel_1 \quad (13)$$

The edge and depth images with the masked RGB image are fed into the RGB inpainter. Since the boundaries of the edge and depth images are different as shown in Fig. 4, these two images provide complementary cues for image inpainting in the masked RGB image. In Fig. 4, the boundary lines of a pole located in the image center are not detected in (b) the edge image but detected in (c) the depth image. This difference leads to the presence and absence of the pole in (f) and (e), respectively.

For complementarily extracting the aforementioned features from multiple sources, multimodal data fusion is useful. As well as most computer vision methods, multi-modal data fusion is achieved by CNNs, for example, early fusion [6] and late fusion [31, 46, 49]. In accordance with the fusion scheme employed in EdgeConnect, all of the masked RGB, edge, and depth images are concatenated and fed into the RGB inpainter.

However, standard fixed-shape convolutions may have difficulty in extracting effective features simultaneously from these two images. Instead of the fixed-shape convolutions, in our proposed method, gated convolutions [57] are employed in the RGB inpainter for extracting features from different receptive fields depending on the image/channel. In the original work [57], the gated convolution is proposed to achieve adaptive receptive fields for the mixtures of masked and non-masked pixels in the image inpainting task. Our proposed method, on the other hand, employs the gated convolution in order to adaptively determine the receptive field for each of the RGB, edge, and depth images.

4.3. Inference

As with the the original method [40], an input RGB image and its binary mask image are given in the inference stage. In addition to the edge inpainter, the depth inpainter is employed for inpainting the masked depth image. Its output is "Inpainted depth image" in Fig. 3. This inpainted depth image is concatenated with the masked image and the inpainted edge image, and then fed into the RGB inpainter.

5. Experimental Results

Our code will be available at https://github.com/rain58/Boudary-aware-Image-Inpainting.

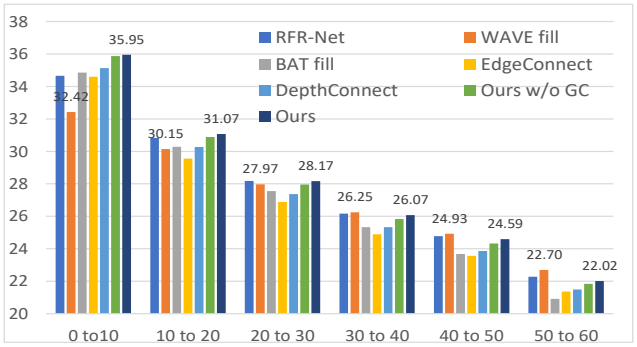
5.1. Depth Estimation

Our proposed method requires monocular depth estimation as a pre-process. This depth estimation was done by Dense Depth [1] in our experiments, as mentioned in Sec. 4.1. Its official implementation is available with the pre-trained model in [1]. While the pre-trained model is trained with images captured by on-vehicle cameras [14], this model is finetuned with a generic scene dataset with image and depth data [50].

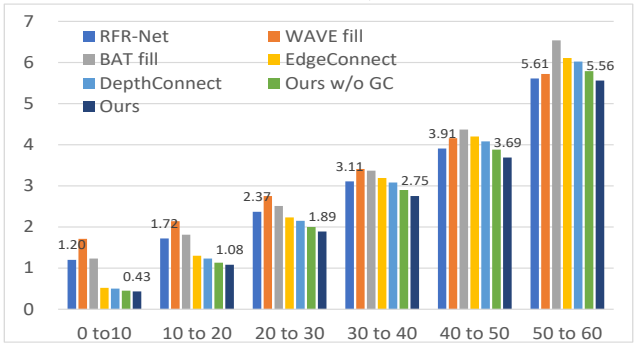
5.2. Datasets

Paris-StreetView Dataset In the Paris-StreetView dataset [7], 14,900 training and 100 test images are contained. While each training image is 936×537 pixels, it is divided into three images (i.e., left, center, and right images, each of which is 537×537 pixels) in accordance with evaluation done in [40]. In total, $14,900 \times 3 = 44,700$ training images are obtained. These 44,700 training images are split into 42,000 training and 2,700 validation images. On the other hand, test images with 228×228 pixels are not split and directly fed into an inpainting network.

