

CR-Fill: Generative Image Inpainting with Auxiliary Contextual Reconstruction

Yu Zeng¹ Zhe Lin² Huchuan Lu^{3,4} Vishal M. Patel¹

zengxianyu18@qq.com, zlin@adobe.com, lhchuan@dlut.edu.cn, vpatel136@jhu.edu

¹Johns Hopkins University

²Adobe Research

³Dalian University of Technology

⁴Peng Cheng Lab

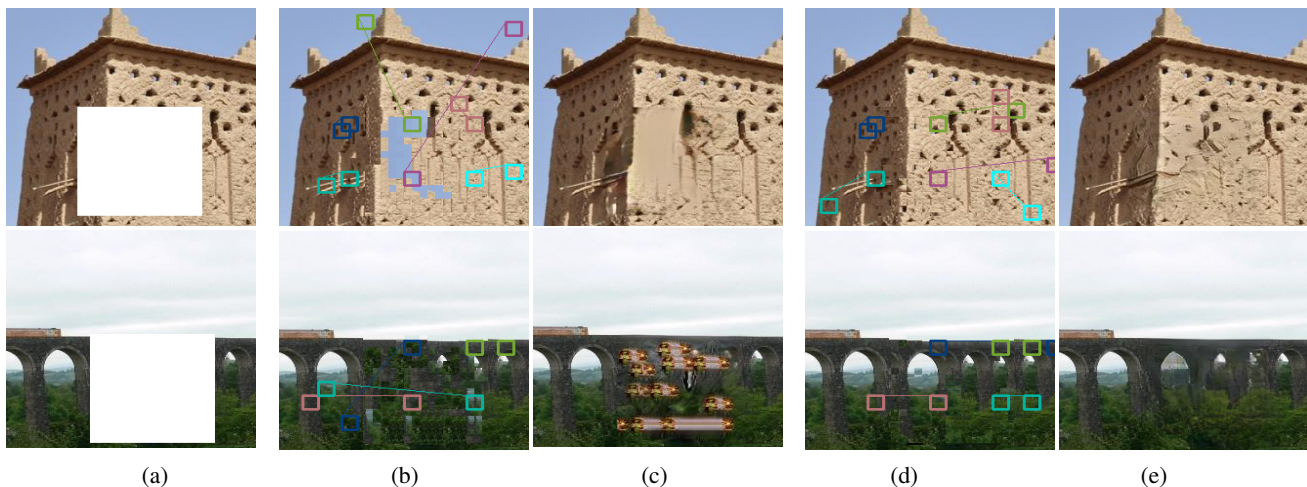


Figure 1: (a) Input images with holes. (b) Sample patch correspondences between missing region and known region established by contextual attention layer, overlaid with the image composed by copying image patches from the known region to the corresponding position in missing region. (c) Inpainting results of DeepFillv2 [29] with CA layer. (d) Patch correspondences generated by the learned CR loss. (e) Inpainting results of our attention-free generator trained with CR loss.

Abstract

Recent deep generative inpainting methods use attention layers to allow the generator to explicitly borrow feature patches from the known region to complete a missing region. Due to the lack of supervision signals for the correspondence between missing regions and known regions, it may fail to find proper reference features, which often leads to artifacts in the results. Also, it computes pair-wise similarity across the entire feature map during inference bringing a significant computational overhead. To address this issue, we propose to teach such patch-borrowing behavior to an attention-free generator by joint training of an auxiliary contextual reconstruction task, which encourages the generated output to be plausible even when reconstructed by surrounding regions. The auxiliary branch can be seen as a learnable loss function, i.e. named as contextual reconstruction (CR) loss, where query-reference feature similarity and reference-based reconstructor are jointly optimized with the inpainting generator. The auxiliary branch (i.e. CR

loss) is required only during training, and only the inpainting generator is required during the inference. Experimental results demonstrate that the proposed inpainting model compares favourably against the state-of-the-art in terms of quantitative and visual performance. Code is available at <https://github.com/zengxianyu/crfill>.

1. Introduction

Image inpainting is a task of predicting missing regions in images. It is an important problem in computer vision and can be used in many applications, e.g. image restoration, compositing, manipulation, re-targeting, and image-based rendering [2, 10, 19]. Traditional methods such as [4, 9, 2] borrow example patches from known regions or external dataset and paste them into the missing regions. They cannot hallucinate novel image contents for challenging cases involving complex, non-repetitive structures. Recent research efforts have shifted the attention to data-driven deep CNN-based approaches [20, 6, 28, 14, 29].

For inpainting, an unlimited amount of paired training data can be automatically generated simply by corrupting images deliberately and using the original images before corruption as the ground-truths. By training on large datasets, deep network-based methods have shown promising results for inpainting complex scenes. An important challenge in inpainting is that there are many plausible answers for filling in a missing region in natural images, and this ambiguity often leads to blurry or distorted structures. To overcome this issue, recent methods [29, 16, 30, 31, 23] try to reduce the uncertainty by explicitly assigning a known region as a reference for filling a missing region. This can be implemented as a patch-borrowing operation, *e.g.* contextual attention (CA) module [28], which copies feature patches from the known reference region and pastes them into the missing region.

