Coherent Semantic Attention for Image Inpainting

Hongyu Liu Bin Jiang* Yi Xiao Chao Yang
College of Computer Science and Electronic Engineering
Hunan University
{kumapower, jiangbin, yangchaoedu, yixiao_csee}@hnu.edu.cn

Abstract

The latest deep learning-based approaches have shown promising results for the challenging task of inpainting missing regions of an image. However, the existing methods often generate contents with blurry textures and distorted structures due to the discontinuity of the local pixels. From a semantic-level perspective, the local pixel discontinuity is mainly because these methods ignore the semantic relevance and feature continuity of hole regions. To handle this problem, we investigate the human behavior in repairing pictures and propose a fined deep generative model-based approach with a novel coherent semantic attention (CSA) layer, which can not only preserve contextual structure but also make more effective predictions of missing parts by modeling the semantic relevance between the holes features. The task is divided into rough, refinement as two steps and we model each step with a neural network under the U-Net architecture, where the CSA layer is embedded into the encoder of refinement step. Meanwhile, we further propose consistency loss and feature patch discriminator to stabilize the network training process and improve the details. The experiments on CelebA, Places2, and Paris StreetView datasets have validated the effectiveness of our proposed methods in image inpainting tasks and can obtain images with a higher quality as compared with the existing state-of-the-art approaches [1].

1. Introduction

Image inpainting is the task to synthesize the missing or damaged parts of a plausible hypothesis, and can be utilized in many applications such as removing unwanted objects, completing occluded regions, restoring damaged or corrupted parts. The core challenge of image inpainting is to maintain global semantic structure and generate realistic texture details for the missing regions.

Traditional works [11, 1, 4, 5, 33] mostly develop texture synthesis techniques to address the problem of hole filling. In [4], Barnes et al. propose the Patch-Match algorithm which iteratively searches for the best fitting patches from hole boundaries to synthesize the contents of the missing parts. Wilczkowiak et al. [33] take further steps and detect desirable search regions to find better match patches. However, these methods fall short of understanding high-level semantics and struggle at reconstructing these locally unique patterns. In contrast, early deep convolution neural networks based approaches [16, 22, 27, 26] learn data distribution to capture the semantic information of the image. Although these methods can achieve plausible inpainting results, they fail to effectively utilize contextual information to generate the contents of holes, thus leading to the results containing noise patterns.

Some recent studies effectively utilize the contextual information and obtain better inpainting results. These methods can be divided into two types. The first type [40, 35, 30] utilizes spatial attention which takes surrounding image features as references to restore missing regions. These methods can ensure the semantic consistency of generated content with contextual information. However, they focus only on rectangular shaped holes, and the results show pixel discontinuous and have semantic chasm (See in Fig 1(b, c)). The second type [23, 39] is to make the prediction of the missing pixels condition on the valid pixels in the original image. These methods can handle irregular holes properly, but the generated contents still meet problems of semantic fault and boundary artifacts (See in Fig 1(g, h)). The reason that the above mentioned methods do not work well is because they ignore the semantic relevance and feature continuity of generated contents, which is crucial for the local pixel continuity in image level.

In order to achieve better image inpainting results on both centering and irregular cases, we investigate the human behavior in inpainting pictures and find that such process involves two steps as conception and painting to guarantee both global structure consistency and local pixel continuity of a picture. To put it more concrete, a man first observes

*Corresponding author

[1]The codes will be available at https://github.com/KumapowerLIU/CSA-inpainting.