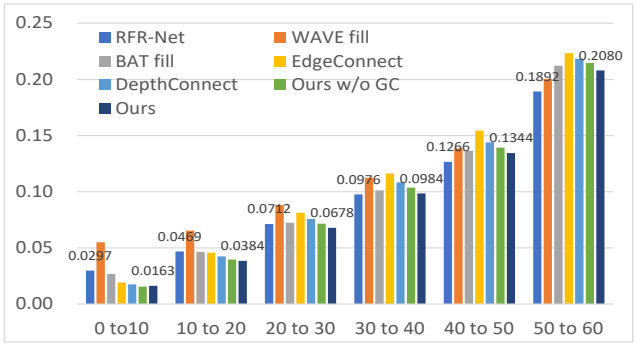
For both training and inference stages, random mask regions are provided by the dataset of [36].



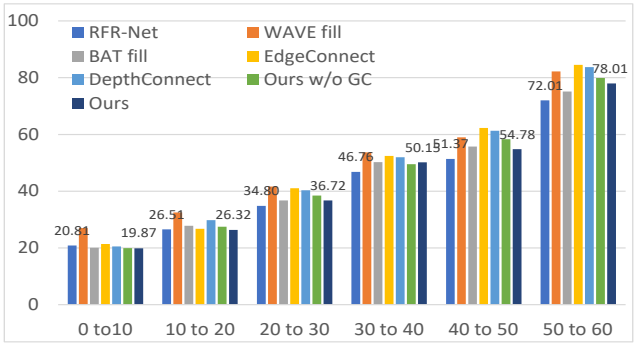
(a) PSNR↑



(b) L1 error↓

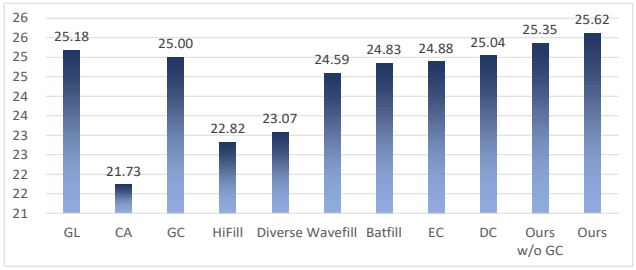


(c) LPIPS↓

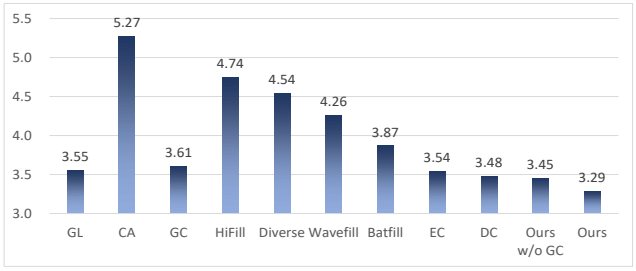


(d) FID↓

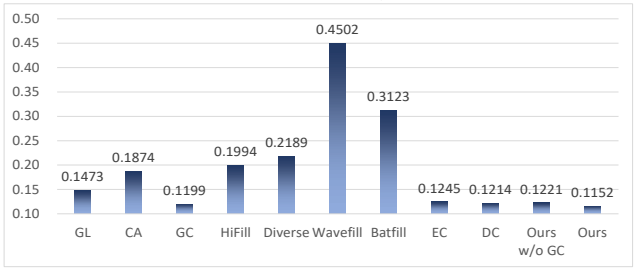
Figure 5. PSNR, L1 error, LPIPS, and FID scores on the Paris-StreetView dataset. These scores are separately evaluated with the mask images of different sizes. The mask images are separated based on the percentage of mask pixels to all pixels in each image, which is indicated in the horizontal axis.