However, there is no direct supervision on feature similarity nor the information of patch correspondences in the CA module, thus sometimes inappropriate patches are chosen with higher weights, which leads to artifacts in the consequent inpainting results. Consider the image shown in Fig. 1 (a) with missing pixels. Fig. 1 (b) shows the reference patches selected by CA layer with the largest weight. For visual analysis, we copy each reference patch to the corresponding location in the missing region. (c) shows the inpainting results of DeepFillv2 [29]. By comparing Fig. 1 (b) and Fig. 1 (c), we can see the artifacts are caused by the incorrect reference regions found by the CA module. Moreover, a patch-borrowing operation in the CA layer requires computing the similarity of every pair of patches in the feature map, which is computationally expensive especially for high-resolution images.

To address these issues, we consider to avoid using the explicit patch-borrowing in the inpainting generator (to make the generator efficient and robust to borrowing incorrect reference patches) while retaining and encouraging the patch/feature copy-pasting behaviors through jointly training of contextual reconstruction (to make the result realistic). The idea is partially motivated by Generative Adversarial Networks where its generator and discriminator are trained jointly with losses on both, while only the generator is used during testing.

Specifically, we propose to attach an auxiliary contextual reconstruction branch to the inpainting network as a new training loss. The auxiliary branch can be regarded as a contextual reconstruction loss (CR loss) which encourages the generated output to be plausible even when reconstructed by surrounding (contextual) regions/features. CR loss not only encourages the individual features of generated missing regions reconstructable from features of the surrounding regions but also the complete contextually reconstructed composite visually plausible, in analogy to a jigsaw puzzle.

Through extensive experiments, we validate that a

vanilla attention-free CNN trained with CR loss can learn to inherit the patch-borrowing behavior which was needed to be explicitly enforced with attention layers by previous approaches, as shown in Fig. 1. Moreover, since CR loss is required only during training, there is no computational overhead brought to the inpainting model during inference, making the model much more efficient for testing.

We summarize the contributions of this paper as follow:

- A new learnable, auxiliary contextual reconstruction branch/loss (CR loss) to encourage the generator network to borrow appropriate known regions as references for filling in a missing region.
- An efficient and robust attention-free inpainting generator trained jointly with the traditional inpainting loss and the auxiliary contextual reconstruction loss while allowing efficient inference for testing.
- Extensive experiments showing the effectiveness of the CR loss and favorable performances over the state-of-the-art methods.

2. Related work

Earlier inpainting methods rely on the principle of borrowing known regions to fill missing regions. Diffusion-based methods [1, 3] propagate neighboring content to the missing regions and often result in significant artifacts when filling large holes or when texture is of large variation. Patch-based methods [4, 9, 2] search for the most similar patches from known regions to complete missing regions. They can produce high-quality results for textures and repeating patterns. However, due to the lack of high-level structural understanding and inability of generating new content, the results may not be semantically reasonable. Encouraged by the success of deep CNNs in image restoration tasks, recent research efforts have shifted their attention to deep learning-based methods [26, 22, 14, 18, 11, 12]. To produce sharper results, these methods typically adopt adversarial training inspired by GANs [5]. Pathak *et al.* [20] first attempted to use a CNN for hole filling. Li *et al.* [13] propose a deep generative model for face completion. Iizuka *et al.* [6] use two discriminators to make the inpainted content both locally and globally consistent.

Inspired by exemplar-based inpainting methods, patch-borrowing operations have been integrated into deep learning models. Yu *et al.* [28] propose a contextual attention layer (CA layer) which replaces the generated features in the missing region with linear combinations of feature patches from known region using similarity as weight. Zeng *et al.* [30] propose to use region affinity from a high-level feature map to guide the patch replacement operation in the previous low-level feature map. Yang *et al.* [25] propose a multi-scale neural patch synthesis approach based on joint optimization of image content and texture constraints.

Zeng *et al.* [31] use a related neural patch-vote approach to upsample but avoids the slow optimization by using a modified contextual attention layer. Yan *et al.* [24] shift the encoder features of the known region to the missing region in the mirrored layer of the decoder serving as an estimation of the missing parts. Song *et al.* [21] propose a patch-swap layer that replaces each feature patch in the missing regions with the most similar patch on the known regions. Beside these methods, related variants of patch replacement operations have also been used in other image inpainting methods [16, 15, 29, 23]. These methods implement a heuristic patch-borrowing operation in a network based on the distance between deep features in missing regions and known regions. Different from them, the proposed method teaches a vanilla CNN to borrow more reasonable patches by jointly training a contextual reconstruction branch.

Previous research has studied attention-based loss function for image inpainting. Ma *et al.* [17] propose correlation loss, which is the L1 loss between the feature affinity matrices of the inpainting result and the ground-truth. It directly penalizes the difference of similarity structure of the generated image from the ground-truth, which is too strict for large missing regions. In comparison, CR loss encourages visual realism of the reconstructed image without explicitly forcing it to match the ground-truth. This is more relaxed and provides better guidance to copy appropriate known patterns.

3. Generative inpainting network

We adopt generative adversarial network-based approach for image inpainting, which has a generator and a discriminator. The objective of the discriminator is to discriminate between real images (without any missing pixels) and images inpainted by the generator.

