Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting

Yanhong Zeng[1,2]*, Jianlong Fu[3], Hongyang Chao[1,2], Baining Guo[3]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, P.R. China
[2]The Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, Guangzhou, P.R. China
[3]Microsoft Research, Beijing, P.R. China
zengyh7@mail2.sysu.edu.cn, {jianf,bainguo}@microsoft.com, isschhy@mail.sysu.edu.cn

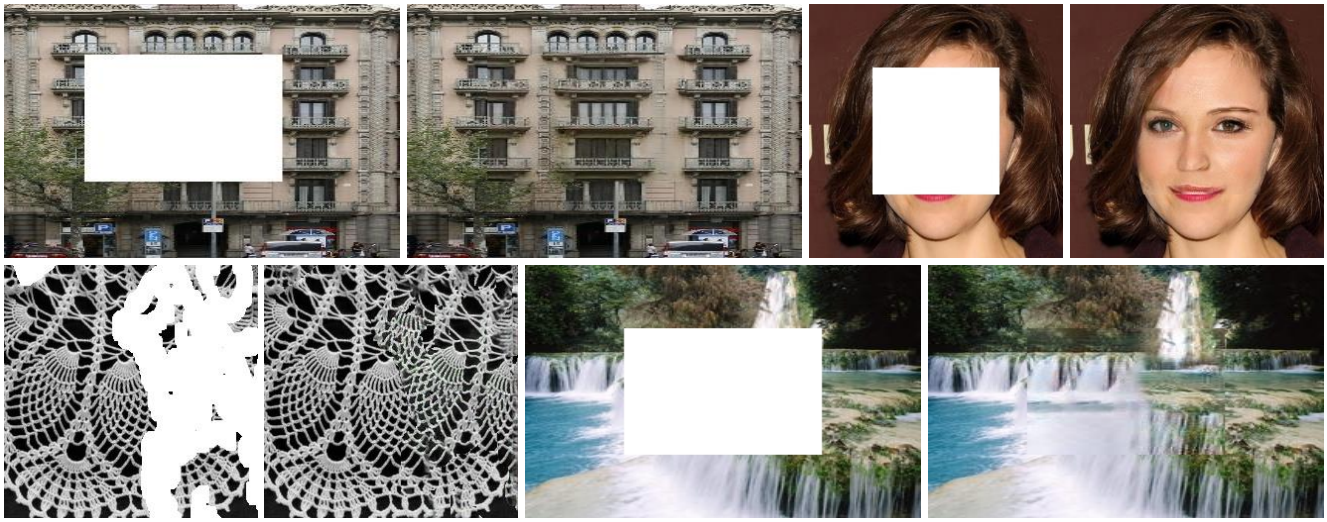


Figure 1: High-quality image inpainting results generated by the proposed **P**yramid-context **EN**coder **Net**work (PEN-Net). In each pair, the left is a damaged image masked in white, and the right is the result of image inpainting. PEN-Net shows excellent performance on a variety of images, including facades, natural scene, face and texture. [Best viewed in color]

Abstract

*High-quality image inpainting requires filling missing regions in a damaged image with plausible content. Existing works either fill the regions by copying image patches or generating semantically-coherent patches from region context, while neglect the fact that both visual and semantic plausibility are highly-demanded. In this paper, we propose a **P**yramid-context **EN**coder **Net**work (PEN-Net) for image inpainting by deep generative models. The PEN-Net is built upon a U-Net structure, which can restore an image by encoding contextual semantics from full resolution input, and decoding the learned semantic features back into images. Specifically, we propose a pyramid-context encoder, which progressively learns region affinity by attention from a high-level semantic feature map and transfers the learned attention to the previous low-level feature map. As the missing content can be filled by attention transfer from deep to shallow in a pyramid fashion, both visual and semantic coherence for image inpainting can be ensured. We further propose a multi-scale decoder with deeply-supervised pyramid losses and an adversarial loss. Such a design not only results in fast convergence in training, but more realistic results in testing. Extensive experiments on various datasets show the superior performance of the proposed network.*

1. Introduction

Image inpainting aims at filling missing pixels in a damaged image given a corresponding mask [2]. This task has drawn great attention and become a valuable and active research topic for decades [5, 12, 17], because high-quality

*This work was performed when the first author was visiting Microsoft Research as a research intern.

image inpainting can benefit a broad range of applications, such as old photo restoration, object removal, and so on.

High-quality image inpainting usually requires synthesizing not only visually-realistic but semantically-reasonable content for missing regions [3, 5, 28, 29, 31]. Existing approaches can be roughly divided into two groups. As shown in Table 1, the first group inspired by texture synthesis techniques attempts to fill regions at image-level [1, 5, 22]. Specifically, such approaches usually sample and paste full image resolution patches from source images into missing regions, which allows synthesizing results with details. However, as the lack of high-level understanding of an image, such approaches often fail in generating semantically-reasonable results. To solve this problem, the second group of approaches proposes to encode the semantic context of an image into a latent feature space by deep neural networks and then generate semantic-coherent patches by generative models [13, 17, 31]. However, it remains challenging to generate visually-realistic results from a compact latent feature, as full image resolution details can be usually smoothed by stacked convolutions and poolings.

To ensure that both visual and semantic coherence can be satisfied, we propose to fill regions at both image and feature levels. First, we adopt a U-Net [19] structure as our backbone, which can encode the context from low-level pixels to high-level semantic features and decode the features back into an image. Specifically, we propose a **P**yramid-context **EN**coder **Net**work (PEN-Net) with three tailored key components, *i.e*., a pyramid-context encoder, a multi-scale decoder, and an adversarial training loss, to boost the capacity of U-Net in image inpainting. Second, once the compact latent features have been encoded from images, the pyramid-context encoder fills regions from high-level semantic features to low-level features (with richer details) in a pyramid pathway before decoding. To this end, we propose an **A**ttention **T**ransfer **N**etwork (ATN) to learn region affinity between patches inside/outside missing regions in a high-level feature map, and then transfer (*i.e*., weighted copy by affinity) relevant features from outside into inside regions of previous feature map with higher resolution. Third, the proposed multi-scale decoder takes as input the reconstructed features from ATNs through skip connections and the latent features for final decoding. The PEN-Net is optimized by minimizing deeply-supervised pyramid L1 losses and an adversarial loss.

To the best of our knowledge, the proposed PEN-Net is the first work that is able to fill missing regions at both image-level and feature-level for image inpainting. we highlight our contributions as follows:

- **Cross-layer attention transfer**. We propose a novel network, ATN, to learn region affinity from high-level feature maps (e.g., the compact latent features in the encoder). The resultant affinity map can guide feature transfer in adjacent low-level layers in an encoder.
- **Pyramid filling**. Our model can fill holes multiple times (depends on the depth of the encoder) by repeating using ATNs from deep to shallow, which can restore an image with more fine-grained details.