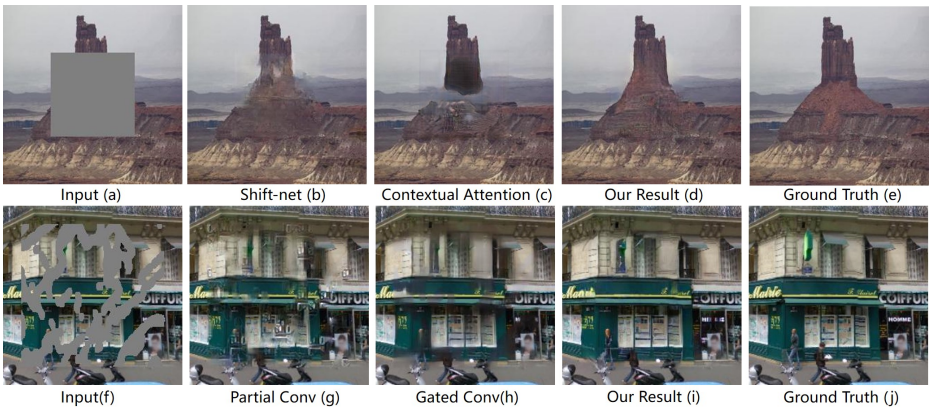

Figure 1. Our results compared with Contextual Attention [40], Shift-net [35], Partial Conv [23], and Gated Conv [39]. First row, from left to right are: image with centering mask, Shift-net [35], Contextual Attention [40], our model, Ground Truth, respectively. Second row, from left to right are: image with irregular mask, Partial Conv [23], Gated Conv [39], our model, Ground Truth, respectively. The size of images are 256×256.

the overall structure of the image and conceives the contents of missing parts during conception process, so that the global structure consistency of the image can be maintained. Then the idea of the contents will be stuffed into the actual image during painting process. In the painting process, one always continues to draw new lines and coloring from the end nodes of the lines drawn previously, which actually ensures the local pixel continuity of the final result.

Inspired by this process, we propose a coherent semantic attention layer (CSA), which fills in the unknown regions of the image feature maps with the similar process. Initially, each unknown feature patch in the unknown region is initialized with the most similar feature patch in the known regions. Thereafter, they are iteratively optimized by considering the spatial consistency with adjacent patches. Consequently, the global semantic consistency is guaranteed by the first step, and the local feature coherency is maintained by the optimizing step.

Concretely, similar to [40], we divide the image inpainting into two steps. The first step is constructed by training a rough network to rough out the missing contents. A refinement network with the CSA layer in encoder guides the second step to refine the rough predictions. In order to make network training process more stable and motivate the CSA layer to work better, we propose a consistency loss to measure not only the distance between the VGG feature layer and the CSA layer but also the distance between the VGG feature layer and the the corresponding layer of the CSA in decoder. Meanwhile, in addition to a patch discriminator [17], we improve the details by introducing a feature patch discriminator which is simpler in formulation, faster and more stable for training than conventional one [25]. Except for the consistency loss, reconstruction loss and relativistic average LS adversarial loss [20] are incorporated as constraints to instruct our model to learn meaningful parameters.

We conduct experiments on standard datasets CelebA [24], Places2 [43], and Paris StreetView [8]. Both the qualitative and quantitative tests demonstrate that our method can generate higher-quality inpainting results than existing ones. (See in Fig 1(d, i)).

Our contributions are summarized as follows:

- We propose a novel coherent semantic attention layer to construct the correlation between the deep features of hole regions. No matter whether the unknown region is irregular or centering, our algorithm can achieve state-of-the-art inpainting results.
- To enhance the performance of the CSA layer and ensure the training stability, we introduce the consistency loss to guide the CSA layer and the corresponding decoder layer to learn the VGG features of ground truth. Meanwhile, a feature patch discriminator is designed and jointed to achieve better predictions.
- Our approach achieves higher-quality results in comparison with [40, 35, 23, 39] and generates more coherent textures. Besides, even the inpainting task is completed in two stages, our full network can be trained in an end to end manner.