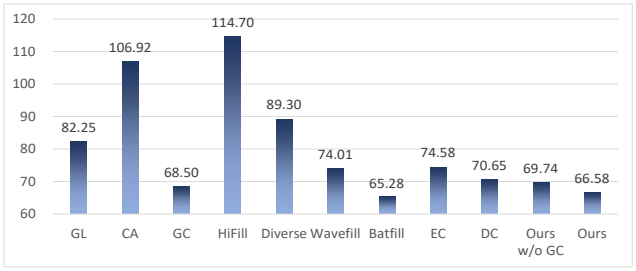
(a) PSNR↑



(b) L1 error↓



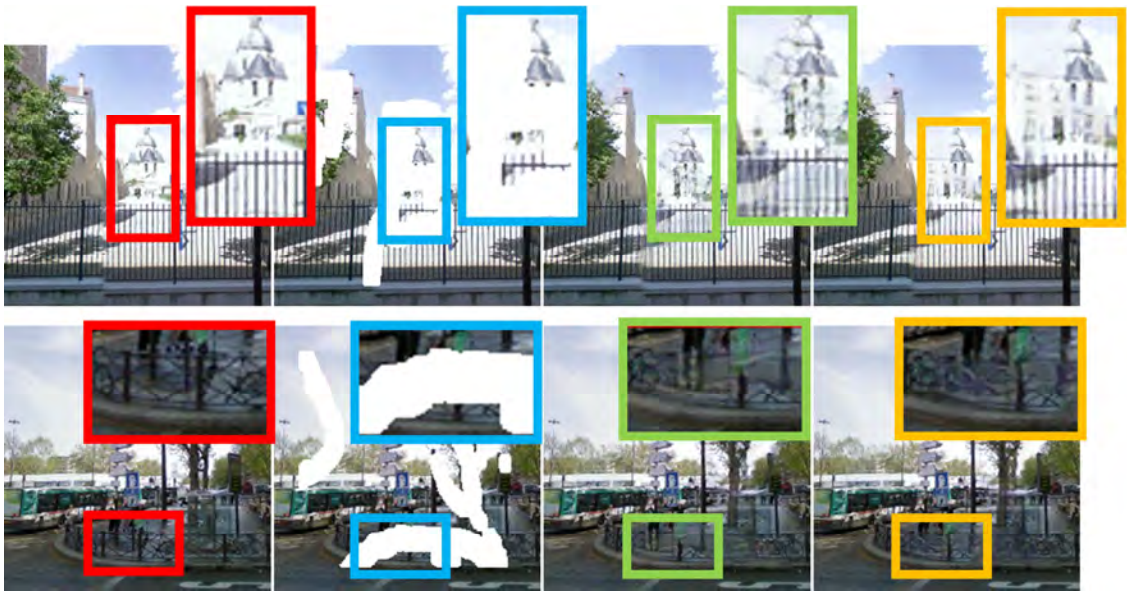
(c) LPIPS↓



(d) FID↓

Figure 6. PSNR, L1 error, LPIPS, and FID scores on the Places dataset. The mean scores over all mask sizes is shown.

Places Dataset The Places dataset [67] has 1,803,460 training, 36,500 validation, and 328,500 test images. While each original image is split into sub-images for training in the Paris-StreetView dataset, each training image is directly fed into a network in the Places dataset. For the evaluation purpose, 100 validation images are used in accordance with related work [25, 34, 42, 55, 57].



(a) Original image (b) Masked image (c) EdgeConnect (d) Our method

Figure 7. Two typical examples of successful image inpainting results obtained by our method on the Paris-StreetView dataset.

5.3. Training Details

The validation images were used for determining the epoch in which the training is finished so that the mean PSNR of the validation images is converged. Adam [29] is employed for an optimizer. The learning rate and the mini-batch size are 0.0001 and 4, respectively.

5.4. Quantitative Evaluation

The performance is assessed using PSNR, L1 error, LPIPS [63], and FID [22] in which larger, lower, lower, and lower values are better, respectively.