Category	Method	Semantic	Details
image level	PatchMatch[1], Region filling[5]		✓
feature level	GL[9], PConv[13], GntIpt[31]	✓	
Ours		✓	✓

Table 1: Two groups of typical approaches for image inpainting. PatchMatch [1] and Region filling [5] ensure that patches with more details can be used for filling, while GL [9], Pconv [13] and GntIpt [31] can generate semantic-coherent results. Compared with those methods, our approach can satisfy both semantic and visual requirements.

2. Related Work

Image inpainting by patch-based methods. Patch-based methods were first proposed for texture synthesis [6, 7]. They were then applied in image inpainting to fill missing regions at image level [24]. They usually sample and paste similar patches from database or undamaged surroundings into missing regions based on distance metrics between patches (*e.g*. Euclidean distance, SIFT distance [15], *etc*.). Bertalmio *et al*. proposed to combine patch-based texture synthesis techniques with diffusion-based propagation under image decomposition [3]. A number of approaches try to improve performance by providing better filling order or optimal patches [5, 22, 27]. PatchMatch was proposed for quickly finding similar matches between image patches [1]. Patch-based methods for image inpainting are able to generate sharp results similar with context. However, it's hard to generate semantically-reasonable results by patch-based methods, due to the lack of high level understanding of images.

Image inpainting by deep generative models. Deep generative models for image inpainting usually encode an image into a latent feature, fill missing regions at the feature-level, and decode the feature back into an image. Promising results have been achieved by deep generative models recently. Based on deep feature learning and adversarial training, Context Encoder, one of the first deep generative models, is able to give reasonable results for semantic hole-filling [17]. Guidance loss was introduced to make the feature maps generated in decoder as close as possible to the feature maps of ground-truth generated in encoder [28]. Dilated convolutions [30] were introduced to increase recep-

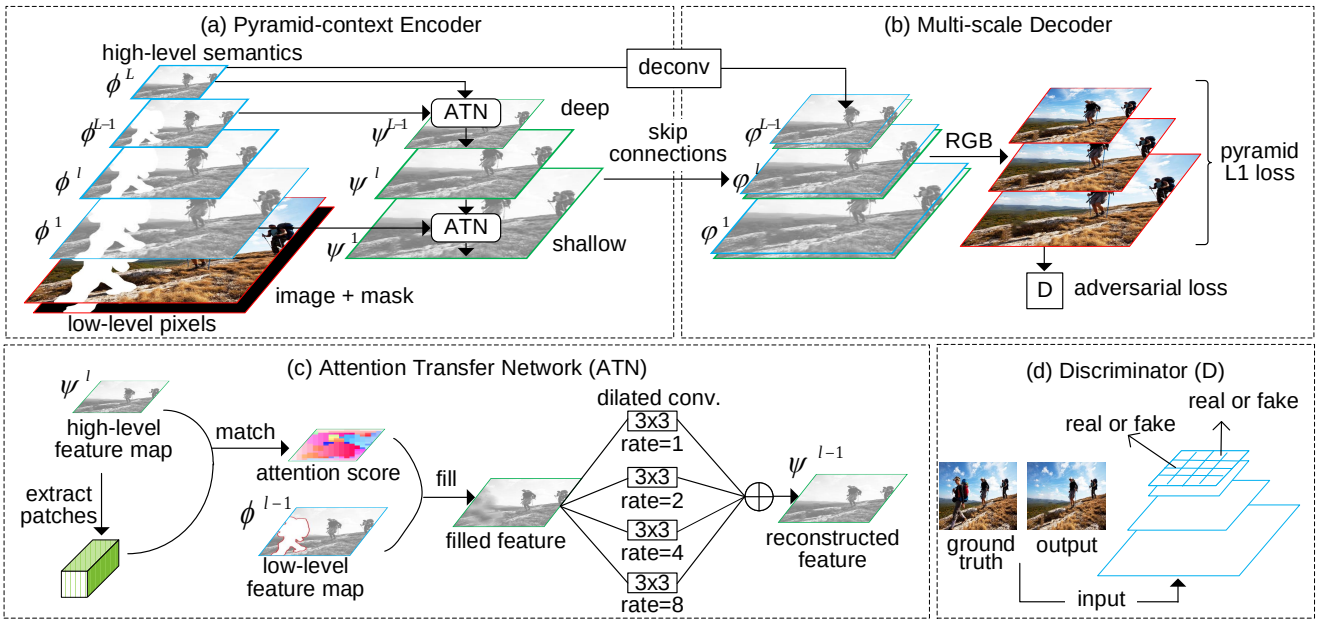


Figure 2: The **P**yramid-context **En**coder **Net**work (PEN-Net) is proposed to boost the capability of U-Net in image inpainting with three tailored components, *i.e.*, a pyramid-context encoder (a), a multi-scale decoder (b), and an adversarial training loss (d). First, once the compact latent feature has been encoded, the pyramid-context encoder further improves the encoding effectiveness by filling regions from high-level feature maps to low-level feature maps (with richer details) through the proposed **A**ttention **T**ransfer **N**etwork (ATN) (c). Second, the multi-scale decoder takes as input the reconstructed features from ATNs through skip connections and the latent features for decoding. Finally, the decoder decodes the features back into an image. The whole network is optimized by minimizing pyramid L1 losses and an adversarial loss. [Best viewed in color.]

tive field in completion network by Iizuka *et al*. [9]. Special convolution operations such as PConv [13] and ShCNN [18] were designed for eliminating the effects caused by the placeholder values in masked regions in an image. Contextual attention layer [31] and Patch-swap layer [21] were proposed for filling missing pixels with similar patches from undamaged regions at high-level feature maps. Inspired by image stylization, MNPS proposed to optimize texture details by using a pre-trained classification network during inference [29]. Isola *et al*. try to solve image inpainting by a general image translation framework [10]. Leveraging high-level semantic feature learning, deep generative models are able to generate semantically-coherent results for the missing regions. However, it remains challenging to generate visually-realistic results from a compact latent feature.

3. Pyramid-context Encoder Network

The Pyramid-context Encoder Network (PEN-Net) consists of three parts (as shown in Figure 2), *i.e.*, a pyramid-context encoder (a), a multi-scale decoder (b) and a discriminator (d). The PEN-Net is built upon a U-Net structure, which can encode a damaged image with mask from full input resolution pixels into a compact latent features and decode the features back into an image.

As the compact latent features encode the semantics of the context, the pyramid-context encoder can further improves the encoding effectiveness by filling missing regions from the compact latent feature to low-level features (with higher resolution and richer details). It fills holes by repeating using the proposed **A**ttention **T**ransfer **N**etwork (ATN) (c) multiple times (according to the depth of the encoder) before decoding. Specifically, an ATN learns region affinity between patches inside/outside missing regions from high-level semantic features, and the learned attention is transferred to fill regions (*i.e.*, weighted copy from the context by affinity) in its previous feature map with higher resolution. Multi-scale information is further aggregated to refine the filled features by four groups of dilated convolutions with different rates in an ATN. Finally, the multi-scale decoder takes as input the reconstructed features from ATNs through skip connections and the latent features for decoding. In addition to an adversarial loss, pyramid L1 losses are used to progressively refine the prediction output by the decoder at all scales.