Discriminator. We use a PatchGAN discriminator [7] with spectral normalization and hinge loss to stabilize training following [29]. Inpainted images or the ground-truth complete images are passed to the discriminator, resulting in a score map, where each element is a score corresponding to a local region of the input covered by its receptive field. The loss for the discriminator is:

$$\mathcal{L}_D = \mathbb{E}_{X \sim p_{data}(X)} [\text{ReLU}(\mathbb{1} - D(X))] + \mathbb{E}_{U \sim p_U(U)} [\text{ReLU}(\mathbb{1} + D(G(U) \circ M + U))], \quad (1)$$

where D denotes the discriminator, X represents the real image (ground-truth), U represents the incomplete image with the unknown pixels, M represents a binary mask corresponding of the missing region where $M_{xy} = 1$ indicates that pixel at x, y is missing and $M_{xy} = 0$ indicates that pixel at x, y is valid/known, $G(\cdot)$ represents the generator and \circ denotes element-wise multiplication. The inpainting result $G(U) \circ M + U$ is composed by putting the generator $G(U)$

in the missing region and keeping the original content of U in the known region.

Coarse-to-fine generator. Fig. 2 shows the overall architecture of our generator network. It is a coarse-to-fine architecture, similar to the one in DeepFillv2 [29] but the CA layer is removed and CR loss is applied instead. The coarse network and the refinement network are convolutional encoder-decoder type networks. Dilated convolution layers are used to enlarge the receptive fields. We use gated convolution [29] in all convolution and dilated convolution layers. The coarse network takes an incomplete image where missing pixels are set to zero and a binary mask indicating the missing region as input and generates an initial prediction. Then the refinement network takes this initial prediction as input and outputs the final inpainting result.

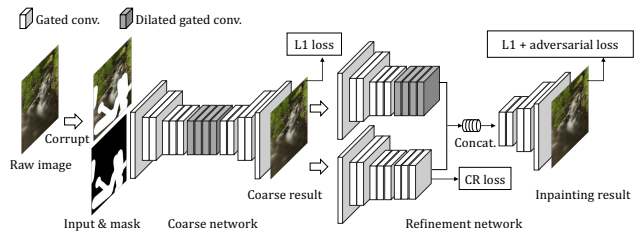


Figure 2: Overall architecture of the generator network.

Training the generator. We expect the coarse network to complete the global structure and then details will be filled in by the refinement network. So only the L1 loss is used to train the coarse network. A combination of L1 loss, adversarial loss and the proposed CR loss is used to train the refinement network. Let $Y = G(U)$ represent the refinement network output. Then the loss for the refinement network is defined as follow:

$$\mathcal{L}_G = \mathbb{E}_{U, X \sim p(U, X)} [L(Y) + \lambda L_{CR}], \quad (2)$$

where $L(Y)$ is the sum of L1 loss and adversarial loss

$$L(Y) = \text{ReLU}(\mathbb{1} - D(Y \circ M + U)) + \beta \|Y - X\|_1. \quad (3)$$

Here, L_{CR} denotes CR loss, which will be elaborated in Section 4. λ is set equal to 0.5. β is set equal to 1.5.

4. Contextual reconstruction

Revisit contextual attention. As we have discussed in Sec. 1, for a generator with attention-based patch-borrowing operations, inappropriately borrowed patches will lead to artifacts in the inpainted image. The artifacts are produced based on the features from the reference regions, and thus resemble their appearance in image space. It also can be expected that moving these patches in the missing region will not result in a reasonable image, as has been shown in Fig. 1 (b). On the other hand, a visually realistic image can

be reconstructed with contextual patches if the reference regions are chosen correctly.

Inspired by this observation, we propose a CR loss to encourage a network to find optimal reference regions by minimizing the L1 and adversarial loss of an auxiliary result composed by image patches in known region, *i.e.* as Fig. 1 (b) and (d) was produced but is made differentiable by exploiting softmax and an autoencoder-like auxiliary network. Furthermore, unlike previous approaches which integrate a patch-borrowing operation in the generator, CR loss distills the information of the optimal contextual patches into an attention-free generator. As illustrated in Fig. 3, it does not directly involve the generation of inpainted images and affects the network only during training for learning better features.

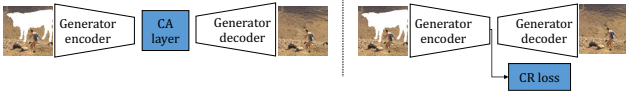


Figure 3: Comparison of the usage of CA layer and CR loss.

Contextual reconstruction loss. The overall training system with CR loss is shown in Fig. 5, where the inpainting model takes an incomplete image and the binary mask of the missing regions as input and outputs the inpainted image. The training system with CR loss consists of a similarity encoder and an auxiliary encoder-decoder type network and re-uses the inpainting loss defined in Sec. 3. The similarity encoder takes the generator feature as input and encodes the similarity among image regions. The auxiliary encoder-decoder network produce an auxiliary image in which the known regions are unchanged while the missing regions are filled with similar known regions based on the similarity provided by the similarity encoder. CR loss of the generator feature is defined to be the inpainting loss (*i.e.* L1 and adversarial loss) of the auxiliary image. By minimizing CR loss, the generator features are encouraged to be close to the known image features of the smallest inpainting loss. In what follows, we elaborate on the formal definition and explanation for this loss.

