Learning to Incorporate Structure Knowledge for Image Inpainting

Conference Paper · February 2020

CITATIONS
0

READS
136

3 authors, including:

Jie Yang
Chinese Academy of Sciences
5 PUBLICATIONS **1** CITATION
SEE PROFILE

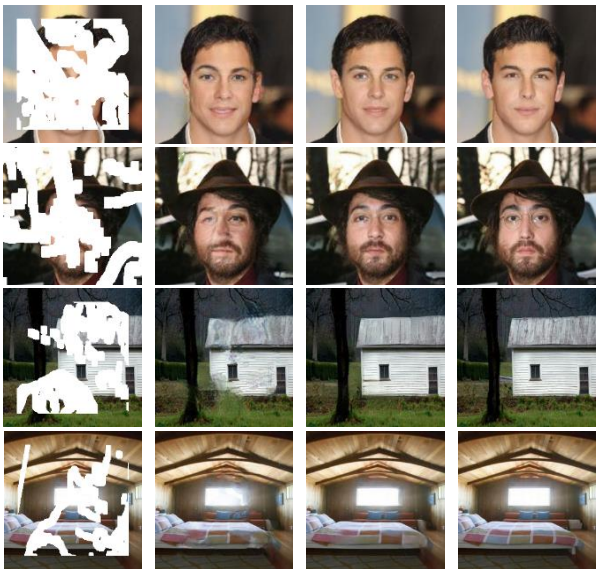
Yong Shi
University of Nebraska at Omaha
458 PUBLICATIONS **6,396** CITATIONS
SEE PROFILE

Learning to Incorporate Structure Knowledge for Image Inpainting

Jie Yang, Zhiquan Qi*, Yong Shi†
University of Chinese Academy of Sciences
Beijing 100190, China
yangjie181@mails.ucas.ac.cn, qizhiquan@foxmail.com, yshi@ucas.ac.cn

Abstract

This paper develops a multi-task learning framework that attempts to incorporate the image structure knowledge to assist image inpainting, which is not well explored in previous works. The primary idea is to train a shared generator to simultaneously complete the corrupted image and corresponding structures — edge and gradient, thus implicitly encouraging the generator to exploit relevant structure knowledge while inpainting. In the meantime, we also introduce a structure embedding scheme to explicitly embed the learned structure features into the inpainting process, thus to provide possible preconditions for image completion. Specifically, a novel pyramid structure loss is proposed to supervise structure learning and embedding. Moreover, an attention mechanism is developed to further exploit the recurrent structures and patterns in the image to refine the generated structures and contents. Through multi-task learning, structure embedding besides with attention, our framework takes advantage of the structure knowledge and outperforms several state-of-the-art methods on benchmark datasets quantitatively and qualitatively.



(a) Input (b) EG (c) Ours (d) GT

Figure 1. Our results compared with EG [16] which exploits structure knowledge with a series-coupled architecture and the ground truth (GT). [Best viewed with zoom-in.]

1. Introduction

Image inpainting aims at filling corrupted or replacing unwanted regions of images with plausible and fine-detailed contents, which is widely applied in fields of restoring damaged photographs, retouching pictures, et al.

Existing inpainting approaches can be roughly divided into two groups: conventional and deep learning based approaches. Conventional inpainting approaches usually make use of low-level features (e.g. color and texture descriptors) hand-crafted from the incomplete input image and resort to priors (e.g. smoothness and image statistics) or auxiliary data (e.g. external image databases). They either propagate low-level features from surroundings to the missing regions following a diffusive process [2, 12, 20] or fill holes by searching and fusing similar patches from the same image or external image databases [6, 1, 19, 7]. Without a high-level understanding of the image contents and structures, conventional approaches usually struggle to generate semantically meaningful content, especially when a large portion of an image is missing or corrupted. Deep learning-based approaches can understand the image content by automatically capturing the intrinsic hierarchical representations and generate high-level semantic features to synthesize the missing contents, which generally outperform the conventional methods in the inpainting task. Context Encoder proposed by Deepak Pathak et al. [17] is the first attempt to exploit a deep convolution encoder-decoder trained with an adversarial strategy for image inpainting. The method produces semantic reasonable contents, but the results often lack fine-detailed textures and contain visible

*Corresponding author
†Corresponding author

artifacts. To achieve more pleasing results, Context Encoder is extended by researchers [10, 32, 34, 24, 31, 27] in different ways, such as in the aspects of architectures and learning strategies.

Recently, Nazeri Kamyar et al. [16] propose to utilize explicit image structure knowledge for inpainting. They develop a two-stage model which comprises of an edge generator followed by an image generator. The edge generator is trained to hallucinate the possible edge sketches of the missing regions. Then the image generator makes the generated sketches as a structure prior or precondition to produce final results. Xiong Wei et al. [30] propose a similar model but take a contour generator instead of an edge generator which is more applicable in the cases where the corrupted image contains salient objects. By introducing the structure information, both methods generate more visually plausible inpainting results.

The success of the above two-stage models suggests that structure knowledge such as edges and contours plays an important role to generate reasonable and detailed contents for image inpainting. It also indicates that, without advisable guidance of structure knowledge in the learning process, previous deep learning-based approaches may struggle to understand the plausible semantic structures of the corrupted images. However, the two-stage strategy may suffer several limitations: 1) it takes much more parameters since using two generators; 2) it is easy subjected to the adverse effects from unreasonable structure preconditions during the inference time due to using a series-coupled architecture; 3) without an explicit structure guidance as a loss function during the learning process, it may not sufficiently incorporate the structure information since they may be weakened or forgotten due to the sparsity of the structures and the depth of the network.

Based on these insights, we propose to use a multi-task framework to better incorporate structure knowledge for image inpainting. Instead of explicit modeling the structure preconditions, we utilize a shared generator to simultaneously generate the completed image and corresponding structures, thus supervising the generator to incorporate relevant structure knowledge for inpainting. This is reasonable because both tasks require a high-level understanding and share the same semantics of the image content. Besides, Nazeri Kamyar et al. [16] and Xiong Wei et al. [30] and have demonstrated that structure priors are benefiting to image completion; the other way round, it is more likely to figure out the complete structures from a relatively intact image compared with a corrupted one.

In addition, to further incorporate the structure information, we introduce a structure embedding scheme which explicitly feeding the learned structure features into the inpainting process serving as preconditions for image completion. Moreover, an attention mechanism is developed to exploit the recurrent structures or patterns in the image to refine the generated structures and contents. Specifically, we also propose a novel pyramid structure loss to supervise the learning of the structure knowledge. We summarize the main contributions as follows:

- We propose a multi-task learning framework to incorporate the image structure knowledge to assist image inpainting.
- We introduce a structure embedding scheme which can explicitly provide structure preconditions for image completion, and an attention mechanism to exploit the similar patterns in the image to refine the generated structures and contents.
- We propose a novel pyramid structure loss specifically for structure learning and embedding. Extensive experiments have been conducted to evaluate the performance of our approach.