We describe details of the pyramid-context encoder and the ATN in Section 3.1. The multi-scale decoder and pyramid L1 losses are introduced in Section 3.2 followed by adversarial training loss described in Section 3.3.

3.1. Pyramid-context encoder

Pyramid-context encoder In order to improve the effectiveness of encoding, the pyramid-context encoder is proposed for filling missing regions before decoding. Once a compact latent feature is learned, the pyramid-context encoder fills regions from high-level semantic features to low-level features (with higher resolution) by repeating using the proposed ATNs in a pyramid fashion. Under the assumption that pixels with similar semantics should have similar details, an ATN is applied at each level to learn region affinity from high-level semantic features, thus the learned region affinity can further guide feature transfer inside/outside missing regions in an adjacent layer with higher resolution.

Given a pyramid-context encoder of L layers, we denote the feature maps from deep to shallow as $\phi^L, \phi^{L-1}, ..., \phi^1$ as shown in (a) of Figure 2. The features constructed by ATNs in each layer from deep to shallow are denoted as:

$$\begin{aligned} \psi^{L-1} &= f(\phi^{L-1}, \phi^L), \\ \psi^{L-2} &= f(\phi^{L-2}, \psi^{L-1}), \\ &\cdots, \\ \psi^1 = f(\phi^1, \psi^2) &= f(\phi^1, f(\phi^2, ...f(\phi^{L-1}, \phi^L))), \end{aligned} \tag{1}$$

where we denote the operation of the ATN as f. By such a cross-layer attention transfer and pyramid filling mechanism, both visual and semantic coherence for the missing regions can be ensured. The details of f (*i.e.*, ATN) are introduced as below.

Attention Transfer Network We follow state-of-the-art approaches to fill missing regions by using attention [21, 28, 31]. The attention is usually obtained by region affinity between patches (usually 3×3) inside/outside missing regions, thus relevant features outside can be transferred (*i.e.*, weighted copy from the context by affinity) into inside regions. As shown in (c) of Figure 2, the ATN first learns region affinity from a high-level feature map, ψ^l. It extracts patches from ψ^l and calculate the cosine similarity between patches inside and outside missing regions:

$$s^l_{i,j} = \langle \frac{p^l_i}{\left\| p^l_i \right\|_2}, \frac{p^l_j}{\left\| p^l_j \right\|_2} \rangle, \tag{2}$$

where p^l_i is the i-th patch extracted from ψ^l outside mask, p^l_j is the j-th patch extracted from ψ^l inside the mask. Then softmax is applied on the similarities to obtain the attention score for each patch:

$$\alpha^l_{j,i} = \frac{\exp(s^l_{i,j})}{\sum_{i=1}^{N} \exp(s^l_{i,j})}. \tag{3}$$

After obtaining the attention score from a high-level feature map, the holes in its adjacent low-level feature map can be filled with context weighted by the attention score:

$$p^{l-1}_j = \sum_{i=1}^{N} \alpha^l_{j,i} \ p^{l-1}_i, \tag{4}$$

where p^{l-1}_i is the i-th patch extracted from ϕ^{l-1} outside masked regions, and p^{l-1}_j is the j-th patch to be filled in missing regions. After calculating all patches, we can finally obtain a filled feature ψ^{l-1} by attention transfer from ψ^l. In particular, all these operations can be formulated into convolution operations for end-to-end training [31].

We propose to further refine the filled features in an ATN as shown in (c) of Figure 2. Specifically, multi-scale contextual information can be aggregated by four groups of dilated convolutions with different rates. Such a design ensures structure coherence with context in the final reconstructed features, which improves the inpainting results in testing.

3.2. Multi-scale decoder

Multi-scale decoder The proposed multi-scale decoder takes as input the reconstructed features from ATNs through skip connections and the latent features from the encoder. We denote the feature maps generated by the multi-scale decoder as $\varphi^{L-1}, \varphi^{L-2}, ..., \varphi^1$ from deep to shallow, which are obtained as follows:

$$\begin{aligned} \varphi^{L-1} &= g(\psi^{L-1} \oplus g(\phi^L)), \\ \varphi^{L-2} &= g(\psi^{L-2} \oplus \varphi^{L-1}), \\ &\cdots, \\ \varphi^1 &= g(\psi^1 \oplus \varphi^2), \end{aligned} \tag{5}$$

where g denotes transposed convolution operation, $\oplus$ denotes feature concatenation, and ψ^l is the reconstructed feature from an ATN in the l-th layer of the encoder.

On one hand, the reconstructed features generated by ATNs encode more low-level information for missing regions. Such a design enables the decoder to generate visually realistic results with fine-grained details. On the other hand, the features obtained from the compact latent features by convolutions are able to synthesize novel objects in missing regions, even when the objects cannot be found outside missing regions. Combining those two kinds of features, the decoder is able to synthesize novel objects with high coherence in semantics and textures with the context of the image. For example, the proposed decoder is able to synthesize eyes in human face images with both eyes masked.

Pyramid L1 losses We also propose deeply-supervised pyramid L1 losses to progressively refine the predictions for missing regions at each scale. Specifically, each pyramid loss is a normalized L1 distance between a prediction of specific scale and the corresponding ground truth:

$$L_{pd} = \sum_{l=1}^{L-1} \left\| x^l - h(\varphi^l) \right\|_1, \tag{6}$$

Figure 3: Qualitative comparisons with baselines on four datasets with different characteristics. In each row, the first image is the input with a large mask in the center (*i.e.*, 128×128), and the left images from left to right are the results generated by PatchMatch [1], GL [9], CA [31], PConv [13] and our model respectively. [Best viewed with zoom-in.]

where h denotes a 1×1 convolution which decodes φ^l into an RGB image with the same size, and x^l is the ground truth scaled to the same size as φ^l. The overall objective function incorporating pyramid L1 losses and an adversarial loss is described in the next section.

3.3. Adversarial training loss

As image inpainting is an ill-posed problem that there are many possible results for the missing regions, we use adversarial training to select the most realistic one. Adversarial training usually involves a generator (G) and a discriminator (D), which aims at achieving a Nash equilibrium, so that fake data generated by the generator cannot be distinguished from real data by the discriminator. As shown in (d) of Figure 2, the pyramid-context encoder and the multi-scale decoder form a generator, and we adopt PatchGAN [10] as our discriminator. Spectral normalization is used in the discriminator to stabilize the training [16].