For the convenience of the description of the proposed method, here we define the global and patch-wise perspective of a CNN. If we view an image U passed through a CNN as the combination of square patches u_1, u_2, \dots , a convolution layer feature map $F(U)$ of this image U can be seen as the combination of local function of image patches $f(u_1), f(u_2), \dots$. Hereafter an upper case letter, *e.g.* $F(U)$, will represent a whole image or feature map with the corresponding lower case one for a patch in it *e.g.* $f(u_i)$. Each image patch can correspond to feature patches of various sizes in different layers of a CNN. For example, for the layers shown in Fig. 4, an $N \times N$ image patch u centered at (x, y) corresponds to an $\frac{N}{2} \times \frac{N}{2}$ feature patch $f_2(u)$ centered

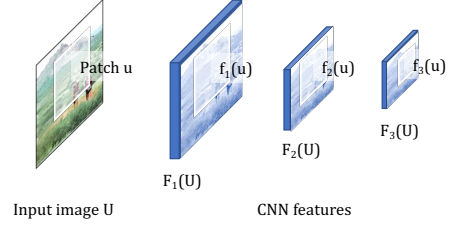


Figure 4: If view an input image as combination of patches, corresponding feature patch in a layer l can be seen as a representation $f_l(u)$ of an image patch u .

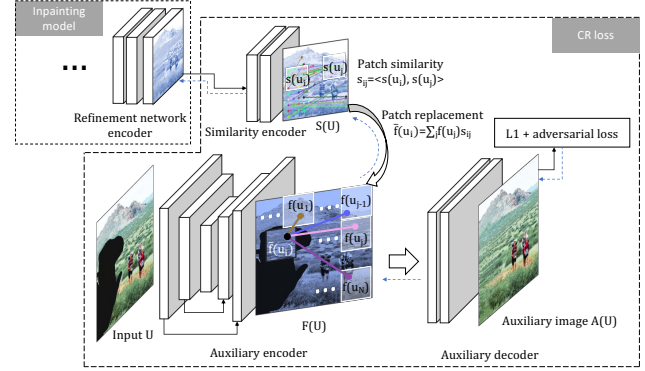


Figure 5: Training system with the proposed contextual reconstruction loss. The dashed blue lines indicates the gradient back-propagation flow.

at $(\frac{x}{2}, \frac{y}{2})$ in the second layer and an $\frac{N}{4} \times \frac{N}{4}$ feature patch $f_3(u)$ centered at $(\frac{x}{4}, \frac{y}{4})$ in the third layer. They can be seen as different representations of the same patch. Therefore, we can take the similarity of a pair of feature patches $f_l(u_i), f_l(u_j)$ at a layer l and use it as a similarity s_{ij} of the corresponding image patch or feature patches at arbitrary other layers. We take the pair-wise cosine similarity of feature patches $\langle s(u_i), s(u_j) \rangle$ at the last layer of the similarity encoder as s_{ij} :

$$s_{ij} = \frac{s(u_i)^\top s(u_j)}{\|s(u_i)\| \cdot \|s(u_j)\|}, \quad (4)$$

where the feature patches $s(u_i), s(u_j)$ are viewed as vectors when taken the inner product. It can be implemented as processing the feature map with patches extracted from itself as convolution filters as described in [28].

Before the feature of the auxiliary encoder is passed through the auxiliary decoder, each feature patch is replaced with a weighted sum of patches in the known region with the softmax of the similarity provided by the similarity encoder as weight. Let $f(u_i)$ be the auxiliary encoder feature of image patch u_i , $\bar{f}(u_i)$ the feature of u_i after patch replacement is obtained as follows,

$$\bar{f}(u_i) = \sum_{j \in \mathcal{V}} \text{softmax}(\alpha s_{ij}) f(u_j), \quad (5)$$

where \mathcal{V} represents the index set of patches in the known region. α is set equal to 10. It can be implemented as transposed convolution which processes the similarity maps with patches extracted from the auxiliary encoder features as filters after dropout with the similarity to all patches in the missing region.

Then the feature map after the patch replacement is translated into an auxiliary image by the auxiliary decoder. Let $\bar{F}(U)$ be the feature map consisting of feature patches $\bar{f}(u_i)$ given in Eqn. 5, the auxiliary image $Aux(U)$ of the input image U is obtained as follow,

$$Aux(U) = H(\bar{F}(U)), \quad (6)$$

where $H()$ represents the auxiliary decoder connected to the auxiliary encoder. Based on this discussion, the contextual reconstruction loss L_{CR} is defined as the inpainting loss $L()$ (Eqn. 3) of the auxiliary image:

$$L_{CR} = L(Aux(U)). \quad (7)$$

Assumes that the auxiliary decoder inverts the auxiliary decoder for known regions. Then the auxiliary encoder-decoder approximately conducts the process of producing Fig. 1 (b) and (d) described in Sec. 1: for each patch u_i in the missing regions, the most similar patch u_{i^*} from the known regions, *i. e.* $i^* = \arg \max_j s_{ij}$, is moved to the position of u_i . This results in a simplified view of the training system shown in Fig. 6, which is an analogue of learning to solve a jigsaw puzzle. To minimize the loss of the auxiliary image, proper known patches should be chosen and moved to the right place. Furthermore, as the reference known patches are selected according to the similarity of generator features, it requires the generator feature in the missing region to be closest to the proper known region. Since the searching of known patches is throughout the whole image, distant relationship can be captured. A more detailed explanation will be given in the following section.

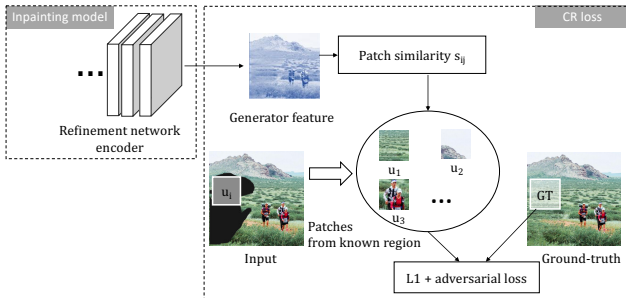


Figure 6: A simplified view of the training system.