2. Related works

Numerous image inpainting approaches have been proposed; here, we focus to review the representative deep learning-based methods.

Context Encoder proposed by Pathak Deepak et al. [17] is one of the first deep learning-based methods for image inpainting, which takes an encoder-decoder architecture and trains with an adversarial learning strategy. It leverages convolutional encoder-decoder and Generative Adversarial Network [5], thus able to develop semantic features and synthesis visually pleasing contents even the missing regions are quite large. But the inpainting results often lack fine-detailed textures due to the information bottleneck layer of the encoder-decoder which may discard some features for image details. Besides, the approach tends to create artifacts around the border of the missing region due to the local consistency is not taken into consideration.

Satoshi Iizuka et al. [10] address the information bottleneck defect by replacing the bottleneck layer with a series of dilated convolution layers and reducing the downsampling times. For local continuity, a local discriminator is designed to enforce the locally filled content is both visually plausible and consistent with the surroundings. Although the method can plausibly fill missing regions, it still takes Poisson blending [18] to tackle the color inconsistency between the completed region and its surroundings. Yang Chao et al. [32], in a different way, enhance Context Encoder by proposing a multi-scale neural patch synthesis approach. The approach first takes the output of the network as initialization and then leverages style transfer techniques [4] to propagate the high-frequency textures from the surroundings to the missing region by iteratively solving a

multi-scale optimization problem. The approach works well for high-resolution semantic inpainting.

Yu Jiahui et al. [34] propose a two-stage coarse-to-fine architecture to generate and refine the inpainting results, where the coarse network makes an initial estimation, and the refinement network takes the initialization to produce finer results. Besides, at the refinement stage, a novel module termed as Contextual Attention is designed to explicitly borrowing information from the surroundings of the missing regions. Song Yuhang et al. [24] develop a similar coarse-to-fine method and introduce a Patch-Swap module which can heuristically propagate the textures from surroundings to the holes. The coarse-to-fine architecture does help to generate finer results; however, it builds upon the assumption that the coarse estimate at the first stage is reasonably accurate. Similar to the ideas of Context Attention and Patch-Swap, Yan Zhaoyi et al. [31] develop a shift-connect module by which the features of the known background at the encoding phase are directly shifted to fill the missing areas at the decoding phase. Unlike using an explicit module to propagate information from the surroundings to missing regions, Wang Yi et al. [27] introduce an implicit diversified Markov random fields (ID-MRF) loss which implicit constraints the network to propagates relevant information to the target inpainting areas. And to leverage features of both image-level and feature-level, Zeng Yanhong et al. [35] propose a pyramid-context encoder network and an attention transfer mechanism which are able to progressively fill the missing regions from high-level to low-level feature map and ensure the semantic consistencies at the same time.

To generalize well in the inpainting tasks of irregular missing regions, Liu Guilin et al. [13] propose partial convolutions. Unlike vanilla convolution, partial convolution only utilizes valid information to inference the missing contents through an automatic mask updating mechanism which is effective in cases of arbitrary missing regions. Yu Jiahui et al. [33] further generalize the partial convolution and propose a gated convolution with a learnable mask updating mechanism which achieve competitive or better inpainting qualities. Besides, the users are able to interact with the inpainting network with hand-drawn sketches to produce user-guided inpainting results.

Recently, several approaches explicitly introduce image structure prior (e.g. edges and contours) for inpainting which produce more impressive results. Nazeri Kamyar et al. [16] propose a model termed as EdgeConnect which consists of an edge generator followed by an image generator. The edge generator is utilized to estimate the possible edges of the missing region, which then as precondition information feed into the successive image completion process. Xiong Wei et al. [30] develop a similar model which takes a contour generator instead of the edge generator. Since the approach predicts contours for salient objects, it is more applicable in the cases where the corrupted image contains salient objects.

3. Method

Our multi-task framework is shown in Figure 2. It estimates a shared generator for simultaneously generating the complete image and corresponding structures at different scales, where the structure generation works as an auxiliary task providing possible structure cues for the image completion task.

Here, we mainly use the edge structures to represent the image structure which describe the profiles of the contents of the image. Instead of directly figuring out the possible edges, we first predict the whole gradient map which inherently contains the edge information and then introduce an implicit regularization scheme in the proposed pyramid structure loss to learn the edge structures. Generating the gradient map is preferable in our multi-task setting. One the one hand, since the edge structure of an image is usually sparse and only conveys binary sketch information of the image, generating such edge structure shares little features with the task of image generation during the last several phases of the generation process, thus task-specific network layers for edge generation have to be designed. One the other hand, the gradient map itself not only conveys the possible edge information but also represents the texture information or high-frequency details which is important for detailed texture synthesis [25, 15].

Formally, let's $\mathbf{I}$ be the ground truth image, $\mathbf{C}$ and $\mathbf{E}$ denote its gradient and edge map respectively. Here, we use Sobel filters shown in Figure 3 to extract the gradient map, and Canny detector to acquire the edge map.

The generator takes the masked image $\hat{\mathbf{I}} = \mathbf{I} \odot (\mathbf{1} - \mathbf{M})$ as the input, and corresponding gradient map $\hat{\mathbf{C}} = \mathbf{C} \odot (\mathbf{1} - \mathbf{M})$ and edge map $\hat{\mathbf{E}} = \mathbf{C} \odot \mathbf{E}$, in addition with the image mask $\mathbf{M}$ (with value 0 for known region 1 otherwise) as preconditions. Here, $\odot$ denotes the Hadamard product. The generator jointly generates the image content and estimates its gradient map at different scales:

$$(\mathbf{I}_{pred}, \mathbf{C}^{(s)}_{pred}) = G(\hat{\mathbf{I}}, \hat{\mathbf{C}}, \hat{\mathbf{E}}, \mathbf{M}) \tag{1}$$

where G represents our generator, $\mathbf{I}_{pred}$ the generated image, $\mathbf{C}^{(s)}_{pred}$ denotes the predicted gradient map at scale s. The final completed image and gradient map are $\mathbf{I}_{comp} = \hat{\mathbf{I}} + \mathbf{I}_{pred} \odot \mathbf{M}$ and $\mathbf{C}^{(s)}_{comp} = \hat{\mathbf{C}}^{(s)} + \mathbf{C}^{(s)}_{pred} \odot \mathbf{M}^{(s)}$, where $\hat{\mathbf{C}}^{(s)}$ is the incomplete gradient map at scale s. The number of scales is upon the specific architecture of the generator.