Paris-StreetView Dataset For ablation study, EdgeConnect (EC) [40], its modified version in which a depth image is used instead of an edge image, which is called DepthConnect (DC), and our method without the gated convolution (GC) are also evaluated. The results of these methods are shown in Fig. 5. In addition, the results of RFR-Net [34], BATfill [60], and WAVEfill [59] are also shown. The results of [34, 59, 60] are given by the publicly-available authors' codes and weights trained with the Paris-StreetView dataset.

In all metrics, our method outperforms all other methods in smaller mask regions (i.e., "0–10%" and "10–20%"), while RFR-Net is the best in other mask sizes. On the other hand, our method is superior to all other methods in all mask sizes in terms of L1 error.

With both edge and depth images, our fusion method can outperform both EdgeConnect and DepthConnect in all mask regions in all metrics. In comparison between ours without the gated convolution and ours, we can see that the gated convolution successfully improves our method.

Places Dataset The results of comparative experiments on the Places dataset are shown in Fig. 6. As SOTA inpainting methods, global-and-local consistency [25] (GL), contextual attention [56] (CA), gated convolution [57] (GC), contextual residual aggregation [55] (HiFill), diverse structure [42] (Diverse), BATfill [60], WAVEfill [59], and EdgeConnect [40] (EC) are compared with the variants of our method. The results of all of these SOTA methods are obtained by the publicly-available authors' codes and weights trained with the Places dataset[1]. For ablation study, DepthConnect (DC) and our method without the gated convolution are also shown in Fig. 6.

In terms of PSNR, L1 error, and LPIPS, our method outperforms all other methods. Our method is the second best in terms of FID, while the gap from the best one (i.e., BATfill) is small: 65.28 vs. 66.58. It can also be seen that our method is improved by all of the edge image, the depth image, and the gated convolution on the Places dataset as well as on the Paris-StreetView dataset.

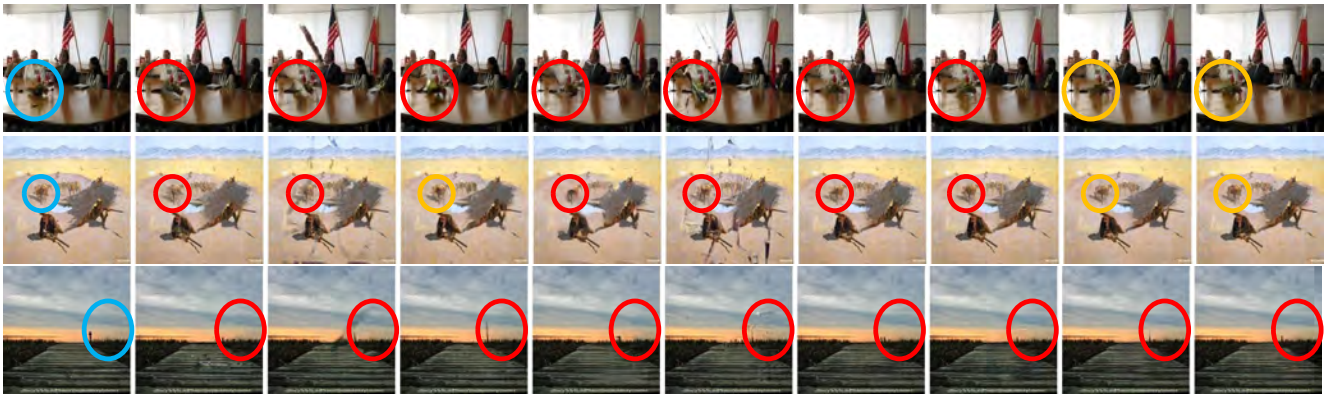
5.5. Visual Evaluation

Paris-StreetView Dataset Figure 7 shows typical examples in which our method can resolve several unnatural artifacts reconstructed by the baseline (EdgeConnect [40]) on the Paris-StreetView dataset.

[1]RFR-Net [34] is not evaluated because its model trained with the Places dataset is not publicly available (on April 18th, 2022).