2. Related Works

2.1. Image inpainting

In the literature, previous image inpainting researches can generally be divided into two categories: Non-learning inpainting approaches and Learning inpainting approaches. The former is traditional diffusion-based or patch-based methods with low-level features. The latter learns the semantics of image to fulfill the inpainting task and generally

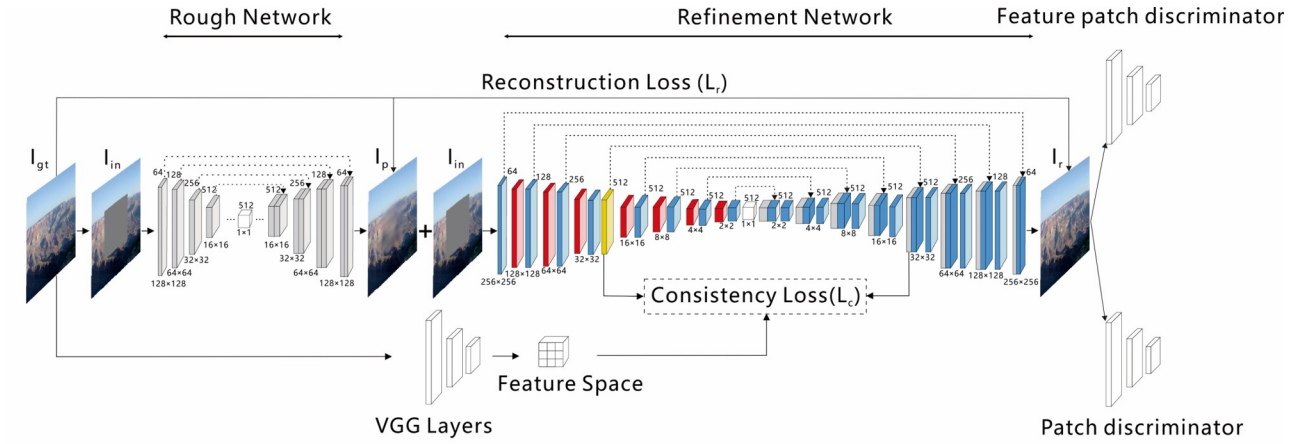

Figure 2. The architecture of our model. We add the CSA layer at the resolution of 32×32 in refinement network.

trains deep convolutional neural networks to infer the content of the missing regions.

Non-learning approaches such as [11, 1, 3, 5, 6, 9, 15, 2, 32, 18, 34, 28, 12, 29] fill in missing regions by propagating neighboring information or copying information from similar patch of the background. Huang et al. [14] blend the known regions into the target regions to minimize discontinuities. However, searching the best matching known regions is a very expensive operation. To address this challenge, Barnes et al. [4] propose a fast nearest neighbor field algorithm which promotes the development of image inpainting applications. Though the non-learning approaches work well for surface textures synthesis, they can not generate semantically meaningful content, and are not suitable to deal with large missing regions.

Learning approaches [38, 22, 31, 37, 41, 7, 42] often use deep learning and GAN strategy to generate pixels of the hole. Context encoders [26] firstly train deep neural networks for image inpainting task, which takes the adversarial training [13] into a novel encoder-decoder pipeline and outputs prediction of missing regions. However, it performs poorly in generating fine-detailed textures. Soon after that, Iizuka et al. [16] extend this work and propose local and global discriminators to improve the inpainting quality. However, it requires the previous processing steps to enforce the color coherency near the hole boundaries. Yang et al. [36] take the result from context encoders [26] as input and gradually increase the texture details to get high-resolution prediction. But this approach significantly increases computational costs due to its optimization process. Liu et al. [23] update the mask in each layer and re-normalize the convolution weights with the mask value, which ensures that the convolution filters concentrate on the valid information from known regions to handle irregular holes. Yu et al. [39] further propose to learn the mask automatically with gated convolutions, and combine with SN-PatchGAN discriminator to achieve better predictions. However, these methods do not explicitly consider the correlation between valid features, thus resulting in color inconsistency on completed image.