We first define the final prediction from the generator as:

$$z = G(x \odot (1 - M), M) \odot M + x \odot (1 - M), \quad (7)$$

where x is the ground truth, $\odot$ is an element-wise multiplication, M is the mask where 1 labels missing regions and 0 labels context. The hinge version of the adversarial loss for the discriminator can be denoted as:

$$\begin{aligned} L_D = \mathbb{E}_{x \sim p_{data}} \left[max(0, 1 - D(x)) \right] \\ + \mathbb{E}_{z \sim p_z} \left[max(0, 1 + D(z)) \right], \end{aligned} \quad (8)$$

where $D(x)$ and $D(z)$ are the logits output from D. The adversarial loss for the generator can be denoted as:

$$L_G = -\mathbb{E}_{z \sim p_z} \left[D(z) \right]. \quad (9)$$

The whole PEN-Net is optimized by minimizing an adversarial loss and pyramid L1 losses defined in Section 3.2. We define the overall objective function as:

$$L = \lambda_G L_G + \lambda_{pd} L_{pd}. \quad (10)$$

4. Experiments

We evaluate the proposed network with baselines from both quantitative and qualitative aspects. Details of experimental settings are introduced in Section 4.1, and the experiments results are described in Section 4.2, followed by the analysis of the effectiveness of our model in Section 4.3.

4.1. Experimental settings

Datasets We conduct experiments on four datasets with different characteristics as below (details in Table 2):

- Facade [25]: a collection of highly-structured facades from different cities around the world.
- DTD [4], an evolving dataset of 47 kinds of describable textures collected in the wild.
- CELEBA-HQ [11], a high-quality version of the human face dataset from CELEBA [14].
- Places2 [32], a dataset that contains images of 365 scenes collected from the natural world.

Baselines We compare with the following baselines for their state-of-the-art performance:

- PatchMatch (PM) [1]: a typical patch-based approach, which copies similar patches from the surroundings.
- GL [9]: a generative model, which leverages both global and local discriminators for image completion.
- CA [31]: a two-stage inpainting model, which leverages contextual attention at high-level features.
- PConv [13]: a generative model, which proposes a special convolution layer for filling irregular holes.

Implementation details We use random blocks for training, following the experimental settings used by baselines [9, 31] for fair comparisons. All images are resized to 256×256 for training and testing. When extracting hole and non-hole patches in each level, we use nearest neighbor down-sampling to evolve the holes. Our full model runs at 0.19 seconds per frame on a GPU TITAN V for images of size 256×256. All the results reported are output directly from the trained models without using any post-processing. The code will be made publicly available. [1]

4.2. Results

Quantitative comparisons As Places2 contains natural-world images, which is considered as the most challenging dataset [9, 31] (compared with Facade/DTD/CELEBA-HQ), we conduct quantitative comparisons on Places2. All images are randomly masked with 128×128 squares for testing. We use L1 loss, multi-scale structural similarity (MS-SSIM) [26], Inception Score (IS) [20] and Fréchet Inception Distance (FID) [8] as evaluation metrics. The results listed in Table 3 show the comparable performance of the proposed approach against baselines.

[1] https://github.com/researchmm/PEN-Net-for-Inpainting

Dataset	#Train	#Test	#Total
Facade [25]	506	100	606
DTD [4]	4,512	1,128	5,560
CELEBA-HQ [11]	28,000	2,000	30,000
Places2 [32]	1,803,460	36,500	1,839,960

Table 2: Training and test splits of four datasets.

Method	L1 Loss†	MS-SSIM¶	IS¶	FID†
PatchMatch [1]	12.90	60.00%	43.03	20.36
GL [8]	9.27	73.40%	42.05	19.18
PConv [12]	**8.92**	74.67%	47.00	18.39
CA [29]	9.91	73.02%	44.81	18.34
PEN-Net (ours)	9.94	**78.09%**	**50.51**	**15.19**

Table 3: Quantitative comparisons on Places2 with L1 Loss, MS-SSIM, IS and FID. † Lower is better. ¶ Higher is better.

L1 loss can roughly reflect the ability of models to reconstruct the original image content. MS-SSIM extracts and evaluates the similarity of structural information from paired images in multi-scale to provide a good approximation to human visual perception. However, there are a great deal of solutions different from original content for the missing regions, while L1 loss and MS-SSIM are limited to comparing with the original image content. Under the assumption that a damaged scene image should maintain the same attributes after image inpainting, an inpainting result should be confidently identified as a specific category by the pre-trained classification network. To this end, we also use the inception score as one of the evaluation metrics:

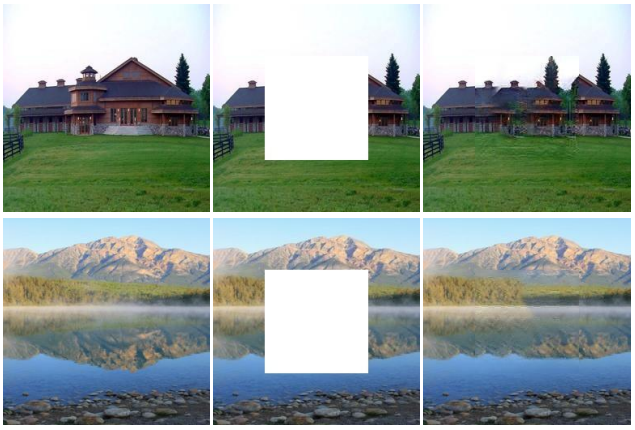
$$I = \exp(\mathop{\mathbb{E}}_{z \sim p_z} [(D_{KL}(p(y|z)) \| p(y))]), \qquad (11)$$

where z is inpainting results defined in Section 3.2, and y is the label predicted by pre-trained classification models. We use the pre-trained classification network released by Zhou *et al*. [32]. Besides, FID has driven an increasing attention and becomes a commonly-used numeric metric in the field of image generation. We also include FID to measure the Wasserstein-2 distance between real and fake images using a pre-trained Inception-V3 model [23].

Qualitative comparisons In order to take both visual and semantic coherence into account, we conduct qualitative comparisons on the test set of four datasets with different characteristics, which are highly-structured with fine-grained textures. We masked the test images with center 128×128 squares, and our model shows superior performance against the state-of-the-art. As shown in Figure 3, the typical patch-based method, PatchMatch, is able to generate clear textures but with distorted structures inconsistent with surrounding areas, while deep generative models including GL, CA and PConv tend to generate blurry textures in the final results. With the help of cross-layer at-

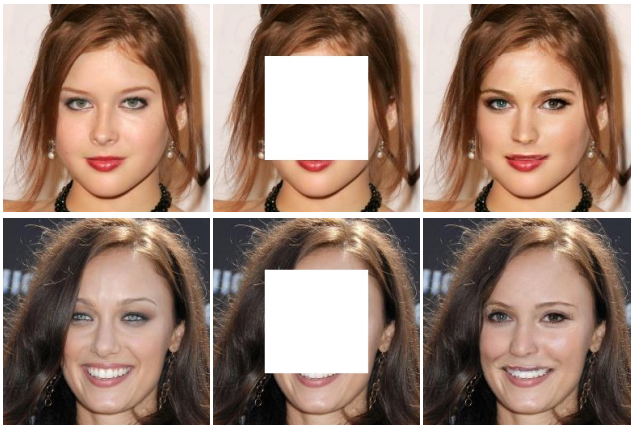
(a) Input (b) PatchMatch (c) GL (d) CA (e) PConv (f) Ours

Figure 4: Qualitative comparisons for image inpainting with irregular masks on Facade. [Best viewed with zoom-in.]