Explanation. For a patch in the known region, *i. e.* $u_j, j \in \mathcal{V}$, since $\bar{f}(u_j) \approx f(u_j)$ as $\text{softmax}(\alpha s_{jj}) \approx 1$, the auxiliary image patch in the known region will be $a(u_j) = h(f(u_j)), j \in \mathcal{V}$. Note that the ground-truth of $a(u_j)$ is u_j

itself for $j \in \mathcal{V}$, the auxiliary encoder-decoder $H(F(U))$ will learn lazily to copy the input to the output so it is easy to make the auxiliary decoder invert the auxiliary encoder: $h(f(u_j)) = u_j$ for $j \in \mathcal{V}$. To make it easier, we put skip connection from shallow layers to the deeper layers of the auxiliary encoder, as indicated in Fig. 5.

Assume that softmax in Eqn. 5 to be “hard”, then the feature patch after replacement can be approximated as:

$$\bar{f}(u_i) = f(u_{i^*}) \text{ where } i^* = \arg \max_{j \in \mathcal{V}} s_{ij}. \quad (8)$$

Then a patch in the auxiliary image is:

$$a(u_i) = h(f(u_{i^*})) = u_{i^*} \text{ where } i^* = \arg \max_{j \in \mathcal{V}} s_{ij}. \quad (9)$$

Resembling the translation between the global and patch-wise perspective of a CNN introduced earlier, the inpainting loss for the whole image also can be distributed to each patch. Thereby the inpainting loss of the auxiliary image $L(Aux(U))$ can be seen as a sum over local loss of patches: $L(Aux(U)) = \sum_i l_i(a(u_i))$ (We differentiate the local loss of different patches as the ground-truth for each patch is different, which means $L(Aux(U))$ is in fact $L(Aux(U); X) = \sum_i l(a(u_i); x_i)$ and it is denoted as $l_i(a(u_i))$ for short). We have argued earlier that the loss on known regions can be minimized with the auxiliary encoder-decoder copying the input to the output. So what count for $L(Aux(U))$ are the patches in the missing region:

$$L(Aux(U)) = \sum_{i \in \mathcal{V}^c} l_i(u_{i^*}) \text{ where } i^* = \arg \max_{j \in \mathcal{V}} s_{ij}. \quad (10)$$

This is a sum over local inpainting loss of known image patches with the largest similarity to the generated patches in the missing region. Note that the number of known patches in an image is limited, so for each local inpainting loss $l_i()$ there must be one patch u_{i^0} that has the smallest loss among all the known patches, *i. e.* $i^0 = \arg \min_{j \in \mathcal{V}} l_i(u_j)$. The minimum of CR loss is achieved when $i^* = i^0$. In other words, when the generator features in missing regions are closest to the features in known regions which have the smallest inpainting loss.

In summary, to minimize the CR loss, two conditions are required. First, appropriate reference patches are selected from the known region. Second, feature patches in missing regions are closest to the corresponding reference patches. Compared with CA layer which searches the known patches using heuristics with no guarantee on the reference patches it reaches, CR loss can find more reasonable reference patches. This will be supported by quantitative experimental results in Sec. 5.

5. Experimental Results

We implement our method with Python and PyTorch. Detailed network architectures and code can be found in

the supplementary material. We train the models using Adam [8] optimizer with the learning rate of 0.0001. To show the effectiveness of the CR loss, we train the network shown in Fig. 2 with and without the CR loss on the Places2 training set. We add images from the salient object segmentation dataset [23] following [31, 23]. Square masks, irregular masks [29] and object-shaped masks [31] are randomly switched at every mini-batch to be used for creating missing regions. Training samples are cropped to 256×256 and missing regions are put at random positions. The obtained models are denoted as Baseline and Baseline+CR, respectively. To compare the proposed CR loss with CA layer, we also train a network with the same architecture as DeepFillv2 [29] but remove the CA layer and apply CR loss instead. The training protocol for this network is made to be the same as the official implementation of DeepFillv2 [29], where Places2 training set with square and irregular masks are used for training; β in Eqn. 3 is set equal to 1 as in the official implementation of DeepFillv2. The obtained model is denoted as DeepFillv2-CA+CR. All evaluations are conducted on the same platform: an Ubuntu machine with a 3GHz Intel i7-9700F CPU, 32GB memory, 256GB swap space, and a GPU NVIDIA RTX2080S.

5.1. Comparison with state-of-the-art

We compare our methods with the following state-of-the-art methods: PENNet [30], DeepFillv2 [29], ReThink [15], HiFill [27]. We use their official implementation and the models trained on Places2 training set provided by the authors of corresponding papers. We use L1 error, PSNR and SSIM to measure the performance quantitatively. For comparison on visual quality, we show the inpainting results and the results of a subject evaluation conducted by human raters.

Quantitative evaluation. Table 1 and Table 2 show quantitative comparisons of our method with state-of-the-art methods on Places2 validation set containing 36500 images and ImageNet validation set containing 50000 images. All images are cropped to 256×256 with missing regions of different shapes (128×128 squares, object and irregular shapes) at random positions. From these table, we can see that our method has a smaller L1 error, larger PSNR and SSIM than the existing methods in most cases, which indicates that the inpainting results corresponding to our method are closer to the ground-truth.