3.1. Architecture

We take the architecture proposed by Nazeri Kamyar et al. [16] as the backbone of our generator, which has

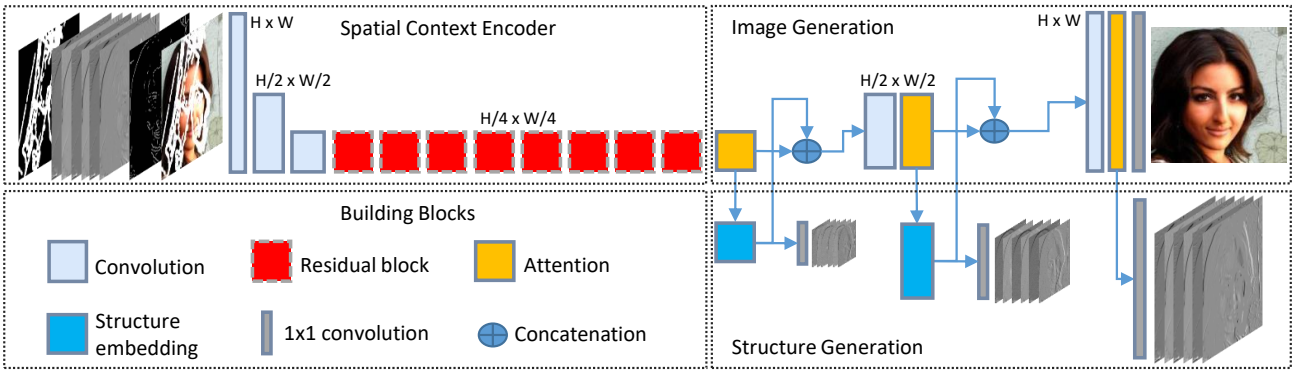


Figure 2. The overview of our multi-task framework. It leverages the structure knowledge with multi-tasking learning (simultaneous image and structure generation), structure embedding and attention mechanism. [Best viewed in color.]

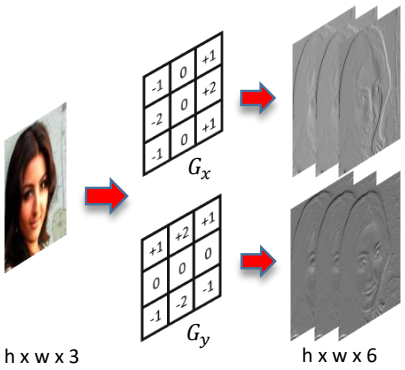


Figure 3. Gradient map ($h \times w \times 6$) extracted from a RGB image with size of ($h \times w \times 3$) by Sobel filters G_x and G_y.

achieved impressive results for image inpainting. As Figure 2 shows, for image generation, the generator consists of a spatial context encoder which down-samples twice followed by eight residual blocks and a decoder which up-samples twice to generate images of the original size. For structure generation, the encoder is shared and the decoder is adapted to a multi-scale style to embed and output the structures of different scales. In addition, two modules are developed to make use of the structure information:

Structure Embedding Layer We use the structure embedding layers to embed the structure features into the decoding phase at different scales serving as priors for image generation. It first separates from the image generation branch to learn the specific structural features and predict the possible structures, then merges the learned features back through a concatenation operation. This parallel/sibling-style scheme not only provides the structure priors for image generation but also avoids the adverse effects from improper preconditions since the decoder can learn to whether to exploit the structure priors or not. Specifically, we implement the layer with a standard residual block [8].

Attention Layer Our attention operation is inspired by the non-local mean mechanism which has been used for deionizing [3] and super-resolution [11]. It calculates the response at a position of the output feature map as a weighted sum of the features in the whole input feature map. And the weight or attention score is measured by the feature similarity. Through attention, similar features from surroundings can be transferred to the missing regions to refine the generated contents and structures (e.g. smoothing the artifacts and enhancing the details).

Given an input feature map, we first extract the feature patches and calculate the cosine similarity $s_{i,j}$ of each pair of the patches:

$$s_{i,j} = \langle \frac{p_i}{||p_i||_2}, \frac{p_j}{||p_j||_2} \rangle \tag{2}$$

where p_i and p_j are the i-th and j-th patch of the input feature map $\mathbf{x}$ respectively. Then softmax operations are applied to compute the attention scores:

$$\hat{s}_{i,j} = \frac{e^{s_{i,j}}}{\sum_{j=1}^{m} e^{s_{i,j}}} \tag{3}$$

Supposing a total of m patches are extracted, the response of a position o_i in the output feature map is calculated as the weighted sum of the patch features:

CA

$$o_i = \sum_{j=1}^{m} \hat{s}_{i,j} p_j \tag{4}$$

In particular, as shown in Figure 4, we formulate all the operations into convolution forms, and make it a residual block which thus can be seamlessly embedded into our architecture:

$$\mathbf{y} = \mathbf{x} + \gamma \mathbf{o} \tag{5}$$

where $\mathbf{y}$ is the residual output, γ is a learnable scale parameter.

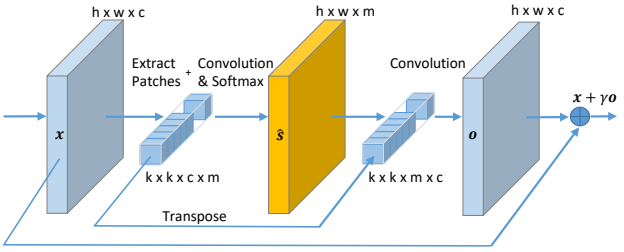


Figure 4. The proposed attention layer. It extracts m feature patches as convolution filters with shape $(k \times k \times c)$ from the input feature map $\mathbf{x}$ $(h \times w \times c)$ and computes the attention score maps $\hat{\mathbf{s}}$ through convolutions between the filters and the input followed by softmax operations, then convolves the scores back to reconstruct the feature map $\mathbf{o}$, finally adds it back to the input feature map with a scale parameter γ.

3.2. Loss Functions

Our generator is expected to achieve two goals — figuring out the structure cues and completing the corrupted image. We introduce a pyramid structure loss to capture the structure knowledge and a hybrid image loss to supervise image inpainting.

Pyramid Structure Loss We propose a pyramid structure loss to guide the structure generation and embedding, thus incorporating the structure information into the generation process. Specifically, it consists of two terms at a specific scale s. One is the L^1 distance between the predicted gradient map and corresponding ground truth, the other is a regularization term for learning the edge structure:

$$\mathcal{L}_{structure} = \sum_{s}^{n_s} [||\mathbf{C}_{pred}^{(s)} - \mathbf{C}^{(s)}||_1 + \beta \mathcal{L}_{edge}^{(s)}] \quad (6)$$

where $\mathcal{L}_{edge}^{(s)}$ denotes the regularization term, β corresponding coefficient and n_s the number of total scales. To implement the regularization on the edge structure, we first use a Gaussian filter g to convolve the binary ground truth edge map $\mathbf{E}^{(s)}$ to create a weighted edge mask as:

$$\mathbf{M}_E^{(s)} = g * \mathbf{E}^{(\mathbf{s})} \quad (7)$$

Then, we computes the edge regularization loss as:

$$\mathcal{L}_{edge}^{(s)} = ||\mathbf{C}_{pred}^{(s)} - \mathbf{C}^{(s)}||_1 \odot \mathbf{M}_E^{(s)} \quad (8)$$

where the weighted edge mask is used to extract the edge information from the gradient map. Using such an edge mask not only considers the positions of the binary edges but also exert constraints on their nearby locations, thus to highlight and intensify the edge structure. In our implementation, a Gaussian filter with size 10×10 and standard deviation 1 is used.