2.2. Attention based image inpainting

Recently, the spatial attention based on the relationship between contextual and hole regions is often used for image inpainting tasks. Contextual Attention [40] proposes a contextual attention layer which searches for a collection of background patches with the highest similarity to the coarse prediction. Yan et al. [35] introduce a shift-net powered by a shift operation and a guidance loss. The shift operation speculates the relationship between the contextual regions in the encoder layer and the associated hole region in the decoder layer. Song et al. [30] introduce a patch-swap layer, which replaces each patch inside the missing regions of a feature map with the most similar patch on the contextual regions, and the feature map is extracted by VGG network. Although [40] has the spatial propagation layer to encourage spatial coherency by the fusion of attention scores, it fails to model the correlations between patches inside the hole regions, which is also the drawbacks of the other two methods. To this end, we propose our approach to solve this problem and achieve better results, which is detailed in Section 3.

3. Approach

Our model consists of two steps: rough inpainting and refinement inpainting. This architecture helps to stabilize training and enlarge the receptive fields as mentioned in [40]. The overall framework of our inpainting system is shown in Fig 2. Let I_{gt} be the ground truth images, I_{in} be the input to the rough network. We first get the rough pre-

diction I_p during the rough inpainting process. Then, the refinement network with CSA layer takes the I_p and I_{in} as input pairs to output final result I_r. Finally, the patch and feature patch discriminators work together to obtain higher resolution of I_r.

3.1. Rough inpainting

The input of rough network I_{in} is a 3×256×256 image with center or irregular holes, which is sent to the rough net to output the rough prediction I_p. The structure of our rough network is the same as the generative network in [17], which is composed of 4×4 convolutions with skip connections to concatenate the features from each layer of encoder and the corresponding layer of decoder. The rough network is trained with the L_1 reconstruction loss explicitly.

3.2. Refinement inpainting

3.2.1 refinement network

We use I_p conditioned on I_{in} as input of refinement network that predicts the final result I_r. This type of input stacks information of the known areas to urge the network to capture the valid features faster, which is critical for rebuilding the content of hole regions. The refinement network consists of an encoder and a decoder, where skip connection is also adopted similar to rough network. In the encoder, each of the layers is composed of a 3×3 convolution and a 4×4 dilated convolution. The 3×3 convolutions keep the same spatial size while doubling the number of channels. Layers of this size can improve the ability of obtaining deep semantic information. The 4×4 dilated convolutions reduce the spatial size by half and keep the same channel number. The dilated convolutions can enlarge the receptive fields, which can prevent excessive information loss. The CSA layer is embedded in the fourth layer of the encoder. The structure of decoder is symmetrical to the encoder without CSA layer and all 4×4 convolutions are deconvolutions.

3.2.2 Coherent Semantic Attention

We believe that it is not enough to only consider the relationship between M and $\overline{M}$ in feature map to reconstruct M similar to [40, 35, 30], because the correlation between generated patches is ignored, which may result in lack of ductility and continuity of pixels in the final result.

To overcome this limitation, we consider the correlation between generated patches and propose the CSA layer. We take the centering hole as an example: the CSA layer is implemented in two phases: Searching and Generating. Fig 3 illustrates the operation of the CSA layer, where the M and $\overline{M}$ denote the missing area and the known area in feature maps respectively. For each (1×1) generated patch m_i in M ($i \in (1 \sim n)$, n is the number of patches), the CSA layer searches the closest-matching contextual patch $\overline{m_i}$ in known region M to initialize m_i during the search process. Then we set the $\overline{m_i}$ as a primary part and all the previous generated patch ($m_{1\sim i-1}$) as a secondary part to restore m_i during the generative process. To measure the weight of the two parts, the following cross-correlation metric is adopted:

$$Dmax_i = \frac{< m_i, \overline{m_i} >}{||m_i||.||\overline{m_i}||} \tag{1}$$

$$Dad_i = \frac{< m_i, m_{i-1} >}{||m_i||.||m_{i-1}||} \tag{2}$$

where $Dmax_i$ stands for the similarity between m_i and the most similar patch $\overline{m_i}$ in contextual region, Dad_i represents similarity between two adjacent generated patches. $Dmax_i$ and Dad_i are normalized as the weight for the part of contextual patch and the part of all the previous generated patches respectively. Next, we will describe these two steps in detail.