(a) Original (b) Input (c) Ours

Figure 5: Example results generated by the proposed network on Places2. [Best viewed with zoom-in.]



(a) Original (b) Input (c) Ours

Figure 6: Example results generated by the proposed network on CELEBA-HQ. [Best viewed with zoom-in.]

tention transfer and pyramid filling mechanisms, our model is able to generate semantically-reasonable and visually-realistic results with clear textures and consistent structures with context. We also verify the ability of the proposed network to fill missing regions given irregular masks. Specifically, we use the images of Facade and masks released by Liu *et al*. [13] for testing. As shown in Figure 4, the baselines tend to create color discrepancies and distorted structures, while our model outperforms the-state-of-the-art with consistent colors and structures. More example results generated by our model on images of natural scene and human face can be found in Figure 5 and Figure 6.

User study In addition to quantitative and qualitative comparisons, we also perform two settings of user study, *i.e*., paired images and single image user study. The volunteers are all image experts with image processing background. They are not informed of mask information.

In the first setting, over 20 volunteers are invited to evaluate the performance of the models on Facade. Each time, a pair of images generated from different models are shown to the volunteers in an anonymous way. The volunteers are asked to choose the more natural one from those two images. We collected 613 valid votes in total, and the statistics of the results are shown in Table 4. The statistics show that our model is ranked better in most of time (82.10%) over other models. We also found that people prefer clear results generated by PatchMatch (PM), CA and ours.

In the second setting, we randomly distribute the validation set of DTD into four groups. Images in three groups are masked with 32×32, 64×64 or 128×128 squares, and the last group is unmasked. Over 25 volunteers are invited to evaluate the naturalness of inpainting results generated by our model with different mask size. Each time, an image sampled from real data or our inpainting results is shown to

Method	PM	GL	CA	PConv	Ours
Percentage	40.15%	34.25%	70.30%	23.70%	82.10%

Table 4: Statistics of paired images user study. The value indicates the percentage of being ranked as better.

Mask size	0 (real)	32	64	128
Percentage	92.66%	82.23%	52.63%	32.70%

Table 5: Statistics of single image user study. The value indicates the percentage of being considered as real.

Method	L1 loss†	ms-ssim¶	IS¶	FID†
patch-swap [21]	12.13	64.00%	29.26	36.85
single ATN (ours)	**9.85**	71.61%	37.02	26.38
PEN-Net (ours)	9.94	**78.09%**	**50.51**	**15.19**

Table 6: Ablation comparison of cross-layer attention transfer network (ATN) and pyramid filling mechanism over Places2. † Lower is better. ¶ Higher is better.

the volunteers to guess whether the image is a real image from the dataset. We collected 1,425 valid votes in total, and the statistics are shown in Table 5. We found that, the inpainting results from the group with 32×32 masks can be considered as real in 82.23% of the time. Even in the challenging 128×128 case, we received 32.70% votes.

4.3. Analysis

We analyze the effectiveness of different components of the proposed network by visualizing the learned feature maps or ablation study as follows.

Effectiveness of the pyramid L1 loss Pyramid L1 losses is proposed to progressively refine the predictions at each scale. We conduct experiments on images of human faces and visualize the images decoded at each scale. As shown in Figure 7, the pyramid loss is helpful to decode the compact latent feature into an image layer by layer.

Effectiveness of the ATN In order to verify the effectiveness of the attention transfer network (ATN), we visualize the learned feature maps on a same U-Net backbone with different attention mechanisms. As shown in Figure 8, the vanilla U-Net encoder (without using attention) encodes little information inside missing regions, and it fails in generating plausible results. Without the guidance (*i.e.*, attention map) from deeper layers, CA [31] (the commonly-used attention method) failed in filling coherent patches inside the missing regions in shallow layers. With the proposed cross-layer ATN, our model is able to fill regions with coherent patches. In addition to comparing with CA in Figure 8, we further compare with patch-swap layer [21] (the latest attention method) on a same U-Net backbone in Table 6. We can observe that, both cross-layer attention transfer network and pyramid filling mechanism bring improvements of performance on a U-Net backbone.

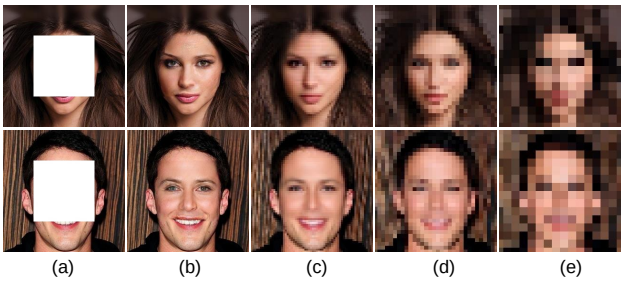


Figure 7: Images generated by the decoder at each scale. (a) is the input. (b) is the final prediction generated by our model. (c), (d) and (e) are prediction output by the decoder at multiple scales (all resized to 256×256 for visualization). [Best viewed with zoom-in.]

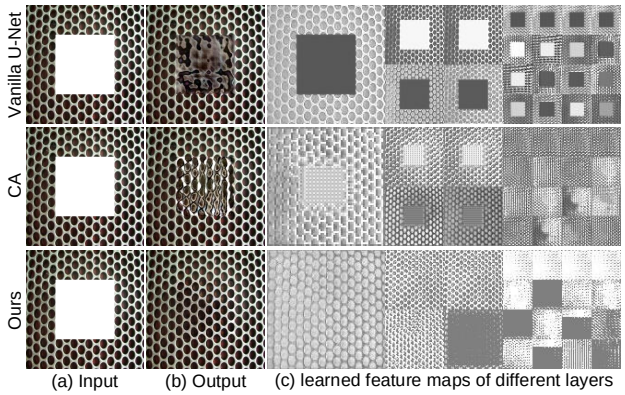


Figure 8: Visualization of the feature maps learned by the encoder. (a) is the input. (b) is the final prediction generated by the models. (c) are visualized feature maps from different layers. [Best viewed with zoom-in.]

5. Conclusion

In this paper, we propose a Pyramid-context Encoder Network (PEN-Net) to generate semantically-reasonable and visually-realistic results for image inpainting. Specifically, the proposed network boosts both the encoding and decoding effectiveness of a vanilla U-Net by using a pyramid-context encoder and a multi-scale decoder. We highlight two key differences of the attention transfer network used in the encoder, cross-layer attention transfer and pyramid filling from high-level semantic features to low-level features with more details. As a future work, we plan to refine the proposed network for higher resolution images.