Visual quality. Fig. 7 shows the visual comparison of our method and existing methods. We can see from the figure that the missing region recovered by our method is more visually coherent with surrounding known regions. This figure implies the effectiveness of CR loss in exploiting structured information in known regions. Furthermore, unlike a previous method PENNet which has high performance in quantitative evaluation but tends to generate blurry images,

the results of our method are more visually realistic.

User study. We invite 9 human raters to conduct a user study on 45 images randomly sampled from the Places2 validation set. Every 1/3 of them are corrupted with missing regions of different shapes, *i. e.* square, irregular and object shaped, at random positions. Each time the incomplete image and the results by different methods are presented to the raters in random order. The raters are asked to select one best result. The number of user preference for all methods are shown in the last column of Table 1. From the user study results, we can see that the results of our method are preferred by the human raters most frequently.

High-resolution inpainting. The efficiency of the proposed method makes it potentially capable of processing high-resolution input. We make a simple adjustment to the network to adapt to high-resolution input: running the coarse network at $1/2$ the input size and interpolate its output back to the original resolution before passed through the refinement network. Fig. 8 shows high-resolution inpainting results by our adjusted inpainting model and HiFill [27] which was proposed for inpainting at high-resolution. More results and comparison at high-resolution can be found in the supplementary material.

5.2. Ablation study

Inpainting results. Fig. 9 shows a visual comparison of the results obtained by the baseline model (Baseline), the baseline model trained with the CR loss (Baseline+CR), and the results of DeepFillv2 with CA layer. It can be seen that both CA layer and CR loss enable the inpainting model to capture a long term relationship among image regions and the distant known regions while while the baseline model tends to be short-sighted which propagates the most nearby features. For instance, in the second example, both the model with CA layer and CR loss correctly capture the grid pattern while the baseline model just extends the lines. The first two rows of Table 3 show the quantitative comparison of the baseline model (Baseline) and the baseline model with the CR loss (Baseline+CR). The last two rows show the comparison of the original DeepFillv2 and the model obtained by removing CA layer and applying the CR loss (DeepFillv2-CA+CR) with network architectures and training protocols unchanged. From this table, we can see that the CR loss can improve the quantitative performance of different models, and DeepFillv2 with CR loss yields better results than the official implementation with CA layer.

Reference patch. To quantify the difference of the reference regions used by the CR Loss and CA layer, we measure the average L1 error, PSNR and SSIM of the images constructed by moving each reference patch to the corresponding position in the missing region like piecing together a jigsaw. The quantitative comparison is shown in Table 5, from which we can see that the images pieced by the CR loss