(a) Original (b) Mask (c) RFR-Net (d) BATfill (e) WAVEfill (f) EC (g) DC (h) Ours w/o GC (i) Ours

Figure 8. Visual Results on the Paris-StreetView dataset. In each row, circles are located in the same positions, and orange and red circles enclose regions that are similar and dissimilar to the ground-truth region, respectively, which is enclosed by a blue circle.



(a) Original (b) GL (c) CAD (d) GC (e) HiFill (f) Diverse (g) EC (h) DC (i) Ours w/o GC (j) Ours

Figure 9. Visual Results on the Places dataset. Circles are overlaid in the same manner in Fig. 8.

In the upper example, the structure of a building is corrupted in (c) EdgeConnect, while our method better reconstructs windows on the wall as shown in (d). In the lower example, the legs of a person are observed beyond the fence. In (c) EdgeConnect, the legs are unnaturally extended, while the image inpainted by our method is more similar to (a) the original image.

More results are shown in Fig. 8.

Places Dataset Figure 9 shows several examples of our method and the SOTA methods. For example, in the middle example, many methods fail to reconstruct the horse and its shadow, while our result is better than those enclosed by red circles. In the bottom example, on the other hand, our method cannot outperform others. In this example, the depth cue is not effective for inpainting a distant region such as the one enclosed by the blue circle in (a), because no depth difference is estimated in such a distant region.

6. Concluding Remarks

This paper proposed an image inpainting method using a depth image as an auxiliary cue for reconstructing ambiguous object boundaries. While the object boundaries take an important role in image inpainting, it is difficult to perfectly detect them. For effective fusion of imperfect edge and depth cues for RGB image inpainting, we propose to utilize the gated convolution instead of the standard convolution.

Future work includes joint end-to-end learning with an edge detection network, while a simple edge detector such as the Canny edge detector is employed in a pre-process. This end-to-end learning allows the whole network to optimize edge boundaries for the image inpainting task. While the effectiveness of the gated convolution for fusing edge and depth features is validated in our experiments, other fusion approaches should be also evaluated.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv*, 1812.11941, 2018. https://github.com/ialhashim/DenseDepth. 2, 3, 4, 5

[2] Marcelo Bertalmío, Luminita A. Vese, Guillermo Sapiro, and Stanley J. Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, 12(8):882–889, 2003. 1

[3] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian D. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019. 2, 3

[4] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence.*, 8(6):679–714, 1986. 1

[5] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In *NeurIPS*, 2020. 2

[6] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013. 5

[7] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. What makes paris look like paris? *Commun. ACM*, 58(12):103–110, 2015. 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 2, 3

[10] Ryo Fujii, Ryo Hachiuma, and Hideo Saito. RGB-D image inpainting using generative adversarial network with a late fusion approach. In Lucio Tommaso De Paolis and Patrick Bourdot, editors, *AVR*, 2020. 2

[11] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. In *ICCV*, 2019. 2, 3

[12] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015. 4

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 4

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 32(11):1231–1237, 2013. 5

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2, 3

[16] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 2, 3

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[18] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 2, 3

[19] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020. 2, 3

[20] Zongyu Guo, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu. Progressive image inpainting with full-resolution residual network. In *ACM International Conference on Multimedia*, 2019. 1, 2

[21] Mohamed Abbas Hedjazi and Yakup Genc. Efficient texture-aware multi-gan for image inpainting. *Knowl. Based Syst.*, 217:106789, 2021. 2

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 7

[23] Xin Hong, Pengfei Xiong, Renhe Ji, and Haoqiang Fan. Deep fusion network for image completion. In *ACM International Conference on Multimedia*, 2019. 2

[24] Zheng Hui, Jie Li, Xiumei Wang, and Xinbo Gao. Image fine-grained inpainting. *arXiv*, 2002.02609, 2020. https://github.com/Zheng222/DMFN. 2

[25] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, 2017. https://github.com/satoshiiizuka/siggraph2017_inpainting. 2, 6, 7

[26] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T. Freeman, David Salesin, Brian Curless, and Ce Liu. SLIDE: single image 3d photography with soft layering and depth-aware inpainting. In *ICCV*, 2021. 2