Hybrid Image Loss We take a similar hybrid loss as in [16] for image completion, which consists of a pixel-wise reconstruction loss, a perception loss, a style loss and an adversarial loss which are detailed as follows.

The reconstruction loss is measured by the L^1 distance between the generated image $\mathbf{I}_{pred}$ and corresponding ground truth at pixel level:

$$\mathcal{L}_{rec} = ||\mathbf{I}_{pred} - \mathbf{I}||_1 \quad (9)$$

The perceptual loss computes the L^1 distance between $\mathbf{I}_{pred}$ and its ground truth in the feature spaces after feeding to the pre-trained VGG-19 network [23] on ImageNet dataset [21].

$$\mathcal{L}_{perc} = \sum_{i} ||\phi_i(\mathbf{I}_{pred}) - \phi_i(\mathbf{I})||_1 \quad (10)$$

where ϕ_i is the feature map of the i'th selected layer from VGG-19. Here, layers $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ and $relu5\_1$ are used.

Style loss also compares the L^1 distance between images in feature spaces, but first computing corresponding Gram matrix [4] of each selected feature map:

$$\mathcal{L}_{style} = \sum_{i} ||G_{\phi_i}(\mathbf{I}_{pred}) - G_{\phi_i}(\mathbf{I})||_1 \quad (11)$$

where G_{ϕ_i} is a $C_i \times C_i$ Gram matrix constructed from feature maps ϕ_i of size $H_i \times W_i \times C_i$.

In our framework, an adversarial training strategy is also used which almost has been a standard practice in image generation tasks. We take PatchGAN [37] as our discriminator D and denote its adversarial loss as:

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{I}}[logD(\mathbf{I})] + \mathbb{E}_{\mathbf{I}_{comp}} log[1 - D(\mathbf{I}_{comp})] \quad (12)$$

and the adversarial loss for our generator as:

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{I}_{comp}} log[1 - D(\mathbf{I}_{comp})] \quad (13)$$

Then, the hybrid image loss $\mathcal{L}_{image}$ is defined as:

$$\mathcal{L}_{image} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_{prec} + \lambda_2 \mathcal{L}_{style} + \lambda_3 \mathcal{L}_G \quad (14)$$

where λ_1, λ_2 and λ_3 are hyperparameters which balance the contributions of different loss terms.

Finally, the generator is optimized by minimizing the pyramid structure loss and the hybrid image loss:

$$\mathcal{L} = \mathcal{L}_{image} + \alpha \mathcal{L}_{structure} \quad (15)$$

where α is a predefined weight to balance the two learning tasks. For our experiments, we choose hyperparameters of the hybrid image loss as in [16], and $\alpha = 0.1$, $\beta = 100$.

4. Experiments

In this section, we present our experimental comparisons with several state-of-the-art image inpainting approaches and ablation studies of the effectiveness of our multi-task framework. More results can reference our supplementary material.

4.1. Experimental Settings

Datasets and Baslines We evaluate our approach on three datasets of CelebA [14], Places2 [36] and Facade [26], and compare the results with the following state-of-the-art methods both qualitatively and quantitatively:

- GL: proposed by Satoshi Iizuka et al. [10], which uses two discriminators to ensure global and local consistency of the generated image.
- CA: proposed by Yu Jiahui et al. [34], which leverages a coarse-to-fine architecture with a contextual attention layer to produce and refine the inpainting results.
- PEN-Net: proposed by Zeng, Yanhong et al. [35], which adopts a pyramid context encoder to fill missing regions with features of both image-level and feature-level.
- EG: proposed by Nazeri Kamyar et al. [16], which leverages the edge structure preconditions for inpainting with a series-coupled architecture.

We utilize the available pre-trained models of the baseline approaches and reimplement PEN-Net [35] as there is no publicly available code yet.

Implementation Details For experiments, we resize images to 256×256 and use both regular and irregular image masks for training and testing. For fair comparisons, we use regular masks (with a size of 128×128) following the common experimental settings of baselines and irregular masks as in baseline [16]. We generate gradient maps with Sobel filters and edge maps with Canny detectors as in [16]. To compute the pyramid structure loss, we scale these maps into corresponding resolutions with nearest-neighbor interpolation. We implement our model in TensorFlow using a single NVIDIA GeForce GTX 1080 Ti and the code will be publicly available.

4.2. Qualitative Evaluation

As shown in Figure 5, our approach is able to generate visually realistic images with sharp edges and fine-detailed textures in both regular and irregular mask settings. Besides, testing on regularly masked images as shown in Figure 6 and Figure 7, ours compared with baselines shows obvious visual enhancement on pleasing image structures,