Acknowledgments This work is partially supported by NSF of China under Grant 61672548, U1611461, 61173081, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *TOG*, 28(3):24:1–24:11, 2009. 2, 5, 6

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *SIGGRAPH*, pages 417–424, 2000. 1

[3] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *TIP*, 12(8):882–889, 2003. 2

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 6

[5] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *TIP*, 13(9):1200–1212, 2004. 1, 2

[6] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346. ACM, 2001. 2

[7] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038. IEEE, 1999. 2

[8] 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 *NeurIPS*, pages 6626–6637, 2017. 6

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *TOG*, 36(4):107, 2017. 2, 3, 5, 6

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 3, 5

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 6

[12] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *ECCV*, pages 377–389, 2004. 1

[13] 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*, pages 85–100, 2018. 2, 3, 5, 6, 7

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 6

[15] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157. IEEE, 1999. 2

[16] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 5

[17] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 1, 2

[18] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *NeurIPS*, pages 901–909, 2015. 3

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 2

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, pages 2234–2242, 2016. 6

[21] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and CC Jay. Contextual-based image inpainting: Infer, match, and translate. In *ECCV*, pages 3–19, 2018. 3, 4, 8

[22] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. In *TOG*, volume 24, pages 861–868, 2005. 2

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 6

[24] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 2

[25] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *GCPR*, pages 364–374, 2013. 6

[26] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *ACSSC*, volume 2, pages 1398–1402. IEEE, 2003. 6

[27] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *TPAMI*, (3):463–476, 2007. 2

[28] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *ECCV*, pages 1–17, 2018. 2, 4

[29] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, pages 6721–6729, 2017. 2, 3

[30] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[31] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018. 2, 3, 4, 5, 6, 8

[32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2018. 6

Supplementary Material

In this section, we present more details of the network architectures and training, additional qualitative comparisons and visual results. We will release the code in the future.

A. Network architectures and training details

Details of the PEN-Net are listed in Table 7 (SUPP.), Table 8 (SUPP.) and Table 9 (SUPP.) respectively. We follow the setting of GntIpt, all images are resized to 256×256 with 128×128 square masks in training. We implement GL [8] and PConv [12] for comparisons, and the performance have achieved the same as reported.

B. More qualitative comparisons and visual results

In addition to Section 4, we show more qualitative comparisons on Facade [23], DTD [4], CELEBA-HQ [10] and Places2 [30] in Figure 9 (SUPP.), Figure 10 (SUPP.) and Figure 11 (SUPP.). More visual results for object removal on images of natural scenes are shown in Figure 12 (SUPP.).

Input: Image $\oplus$ Mask ($256 \times 256 \times 4$)
ϕ^1: Conv. (3, 3, 16), stride=1; LReLU;
ϕ^2: Conv. (3, 3, 32), stride=2; LReLU;
ϕ^3: Conv. (3, 3, 64), stride=2; LReLU;
ϕ^4: Conv. (3, 3, 128), stride=2; LReLU;
ϕ^5: Conv. (3, 3, 256), stride=2; LReLU;
ϕ^6: Conv. (3, 3, 256), stride=2; LReLU;
ϕ^7: Conv. (3, 3, 256), stride=2; ReLU;
ψ^6: ATN(ϕ^6, ϕ^7);
ψ^5: ATN(ϕ^5, ψ^6);
ψ^4: ATN(ϕ^4, ψ^5);
ψ^3: ATN(ϕ^3, ψ^4);
ψ^2: ATN(ϕ^2, ψ^3);
ψ^1: ATN(ϕ^1, ψ^2);

Table 7: The architecture of the pyramid-context encoder. ϕ^i denotes the feature map in the i-th layer defined in Section 3.1, ψ^i denotes the reconstructed features by the ATN in the i-th layer defined in Section 3.1, and LReLU denotes leaky ReLU with the slope of 0.2.

φ^6: DeConv. (3,3,256), stride=2; ReLU; $\oplus$ ψ^6
φ^5: DeConv. (3,3,256), stride=2; ReLU; $\oplus$ ψ^5
φ^4: DeConv. (3,3,128), stride=2; ReLU; $\oplus$ ψ^4
φ^3: DeConv. (3,3,64), stride=2; ReLU; $\oplus$ ψ^3
φ^2: DeConv. (3,3,32), stride=2; ReLU; $\oplus$ ψ^2
φ^1: DeConv. (3,3,16), stride=2; ReLU; $\oplus$ ψ^1
$output\_6$: Conv. (1,1,3), stride=1; clip;
$output\_5$: Conv. (1,1,3), stride=1; clip;
$output\_4$: Conv. (1,1,3), stride=1; clip;
$output\_3$: Conv. (1,1,3), stride=1; clip;
$output\_2$: Conv. (1,1,3), stride=1; clip;
$output\_1$: Conv. (1,1,3), stride=1; clip;

Table 8: The architecture of the multi-scale decoder. $\oplus$ denotes feature concatenation, φ^i denotes the feature maps in the decoder defined in Section 3.2, $output\_i$ denotes the predictions made by the decoder at each scale.

Input: Image ($256 \times 256 \times 3$)
[layer 1]: SNConv. (5,5,64), stride=2; LReLU;
[layer 2]: SNConv. (5,5,128), stride=2; LReLU;
[layer 3]: SNConv. (5,5,256), stride=2; LReLU;
[layer 4]: SNConv. (5,5,512), stride=2; LReLU;
[layer 5]: SNConv. (5,5,1), stride=1; LReLU;

Table 9: The architecture of the discriminator. *SNConv.* denotes the convolutions with spectral normalization.