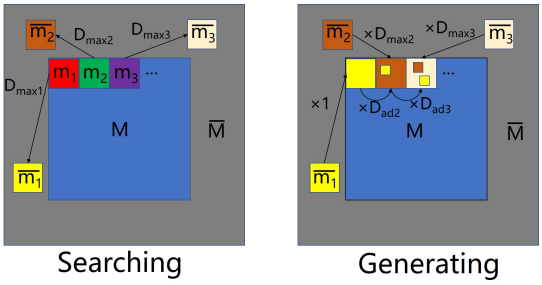

Figure 3. Illustration of the CSA layer. Firstly, we search the most similar contextual patch $\overline{m_i}$ of each generated patch m_i in the hole M, and initialize m_i with $\overline{m_i}$. Then, the previous generated patches and the most similar contextual patch are combined to generate the current one.

Searching: We first extract patches in $\overline{M}$ and reshape them as convolutional filters, then apply the convolution filters on M. With this operation, we can obtain a vector of values denoting the cross-correlation between each patch in M and all patches in $\overline{M}$. On this basis, for each generated patch m_i, we initialize it with the most similar contextual patch $\overline{m_i}$ and assign the maximum cross-correlation value $Dmax_i$ to it for the next operation. **Generating:** The top left patch of M is taken as the initial patch for the generative process (marked by m_1 in Figure 3). Since the m_1 has no previous patch, the Dad_1 is 0 and we replace the m_1 with $\overline{m_1}$ directly, $m_1 = \overline{m_1}$. While the next patch m_2 has a previous patch m_1 as an additional reference, we therefore view the m_1 as a convolution filter to measure the cross-correlation metric Dad_2 between m_1 and m_2. Then, the Dad_2 and $Dmax_2$ are combined and normalized as the weight of m_1 and $\overline{m_2}$ respectively to generate new value of m_2, $m_2 = \frac{Dad_2}{Dad_2+Dmax_2} \times m_1 + \frac{Dmax_2}{Dad_2+Dmax_2} \times \overline{m_2}$. In summary, from m_1 to m_n, the generative process can be

summarized as:

$$
\begin{aligned}
m_1 &= \overline{m_1}, Dad_1 = 0 \\
\underset{i\in(2\sim n)}{m_i} &= \frac{Dad_i}{Dad_i + Dmax_i} \times m_{(i-1)} + \\
&\frac{Dmax_i}{Dad_i + Dmax_i} \times \overline{m_i}
\end{aligned} \tag{3}
$$

As shown in Eq 3, the generating operation is an iterative process, each m_i contains the information of both $\overline{m_i}$ and $m_{1\sim i-1}$, when we calculate Dad_i between m_i and m_{i-1}, the correlations between m_i and $m_{1\sim i-1}$ are all considered. And since the Dad_i value ranges from 0 to 1, the correlation between currently generated patch and the previously generated patches decreases as the distance increases. Based on Eq 3, we get an attention map A_i which records the $\frac{Dmax_i}{Dad_i+Dmax_i}$ and $\frac{Dad_i}{Dad_i+Dmax_i} \times A_{i-1}$ for m_i, then A_1 to A_n form an attention matrix, finally the extracted patches in $\overline{M}$ are reused as deconvolutional filters to reconstruct M. The process of CSA layer is shown in the Algorithm 1.

To interpret the CSA layer, we visualize the attention map of a pixel in Fig 4, where the red square marks the position of the pixel, the background is our inpainted result, dark red denotes the large attention value, and light blue denotes the small attention value.