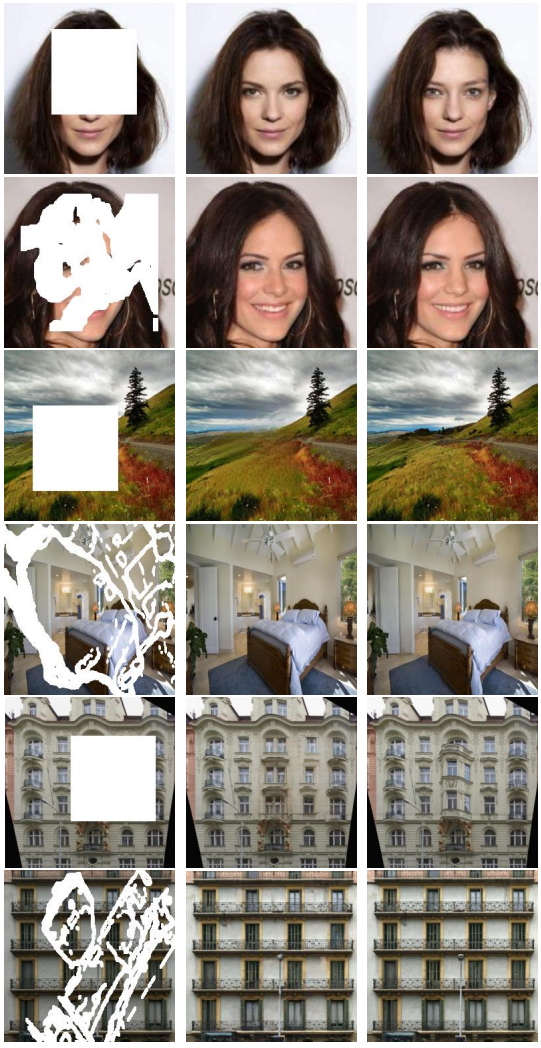
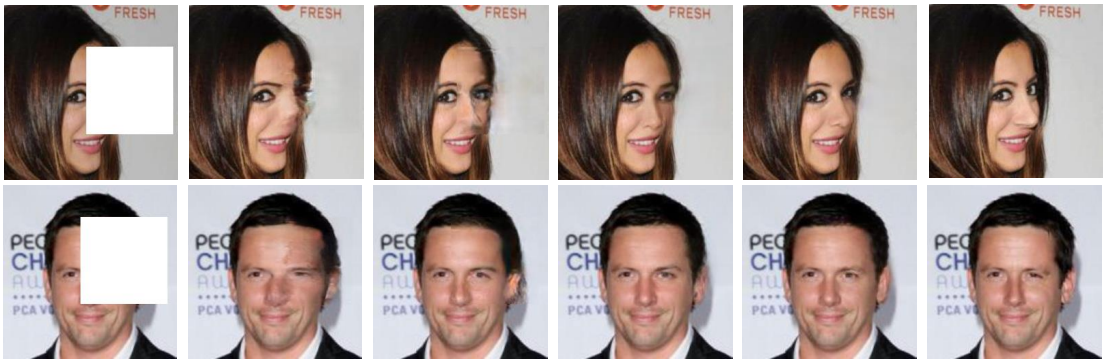


Figure 5. Example inpainting results on CelebA, Places2 and Facade. From left to right: Input, Ours and Ground truth. [Best viewed with zoom-in.]

such as sharp facial contours, crisp eyes and ears, and reasonable object boundaries. And comparing with the approaches CA, GL and PEN-Net where few image structure information is explicitly considered, ours and EG which incorporate the edge structure knowledge are more likely to generate plausible image contents. Moreover, as shown in Figure 1, Figure 6 and Figure 7, comparing against EG which using a serial-coupled architecture to exploiting structure knowledge, our multi-task architecture exhibits superior performance with more visually plausible structures and detailed contents.

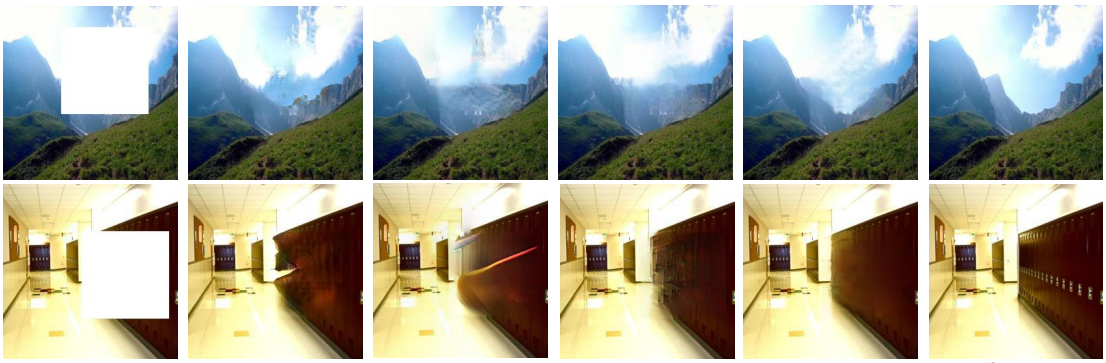
4.3. Quantitative Evaluation

Numerical Metrics We take L_1 loss, peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [29], universal quality index (UQI) [28], visual information fidelity



(a) Input (b) CA (c) PEN-Net (d) EG (e) Ours (f) GT

Figure 6. Qualitative comparisons with baselines and the ground truth (GT) on CelebA. [Best viewed with zoom-in.]



(a) Input (b) CA (c) GL (d) EG (e) Ours (f) GT

Figure 7. Qualitative comparisons with baselines and the ground truth (GT) on Places2. [Best viewed with zoom-in.]

(VIF) [22] and Frechet Inception Distance (FID) [9] as our evaluation metrics. Specifically, we utilize l_1 loss and PSNR to measure the similarity between two images at the pixel level, SSIM and UQI to assess the distortions of the generated image content relative to the ground truth, and VIF and FID to evaluate the overall visual quality, among which VIF correlates well with human perceptions and FID has been a commonly used metric for image generation. In addition, the metrics will be calculated over ten thousand random images in the test sets.

As shown in Table 1, our approach achieves superior performance against all the baselines on datasets CelebA and Places2. The results can be explained by the baseline approaches either ignore the structure knowledge of the image or not well make use of it. Besides, under the scenario with irregular masks, although models such as CA, GL, and PEN-Net can deal with irregular holes (like filling irregular holes with multiple regular patches [30]), they usually show inferior performance since limited to be trained on random square masks.

Table 1. Quantitative comparisons with baselines. ¶ Lower is better. † Higher is better. [Best viewed with zoom-in.]

Datasets	Masks	Models	l_1%¶	PSNR†	SSIM†	UQI†	VIF†	FID¶
Places2	irregular	CA	5.62	21.95	0.732	0.939	0.728	27.81
		GL	6.20	22.40	0.769	0.939	0.689	19.03
		EG	3.33	24.97	0.848	0.967	0.735	13.74
		Ours	**0.69**	**27.07**	**0.887**	**0.975**	**0.787**	**4.883**
	regular	CA	4.44	20.60	0.773	0.951	0.732	7.555
		GL	4.91	21.08	0.777	0.950	0.697	7.848
		EG	3.90	21.63	0.786	0.959	0.709	7.536
		Ours	**3.52**	**22.46**	**0.813**	**0.964**	**0.732**	**7.423**
CelebA	irregular	CA	4.87	23.27	0.790	0.934	0.805	23.13
		PEN-Net	2.94	28.02	0.875	0.972	0.811	10.42
		EG	1.81	30.44	0.941	0.979	0.856	2.443
		Ours	**1.47**	**33.19**	**0.960**	**0.985**	**0.893**	**1.227**
	regular	CA	3.03	23.51	0.864	0.962	0.794	4.033
		PEN-Net	2.54	25.41	0.905	0.971	0.802	3.482
		EG	2.39	25.29	0.901	0.975	0.809	2.421
		Ours	**2.08**	**26.82**	**0.927**	**0.979**	**0.842**	**1.654**

4.4. Ablation Study

We analyze how the proposed components of our framework contribute to the final performance of image inpainting. We take the image generator in [16] as the baseline, then gradually adding our multi-task learning strategy (MT), structure embedding (SE) and attention mechanism(AT) until establishing the whole model we proposed. Correspondingly, we evaluate the model with the gradually added components quantitatively and qualitatively over one thousand random images in the test sets with regular masks.