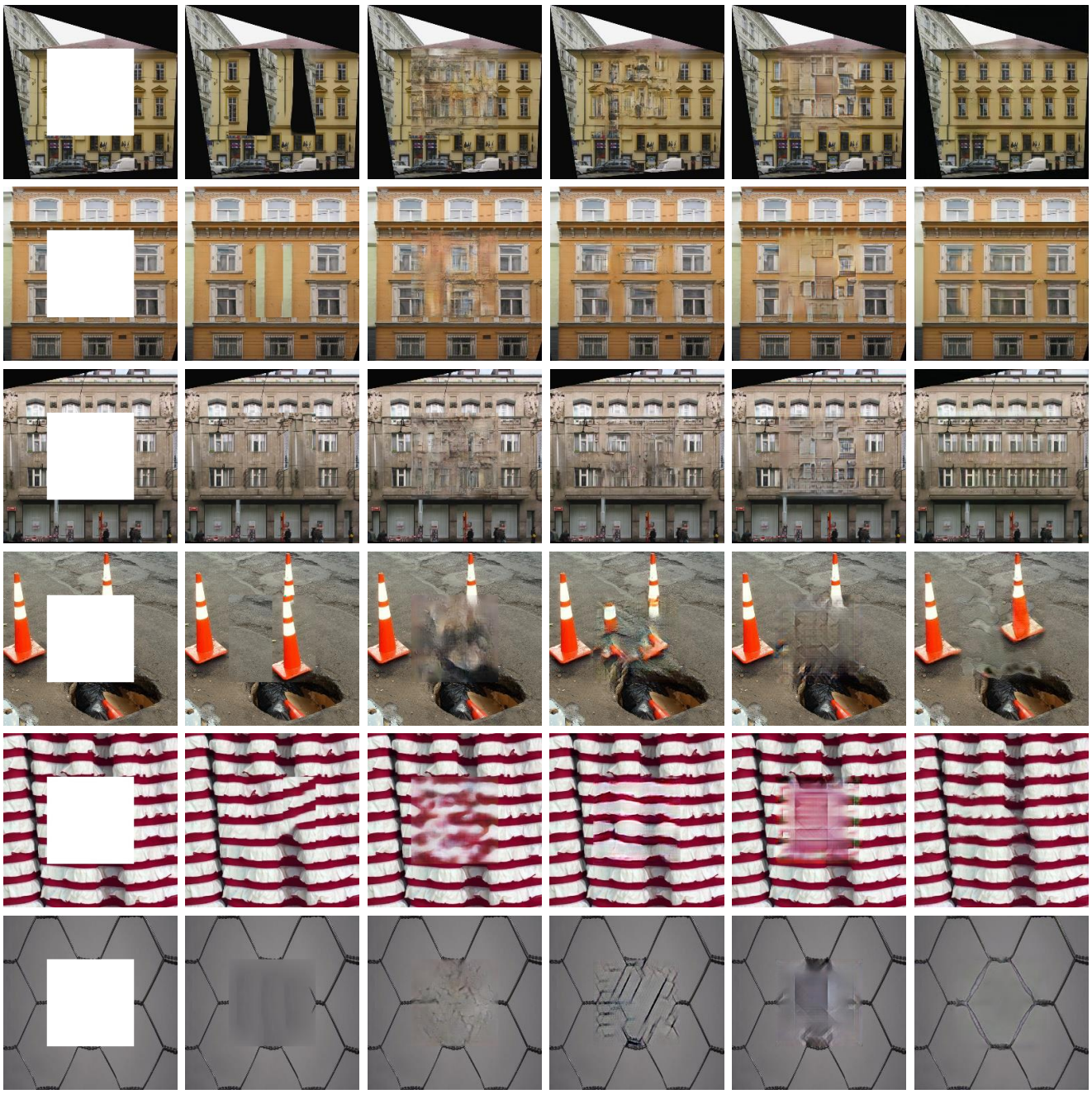
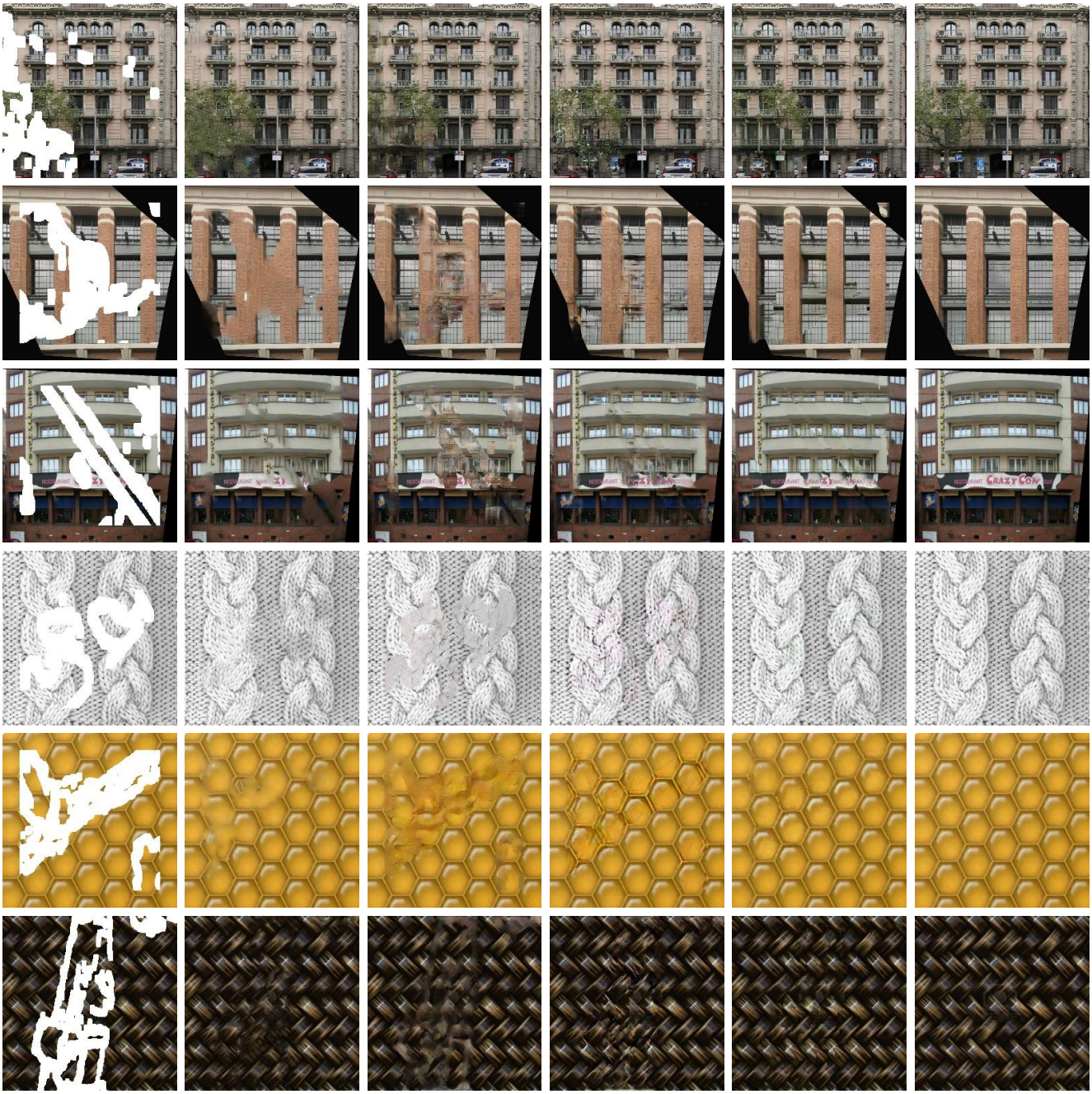


Figure 9: Qualitative comparisons on the test images from Facade and DTD with square masks. In each row, we show from left to right the input, results from PatchMatch, GL, GntIpt, PConv and our model. Compared with baselines, our model is able to generate clear textures and structures that are consistent with context. [Best viewed with zoom-in.]



(a) Input (b) PatchMatch (c) GL (d) CA (e) Ours (f) Ground truth

Figure 10: Qualitative comparisons on the test images from Facade and DTD with irregular masks. In each row, we show from left to right the input, results from PatchMatch, GL, GntIpt, our model, and the ground truth. Compared with baselines, the semantic structures and texture patterns of the results are well-preserved by our model. [Best viewed with zoom-in.]