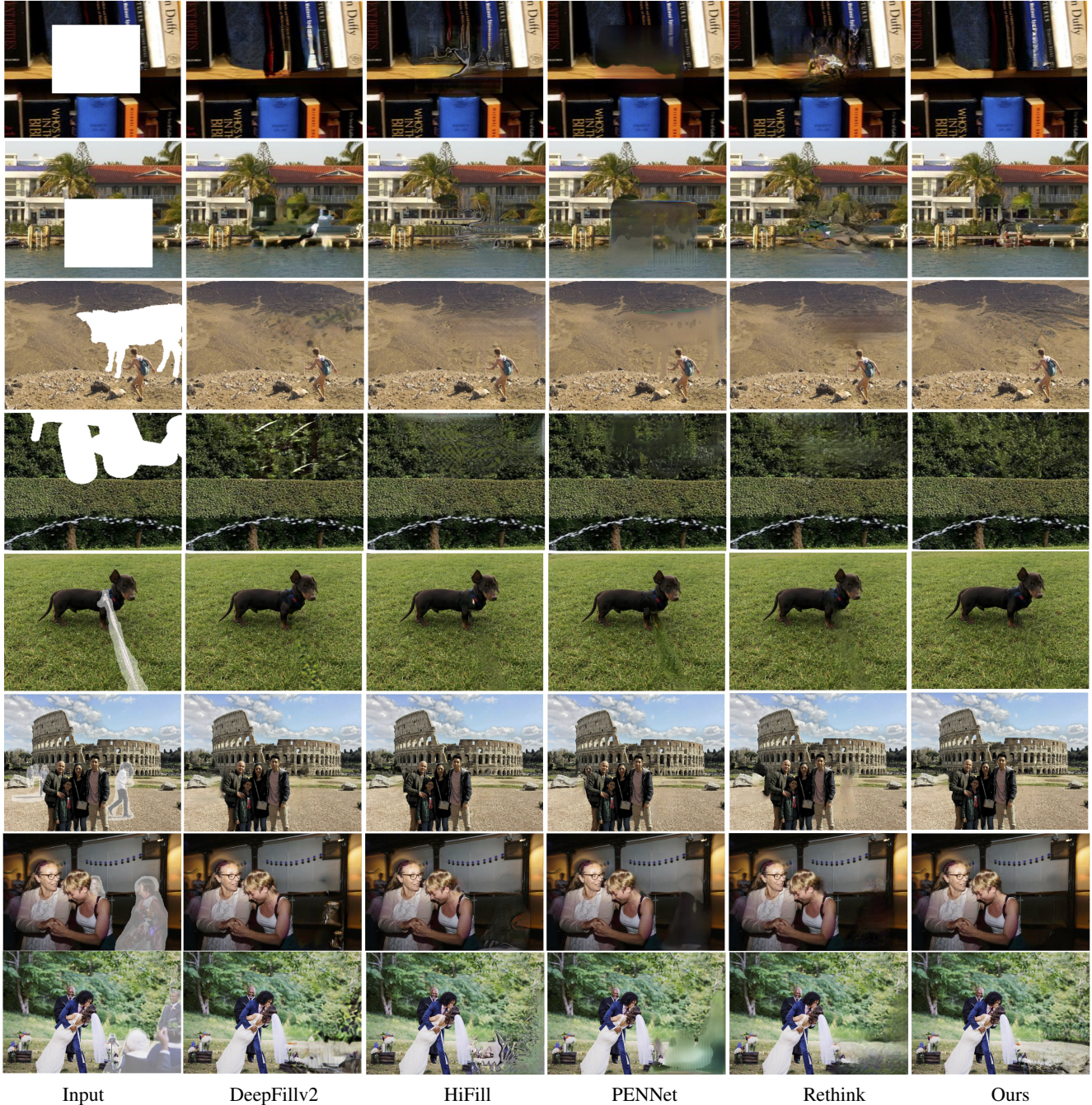


Figure 7: Visual comparison of inpainting results of our method and other methods. Zoom-in to see the details. Images are compressed due to file size limitation. More results and comparison at high-resolution can be found in the supplementary material.

Table 1: Quantitative evaluation results on the Places2 validation set. The best scores are in bold.

Method	Square holes				Irregular holes				Object holes				User preference
	FID	L1 error	PSNR	SSIM	FID	L1 error	PSNR	SSIM	FID	L1 error	PSNR	SSIM	
DeepFillv2 [29] _{ICCV'19}	3.313	0.0290	22.72	0.8429	1.387	0.0178	27.03	0.8947	1.905	0.0225	27.95	0.8887	88
Rethink [15] _{ECCV'20}	11.03	0.0307	22.30	0.8305	6.877	0.0194	26.13	0.8796	5.539	0.0237	27.07	0.8829	13
PENNet [30] _{CVPR'19}	14.67	0.0284	23.28	0.8402	10.02	0.0227	25.35	0.8699	5.539	0.0251	26.28	0.8780	6
HiFill [27] _{CVPR'20}	18.84	0.0329	21.35	0.8103	6.887	0.0210	25.48	0.8712	7.819	0.0267	26.14	0.8647	27
Ours	2.908	0.0263	23.36	0.8462	1.231	0.0157	27.79	0.9002	1.724	0.0202	28.79	0.8929	271

Table 2: Quantitative evaluation results on the ImageNet validation set. The best scores are in bold.

Method	Square holes				Irregular holes				Object holes			
	FID	L1 error	PSNR	SSIM	FID	L1 error	PSNR	SSIM	FID	L1 error	PSNR	SSIM
DeepFillv2 [29] _{ICCV'19}	6.739	0.0335	21.18	0.8200	2.228	0.0205	25.55	0.8765	3.257	0.0250	26.51	0.8739
Rethink [15] _{ECCV'20}	13.49	0.0347	21.13	0.8137	6.604	0.0216	25.15	0.8682	5.778	0.0259	26.11	0.8723
PENNet [30] _{ICCV'19}	14.67	0.0316	22.09	0.8257	7.462	0.0243	24.53	0.8615	6.129	0.0269	25.48	0.8698
HiFill [27] _{CVPR'20}	16.92	0.0329	21.35	0.8103	5.519	0.0210	25.48	0.8712	6.611	0.0287	25.24	0.8570
Ours	5.813	0.0302	21.91	0.8269	1.707	0.0180	26.43	0.8855	3.047	0.0225	27.45	0.8808

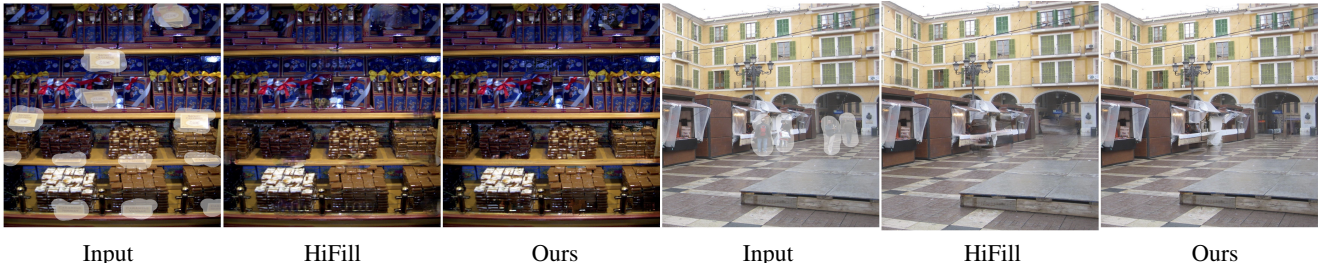


Figure 8: High-resolution results (1200×1600 , 1152×1536) compared with HiFill. Zoom in to see the details. Images are compressed due to file size limitation.

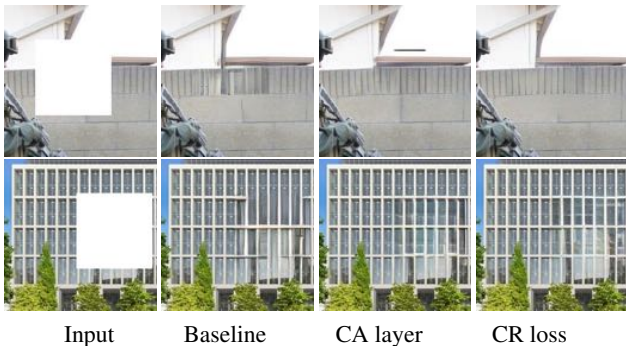


Figure 9: Effect of CA layer and CR loss. Both encourage to capture distant relation among image regions.