As Table 2 shows, the performances of our model on the metrics are gradually improved or retained compared with the baseline as progressively integrating each component.

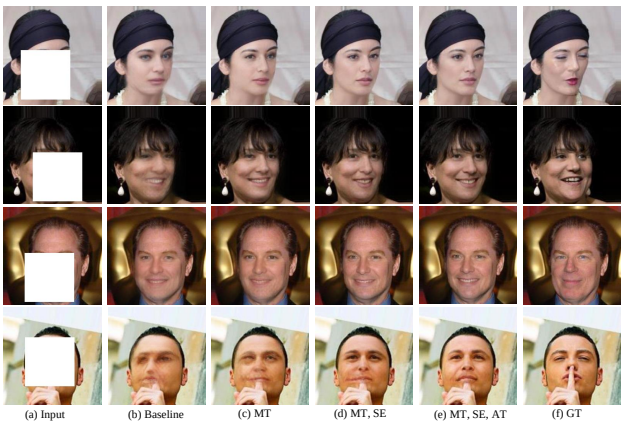


Figure 8. Qualitative results of the ablation study. [Best viewed with zoom-in.]

Specifically, metric VIF and FID are enhanced by a large margin, which indicates the visual quality of the complete images are improved substantially. As qualitative comparisons are shown in Figure 8, when taking a shared generator to simultaneously complete the image and corresponding structures instead of the only image completion task as in baseline, ours generates more pleasing image structures (e.g. sharp facial and month contours), which suggests the proposed multi-task strategy shows great potentials for incorporating the structure knowledge into the inpainting process. Besides, with the explicit embedding of the structure features, the inpainting results are further enhanced (e.g. more sharp contours and textures). Moreover, with the attention mechanism embedded, the results are finally polished by the similar structures and patterns in the images.

Table 2. Quantitative results of the ablation study. ¶ Lower is better. † Higher is better. [Best viewed with zoom-in.]

Model configurations	l_1%¶	PSNR†	SSIM†	UQI†	VIF†	FID¶
Baseline	4.03	26.50	0.908	0.977	0.823	15.25
MT	3.83	26.94	**0.912**	0.978	0.834	13.66
MT, SE	3.84	26.91	0.903	0.978	0.844	12.29
MT, SE, AT	**3.78**	**27.01**	0.911	**0.979**	**0.848**	**11.98**

4.5. Object removal in user scenarios

To further evaluate the generalization ability of our inpainting models, we carry out object removal experiments in user scenarios. We develop a interactive image removal and completion tool with Opencv and based on two inpainting models which trained on datasets of CelebA and Places2 respectively. We take images of faces or natural scenes from the internet that our network doesn't see before, then remove the unwanted parts on the images with brushes and complete the missing parts utilizing corresponding inpainting models. Figure 9 shows examples of object removal results. Our model is able to generate semantically reasonable and visually-realistic results, which shows promising applications in real scenarios.

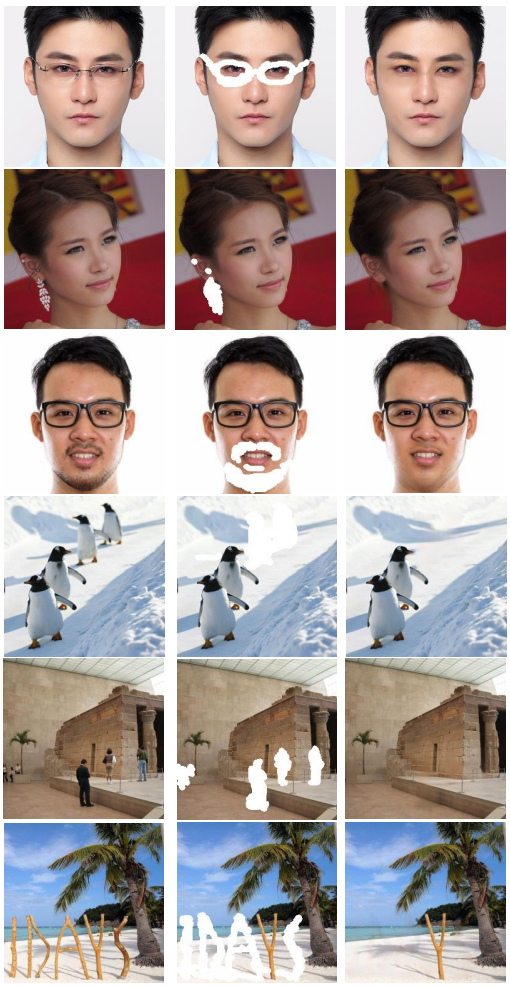


Figure 9. Qualitative results for object removal on images from the internet. From left to right: original image, image with unwanted object removed with masks and inpainted image.

5. Conclusion

We have primarily presented a framework for incorporating image structure knowledge for image inpainting. We propose to utilize the multi-task learning strategy, explicit structure embedding besides with an attention mechanism to make use of the image structure knowledge for inpainting. The experiments results demonstrate that the proposed approach shows superior performance compared with several state-of-the-art inpainting methods which either ignore or not well exploit the structure knowledge. Besides, we have verified each proposed component for incorporating structure knowledge by ablation studies. In future work, we plan to investigate adapting the proposed multi-task framework to other specific inpainting architectures to leverage the structure knowledge.

Acknowledgements

This work is supported by grants from: National Natural Science Foundation of China (No.71932008, 91546201, and 71331005).

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM transactions on graphics*, volume 28, pages 24–32, 2009.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 417–424, 2000.

[3] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition*, volume 2, pages 60–65, 2005.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *The IEEE conference on computer vision and pattern recognition*, pages 272–280, June 2016.

[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*, pages 2672–2680, 2014.

[6] James Hays and Alexei A Efros. Scene completion using millions of photographs. *Communications of the ACM*, 51(10):87–94, 2008.

[7] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European conference on computer vision*, pages 16–29, 2012.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] 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 *Advances in neural information processing systems*, pages 6626–6637, 2017.

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM transactions on graphics*, 36(4):107:1–107:14, 2017.

[11] Daniel Glasner Shai Bagon Michal Irani. Super-resolution from a single image. In *Proceedings of the IEEE international conference on computer vision*, pages 349–356, 2009.

[12] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *Proceedings of the ninth IEEE international conference on computer Vision*, pages 305–312, 2003.

[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 *Proceedings of the european conference on computer vision*, pages 85–100, 2018.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[15] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[16] 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.

[17] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[18] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM transactions on graphics*, 22(3):313–318, 2003.

[19] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. Shift-map image editing. In *2009 IEEE 12th international conference on computer vision*, pages 151–158, 2009.