[27] Youngjoo Jo and Jongyoul Park. SC-FEGAN: face editing generative adversarial network with user's sketch and color. In *ICCV*, 2019. 2

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2, 4

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[30] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided GAN based semantic inpainting. In *CVPR*, 2020. 1

[31] Seungyong Lee, Seong-Jin Park, and Ki-Sang Hong. Rdfnet: RGB-D multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017. 5

[32] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, 2003. 1

[33] Jingyuan Li, Fengxiang He, Lefei Zhang, Bo Du, and Dacheng Tao. Progressive reconstruction of visual structure for image inpainting. In *ICCV*, 2019. 1, 2

[34] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *CVPR*, 2020. https://github.com/jingyuanli001/RFR-Inpainting. 1, 2, 6, 7

[35] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *ECCV*, 2020. 1, 2

[36] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. 1, 2, 5

[37] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *ICCV*, 2019. 1, 2

[38] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *CVPR*, 2021. 1, 2

[39] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021. 2, 3

[40] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCVW*, 2019. 1, 2, 3, 4, 5, 7

[41] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021. 2

[42] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical VQ-VAE. In *CVPR*, 2021. https://github.com/USTC-JialunPeng/Diverse-Structure-Inpainting. 1, 6, 7

[43] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *ICCV*, 2019. 1, 2

[44] Mehdi S. M. Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, 2017. 4

[45] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C.-C. Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. In *BMVC*, 2018. 1, 2

[46] Lei Sun, Kailun Yang, Xinxin Hu, Weijian Hu, and Kaiwei Wang. Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images. *arXiv*, 2002.10570, 2020. 5

[47] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022. https://github.com/saic-mdal/lama. 2

[48] S. M. Nadim Uddin and Yong Ju Jung. Global and local attention-based free-form image inpainting. *Sensors*, 20(11):3204, 2020. https://github.com/SayedNadim/Global-and-Local-Attention-Based-Free-Form-Image-Inpainting. 2

[49] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *Int. J. Comput. Vis.*, 128(5):1239–1285, 2020. 5

[50] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *arXiv*, 1908.00463, 2019. 5

[51] Ning Wang, Yipeng Zhang, and Lefei Zhang. Dynamic selection network for image inpainting. *IEEE Trans. Image Process.*, 30:1784–1798, 2021. https://github.com/wangning-001/DSNet. 2

[52] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018. 1, 2

[53] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018. 2, 3

[54] Jie Yang, Zhiquan Qi, and Yong Shi. Learning to incorporate structure knowledge for image inpainting. In *AAAI*, 2020. 1, 2

[55] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *CVPR*, 2020. https://github.com/Atlas200dk/sample-imageinpainting-HiFill/tree/master/GPU_CPU. 1, 2, 6, 7

[56] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. https://github.com/JiahuiYu/generative_inpainting. 7

[57] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. https://github.com/JiahuiYu/generative_inpainting. 1, 2, 5, 6, 7

[58] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. Region normalization for image inpainting. In *AAAI*, 2020. 1

[59] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *ICCV*, 2021. https://github.com/yingchen001/WaveFill. 2, 7

[60] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, Shijian Lu, Feiying Ma, Xuansong Xie, and Chunyan Miao. Diverse image inpainting with bidirectional and autoregressive transformers. In *ACM MM*, 2021. https://github.com/yingchen001/BAT-Fill. 2, 7

[61] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *CVPR*, 2019. 1, 2

[62] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *ECCV*, 2020. 1, 2

[63] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[64] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. UCTGAN: diverse image inpainting based on unsupervised cross-space translation. In *CVPR*, 2020. 1, 2

[65] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I-Chao Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *ICLR*, 2021. https://github.com/zsyzzsoft/co-mod-gan. 2

[66] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, 2019. 2

[67] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2018. 6

[68] Tong Zhou, Changxing Ding, Shaowen Lin, Xinchao Wang, and Dacheng Tao. Learning oracle attention for high-fidelity face completion. In *CVPR*, 2020. 2