Algorithm 1 Process of CSA layer

Input: The set of feature map for current batch F_{in}
Output: Reconstructed feature map F_{out}

Searching
Reshape $\overline{M}$ as a convolution filter and apply it on M
Use Eq (1) to compute the $Dmax_i$ and get the $\overline{m_i}$
Initialize m_i with $\overline{m_i}$
End Searching
Generating
for $i = 1 \rightarrow n$ **do**
    Use Eq (2) to calculate the Dad_i
    Use Eq (3) to get the attention map A_i for m_i
end for
Combine A_1 to A_n to get an attention matrix
Reuse $\overline{M}$ as a deconvolutional to get F_{out}
End Generating
Return F_{out}

3.3. Consistency loss

Some methods [27, 23] use the perceptual loss [19] to improve the recognition capacity of the network. However, perceptual loss can not directly optimize a specified convolutional layer, which may mislead the training process of the CSA layer. Moreover, perceptual loss does not ensure consistency between the feature maps after the CSA layer and the corresponding layer in the decoder.

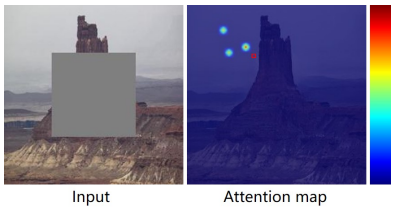

Figure 4. The visualization of attention map. Dark red means the attention value is large, while light blue means the attention value is small.

We then redesign the form of perceptual loss and propose the consistency loss to solve this problem. As shown in Fig 2, we use an ImageNet-pretrained VGG-16 to extract a high level feature space in the original image. Next, for any location in M, we set the feature space as the target for the CSA layer and the corresponding layer of the CSA in decoder respectively to compute the the L_2 distance. In order to match the shape of the feature maps, we adopt 4_3 layer of VGG-16 for our consistency loss. The consistency loss is defined as:

$$
\begin{aligned}
L_c = \sum_{y\in M} &\|CSA(I_{ip})_y - \Phi_n(I_{gt})_y\|_2^2 + \\
&\|CSA_d(I_{ip})_y - \Phi_n(I_{gt})_y\|_2^2
\end{aligned} \tag{4}
$$

Where Φ_n is the activation map of the selected layer in VGG-16. $CSA(.)$ denotes the feature after the CSA layer and $CSA_d(.)$ is the corresponding feature in the decoder.

Guidance loss is similar to our consistency loss, proposed in [35]. They view the ground-truth encoder features of the missing parts as a guide to stabilize training. However, extracting the ground truth features by shift-net is an expensive operation, and the semantic understanding ability of shift-net is not as good as VGG network. Moreover, it cannot optimize the specific convolution layer of the encoder and the decoder simultaneously. In summary, our consistency loss fits our requirements better.

3.4. Feature Patch Discriminator

Previous image inpainting networks always use an additional local discriminator to improve results. However, the local discriminator is not suitable for irregular holes which may be with any shapes and at any locations. Motivated by Gated Conv [39], Markovian Gans [21] and SRFeat [25], we develop a feature patch discriminator to discriminate completed images and original images by inspecting their feature maps. As shown in Fig 5, we use VGG-16 to extract feature map after the pool3 layer, then the feature map is treated as an input for several down-sample layers to capture the feature statistics of Markovain patches [21]. Finally we directly calculate the adversarial loss in this feature map, since receptive fields of each point in this feature map

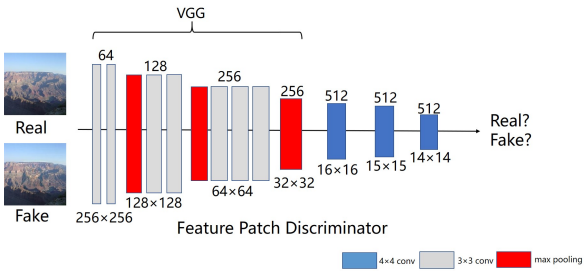

Figure 5. Architecture of our feature patch discriminator network. The number above a convolution layer represents the shape of feature maps.

can still cover the entire input image. Our feature patch discriminator combines the advantages of the conventional feature discriminator [25] and patch discriminator [17], which is not only fast and stable during training but also makes the refinement network synthesize more meaningful high-frequency details.