[20] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *2005 IEEE computer society conference on computer vision and pattern recognition*, volume 2, pages 860–867, 2005.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on image processing*, 15(2):430–444, 2006.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] 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 *Proceedings of the european conference on computer vision*, pages 3–19, 2018.

[25] Yu-Wing Tai, Shuaicheng Liu, Michael S Brown, and Stephen Lin. Super resolution using edge prior and single image detail synthesis. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2400–2407, 2010.

[26] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German conference on pattern recognition*, pages 364–374, 2013.

[27] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in neural information processing systems*, pages 331–340, 2018.

[28] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[30] Wei Xiong, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. June 2019.

[31] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the european conference on computer vision*, pages 1–17, 2018.

[32] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017.

[33] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[34] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[35] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1486–1494, 2019.

[36] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

6. Supplementary Material

In this supplementary material, we present more details of the network architectures and training, additional qualitative comparisons and visual results.

6.1. A. Network Architectures

The detailed architectures of our generator and discriminator are shown in Table 3 and Table 4 respectively. Our model is trained end-to-end using Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.9$. We set the initial learning rate 10^{-4} then lower it to 10^{-5} until metrics converge. The code will be made public in the future[1].

6.2. B. More Experimental Results

For the experiments, we show more qualitative comparison results on Places2 in Figure 10 and CelebA in Figure 11. More visual results are shown in Figure 13 and Figure 14. Additional ablation study results are shown in Figure 12.

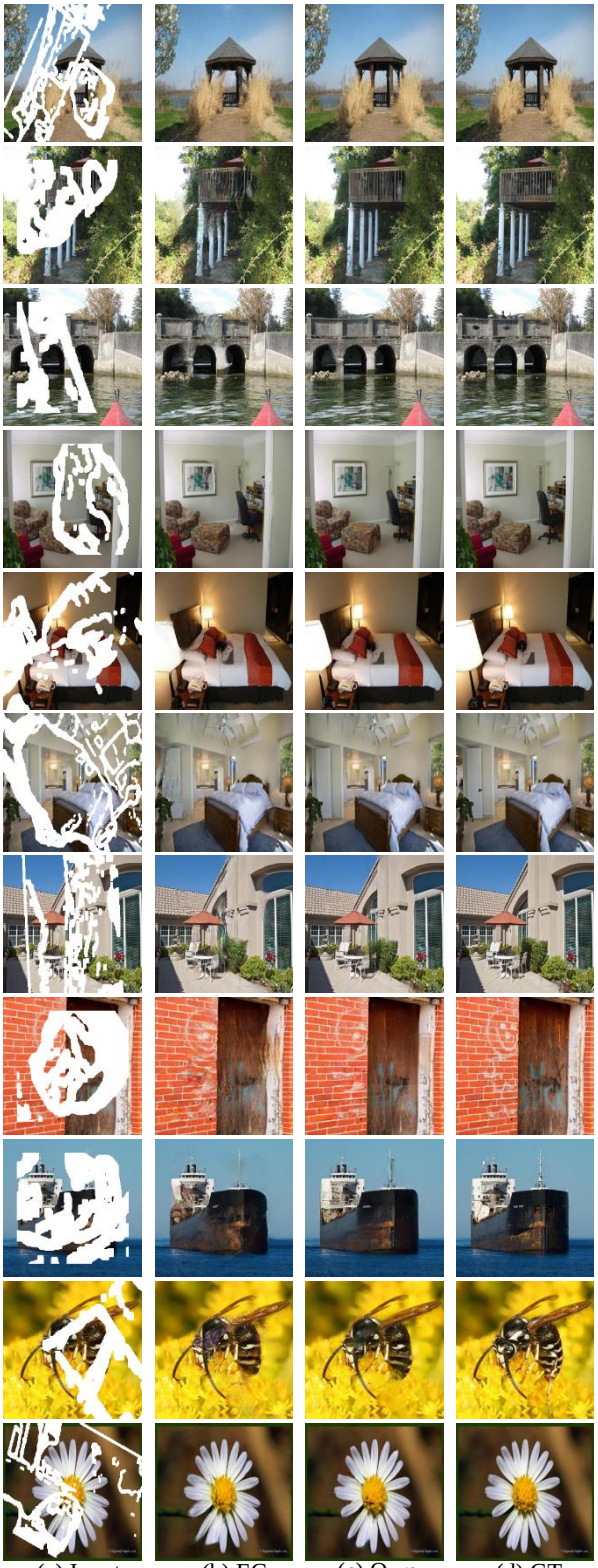
Table 3. The architecture of our generator. $\oplus$ denotes feature concatenation, ϕ^1 the feature maps in the encoder, A^1 attention maps, S^1 structure feature maps, φ^i the features maps in the decoder. [IN: Instance Normalization; RBs: Residual Blocks (Nazeri et al. 2019); AT: Attention Layer; SE: Structure Embedding Layer.]

Input: $\hat{\mathbf{I}} \oplus \mathbf{M} \oplus \hat{\mathbf{E}} \oplus \hat{\mathbf{C}}$ $(256 \times 256 \times 11)$
ϕ^1: Conv. (7, 7, 64), stride=1; IN; ReLU;
ϕ^2: Conv. (4, 4, 128), stride=2; IN; ReLU;
ϕ^3: Conv. (4, 4, 256), stride=2; IN; ReLU;
ϕ^4: Eight RBs(ϕ^3)
A^1: AT(ϕ^4)
S^1: SE(A^1)
Structure Output: Conv. (1, 1, 6), stride=1;
φ^1: $A^1 \oplus S^1$; Deconv. (3, 3, 128), stride=2; IN; ReLU;
A^2: AT(φ^1)
S^2: SE(A^2)
Structure Output: Conv. (1, 1, 6), stride=1;
φ^2: $A^2 \oplus S^2$; Deconv. (3, 3, 64), stride=2; IN; ReLU;
φ^3: AT(φ^2); Conv. (5, 5, 64), stride=1; IN; ReLU;
Structure Output: Conv. (1, 1, 6), stride=1; **Image Output**: Conv. (1, 1, 3), stride=1;

Table 4. The architecture of our discriminator. SNConv. denotes the convolutions with spectral normalization.