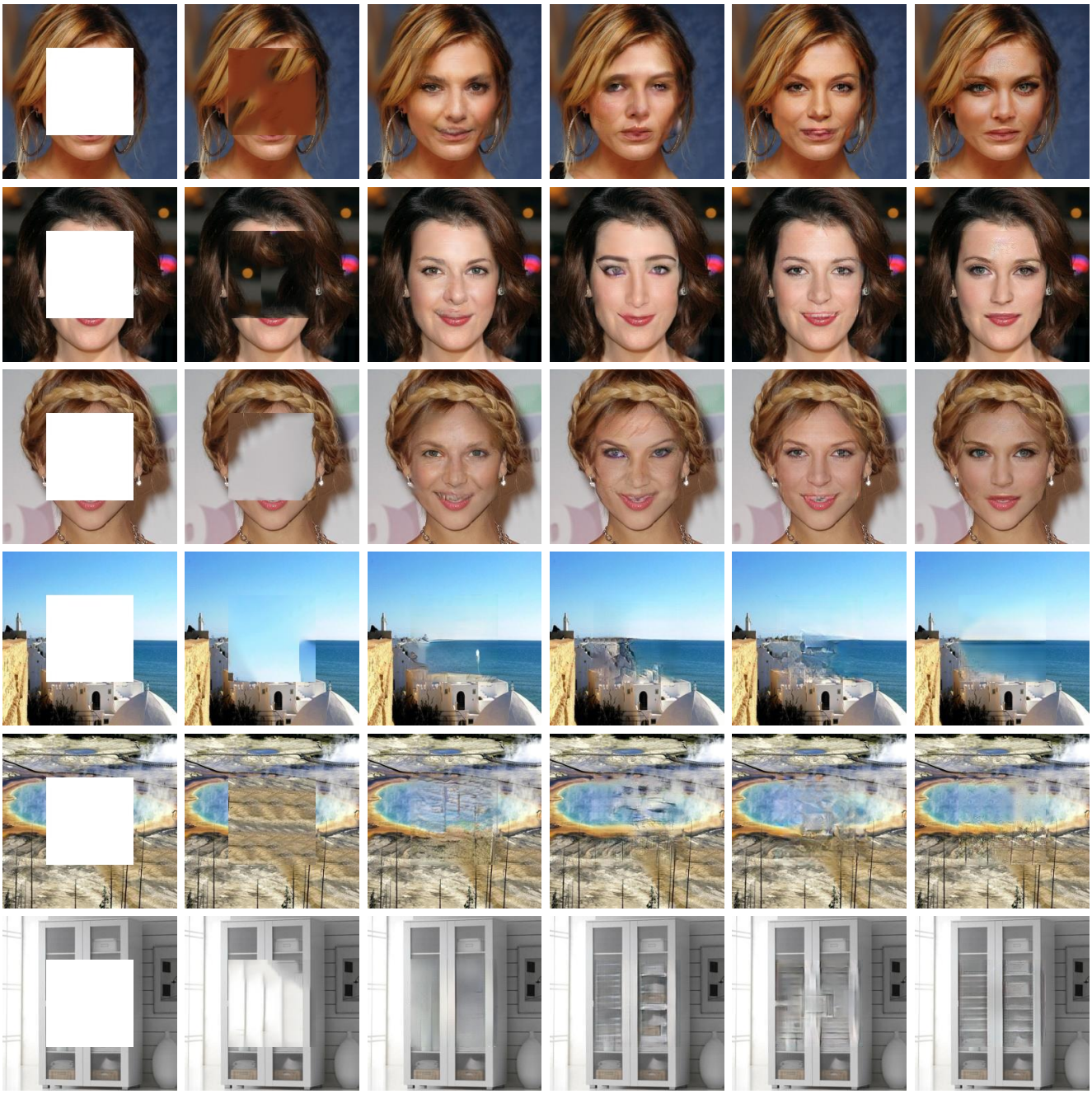


Figure 11: Qualitative comparisons on test images from CELEBA-HQ and Places2 with square masks. In each row, we show from left to right the input, results from PatchMatch, GL, GntIpt, PConv and our model. Compared with baselines, our model is able to generate more natural results for images of human faces and natural scenes. [Best viewed with zoom-in.]

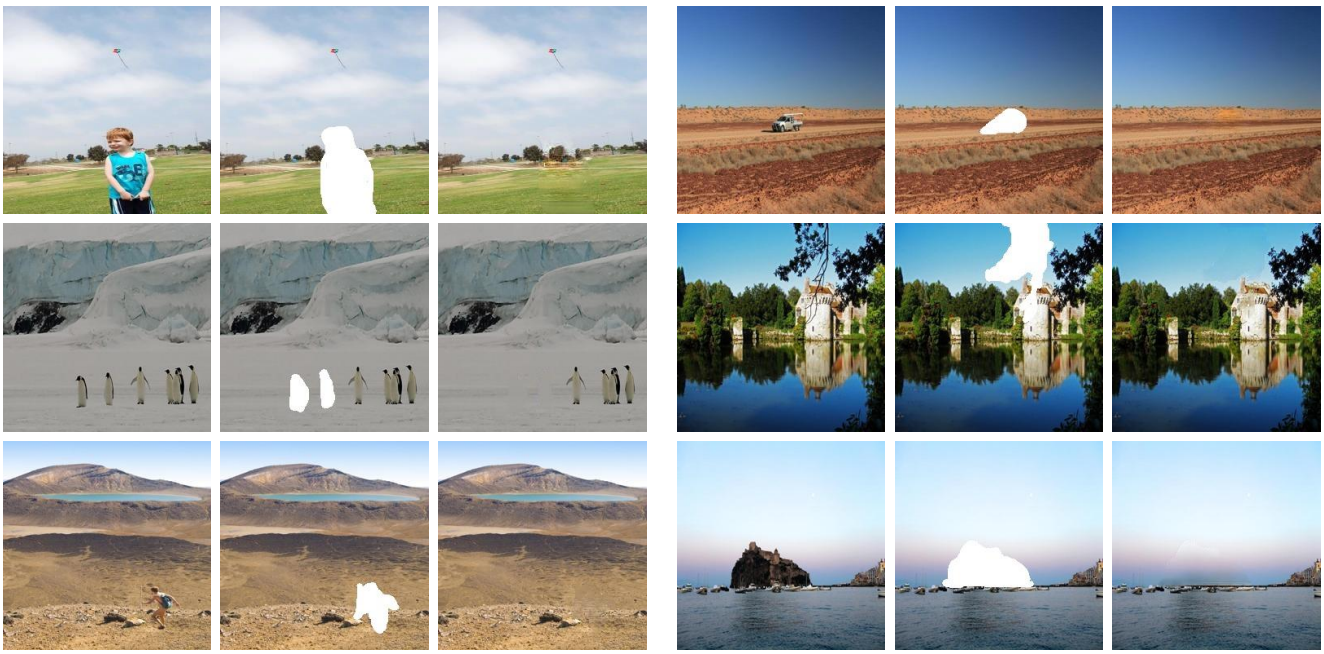


Figure 12: Visual results for object removal on images of natural scenes. Our model is able to generate semantically-reasonable and visually-realistic results, which shows promising applications in user scenarios. [Best viewed with zoom-in.]