Table 3: Comparison with baseline and CA layer on the Places2 validation set.

Method	Square holes			Irregular holes		
	L1 error	PSNR	SSIM	L1 error	PSNR	SSIM
Baseline	.0275	22.92	.8386	.0171	27.03	.8920
Baseline+CR	.0263	23.36	.8462	.0157	27.79	.9002
DeepFillv2 (with CA)	.0290	22.72	.8429	.0178	27.03	.8947
DeepFillv2-CA+CR	.0262	23.37	.8447	.0158	27.77	.8984

have a smaller L1 error, larger PSNR and SSIM, which indicates that the CR loss can find the reference regions closer to the ground-truth.

Efficiency. Since CR loss gets rid of the heavy pair-wise similarity computation in the inference stage, the obtained model is computationally efficient. Table 5 shows time complexity and the running time of the networks of the same architecture with the CR loss and CA layer at different resolutions. The efficiency advantages become increasingly evident when the resolution increases.

Table 4: L1 error, PSNR, SSIM of the image pieced with reference patches selected by the CR loss and CA layer on the Places2 validation set with square holes.

Method	Square holes			Irregular holes		
	L1 error	PSNR	SSIM	L1 error	PSNR	SSIM
DeepFillv2 (with CA)	.0331	21.37	.8104	.0232	24.39	.8562
DeepFillv2-CA+CR	.0315	21.77	.8135	.0205	25.27	.8629
Baseline+CR	.0310	21.90	.8152	.0202	25.39	.8643

Table 5: Comparison of time complexity and the running time measured at different resolution. We are not able to evaluate contextual attention layer at 2080×2048 on GPU due to limited GPU memory size and report the comparison on CPU instead. N is the number of pixels in the input image. T_s^{GPU} and T_s^{CPU} represent the running time (in seconds) on GPU and CPU at $s \times s$ resolution, respectively.

	Complexity	T_{512}^{GPU}	T_{1024}^{GPU}	T_{2048}^{GPU}	T_{2048}^{CPU}
CA layer	$O(N^2)$	0.063	0.371	Fail	266
CR loss	$O(N)$	0.047	0.179	0.720	28.9

6. Conclusion & future work

This paper presents a generative image inpainting model with an auxiliary contextual reconstruction branch. A learnable contextual reconstruction loss (CR loss) is designed to jointly optimize a reference-based reconstructor with the inpainting generator to teach the path-borrowing behavior to an attention-free generator. Experiments show that our inpainting model with the proposed CR loss compares favourably against state-of-the-art in terms of quantitative measurements and visual quality. CR loss can be extended for distilling more information to an inpainting model through the reconstruction of higher-level representation of images, *e.g.* semantic layout. This might be an interesting topic for future work.

References

- [1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001. [2](#)
- [2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. 28(3):24, 2009. [1](#), [2](#)
- [3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *The 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000. [2](#)
- [4] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *The 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346. ACM, 2001. [1](#), [2](#)
- [5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. [2](#)
- [6] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017. [1](#), [2](#)
- [7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. [3](#)
- [8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [6](#)
- [9] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. 24(3):795–802, 2005. [1](#), [2](#)
- [10] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *European Conference on Computer Vision*, pages 377–389. Springer, 2004. [1](#)
- [11] Chu-Tak Li, Wan-Chi Siu, Zhi-Song Liu, Li-Wen Wang, and Daniel Pak-Kong Lun. Deepgin: Deep generative inpainting network for extreme image inpainting. In *European Conference on Computer Vision*, pages 5–22. Springer, 2020. [2](#)
- [12] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7760–7768, 2020. [2](#)
- [13] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [2](#)
- [14] 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 *European Conference on Computer Vision*, 2018. [1](#), [2](#)
- [15] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *European Conference on Computer Vision*, 2020. [3](#), [6](#), [7](#), [8](#)
- [16] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *IEEE International Conference on Computer Vision*, 2019. [2](#), [3](#)
- [17] Yuqing Ma, Xianglong Liu, Shihao Bai, Lei Wang, Dailan He, and Aishan Liu. Coarse-to-fine image inpainting via region-wise convolutions and non-local correlation. In *IJ-CAI*, pages 3123–3129, 2019. [3](#)
- [18] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. [2](#)
- [19] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3500–3509, 2017. [1](#)
- [20] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [1](#), [2](#)
- [21] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *European Conference on Computer Vision (ECCV)*, 2018. [3](#)
- [22] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25:341–349, 2012. [2](#)
- [23] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [2](#), [3](#), [6](#)
- [24] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *European conference on computer vision (ECCV)*, 2018. [3](#)
- [25] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [2](#)
- [26] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017. [2](#)
- [27] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [6](#), [7](#), [8](#)
- [28] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. [1](#), [2](#), [4](#)
- [29] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *IEEE International Conference on Computer Vision*, 2019. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)

- [30] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1494, 2019. [2](#), [6](#), [7](#), [8](#)
- [31] 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 *European Conference on Computer Vision*, 2020. [2](#), [3](#), [6](#)