Input: $\mathbf{I}_{comp}$ $(256 \times 256 \times 3)$
[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;

[1]https://github.com/YoungGod/sturcture-inpainting

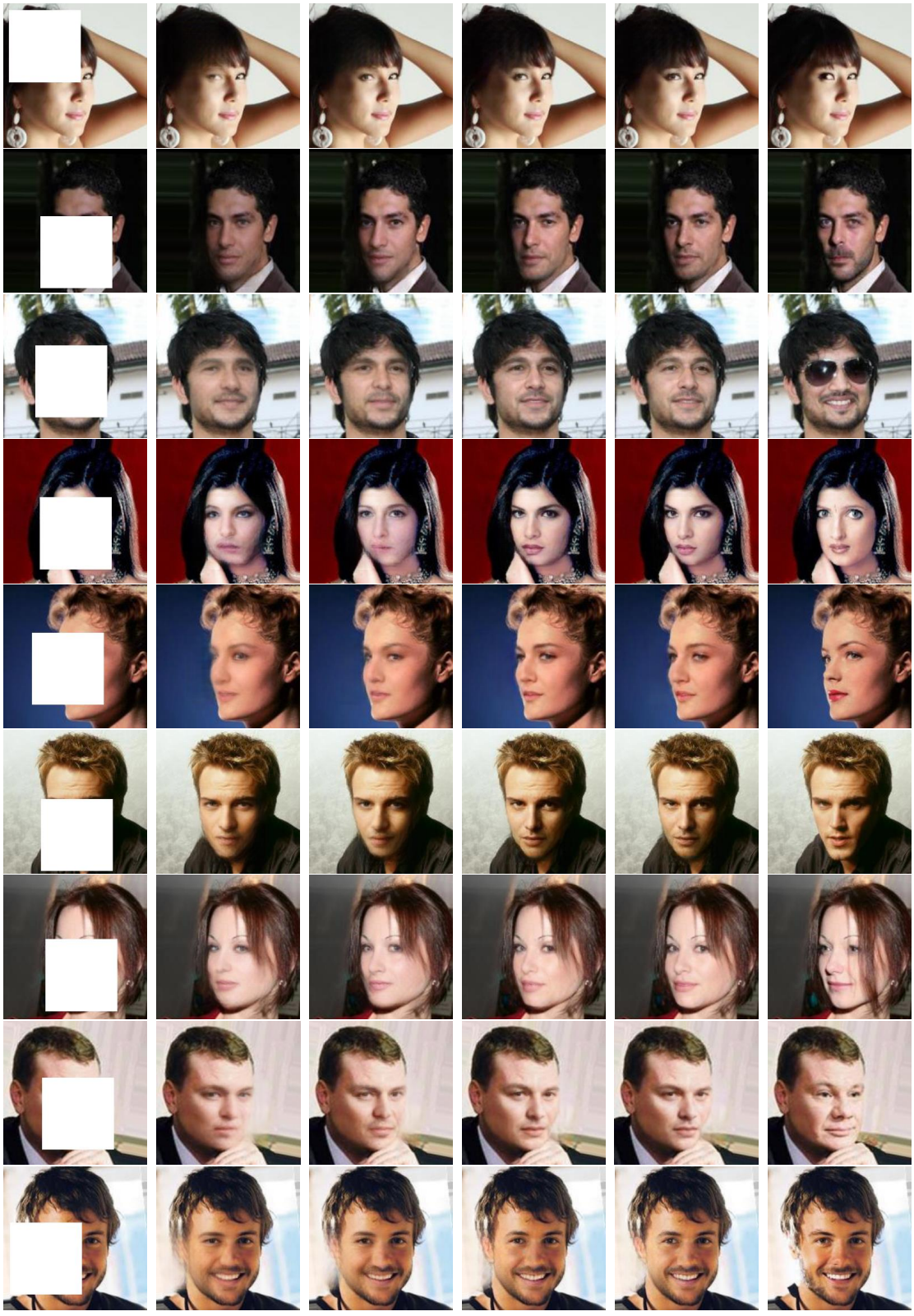


(a) Input (b) EG (c) Ours (d) GT

Figure 10. Ours compared with EG (Nazeri et al. 2019) and the ground truth (GT) on Places2. [Best viewed with zoom-in.]

(a) Input (b) CA (c) PEN-Net (d) EG (e) Ours (f) GT

Figure 11. Ours compared with baselines and the ground truth (GT) on CelebA. [Best viewed with zoom-in.]

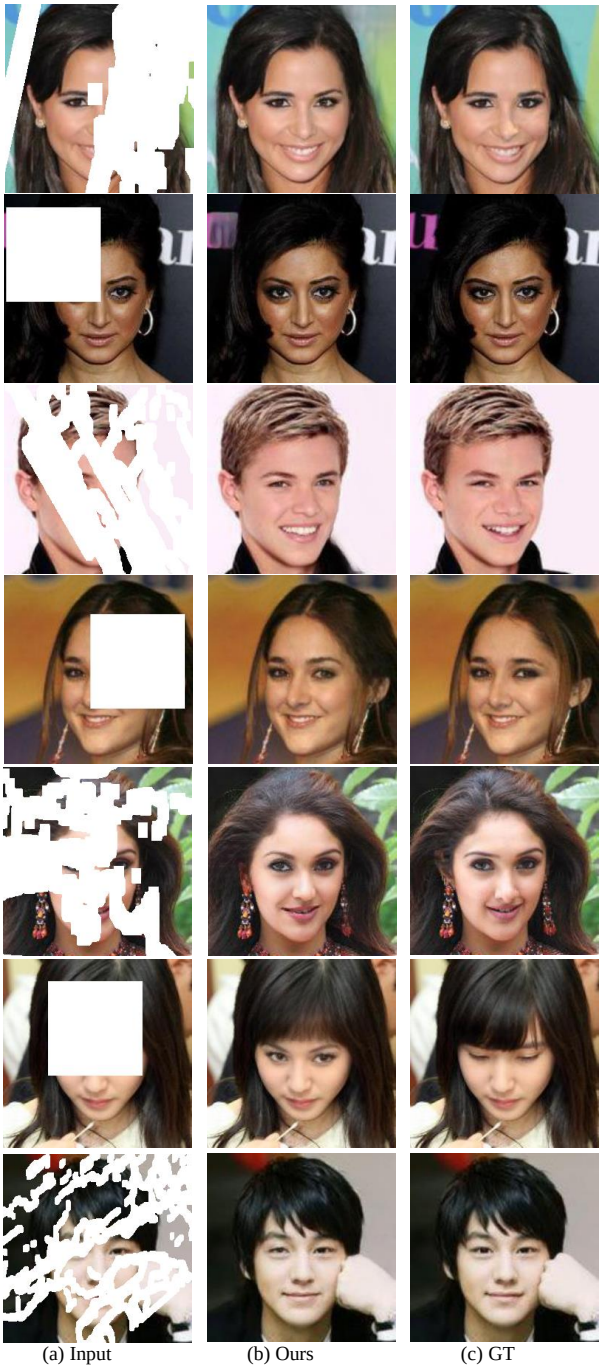


(a) Input (b) Baseline (c) MT (d) MT, SE (e) MT, SE, AT (f) GT

Figure 12. Qualitative results of the ablation study. [Best viewed with zoom-in.]

(a) Input (b) Ours (c) GT

Figure 13. Example inpainting results on Facade. [Best viewed with zoom-in.]



(a) Input (b) Ours (c) GT

Figure 14. Example inpainting results on CelebA. [Best viewed with zoom-in.]