In addition to the feature patch discriminator, we use a 70×70 patch discriminator to discriminate I_r and I_{gt} images by inspecting their pixel values similar to [25]. Meanwhile, we use Relativistic Average LS adversarial loss [20] for our discriminators. This loss can help refinement network benefit from the gradients from both generated data and real data in adversarial training, which is beneficial for the training stability. The GAN loss term D_R for refinement network and the loss function D_F for the discriminators are defined as:

$$D_R = -\mathbb{E}_{I_{gt}}[D(I_{gt}, I_r)^2] - \mathbb{E}_{I_r}[(1 - D(I_r, I_{gt}))^2] \quad (5)$$

$$D_F = -\mathbb{E}_{I_{gt}}[(1 - D(I_{gt}, I_r))^2] - \mathbb{E}_{I_r}[D(I_r, I_{gt})^2] \quad (6)$$

where D stands for the discriminators, $\mathbb{E}_{I_{gt}/I_f}$ [.] represents the operation of taking average for all real/fake data in the mini-batch.

3.5. Objective

Following the [35], we use L_1 distance as our reconstruction loss to guarantee the constrains that the I_p and I_r should approximate the ground-truth image:

$$L_{re} = \|I_p - I_{gt}\|_1 + \|I_r - I_{gt}\|_1 \quad (7)$$

Taking consistency, adversarial, and reconstruct losses into account, the overall objective of our refinement network and rough network is defined as:

$$L = \lambda_r L_{re} + \lambda_c L_c + \lambda_d D_R \quad (8)$$

where λ_r, λ_c, λ_d are the tradeoff parameters for the reconstruction, consistency, and adversarial losses, respectively.

4. Experiments

We evaluate our method on three datasets: Places2 [24], CelebA [43], and Paris StreetView [8]. We use the original train, test, and validation splits for these three datasets. Data augmentation such as flipping is also adopted during training. Our model is optimized by the Adam algorithm [10] with a learning rate of 2×10^{-4} and β_1 = 0.5. The tradeoff parameters are set as λ_r =1, λ_c=0.01, λ_d=0.002. We train on a single NVIDIA 1080TI GPU (11GB) with a batch size of 1. The training of CelebA model, Paris StreetView model, Place2 model have taken 9 days, 5 days and 2 days, respectively.

We compare our method with four methods:

-CA: Contextual Attention, proposed by Yu et al. [40]

-SH: Shift-net, proposed by Yan et al. [35]

-PC: Partial Conv, proposed by Liu et al. [23]

-GC: Gated Conv, proposed by Yu et al. [39]

To fairly evaluate, we conduct experiments on both settings of centering and irregular holes. We obtain irregular masks from the work of PC. These masks are classified based on different hole-to-image area ratios (e.g., 0-10(%), 10-20(%), etc.). For centering hole, we compare with CA and SH on image from CelebA [24] and Places2 [43] validation set. For irregular holes, we compare with PC and GC using Paris StreetView [8] and CelebA [24] validation images. All the masks and images for training and testing are with the size of 256×256, and our full model runs at 0.82 seconds per frame on GPU for images.

4.1. Qualitative Comparison

For centering mask, as shown in Fig 6, CA [40] is effective in semantic inpainting, but the results present distorted structure and confusing color. SH [35] performances better due to the shift operation and guidance loss, but its predictions are to some extent blurry and detail-missing. For irregular mask, as shown in Fig 7, PC [23] and GC [39] can get smooth and plausible result, but the continuities in color and rows do not hold well and some artifacts can still be observed on generated images. This is mainly due to the fact that these methods do not consider the correlations between the deep features in hole regions. In comparison to these competing methods, our model can handle these problems better, and generate visually pleasing results. Moreover, as shown in Fig 6 and Fig 7 (f, g), A_1 and A_2 are attention maps of two adjacent pixels, the first row is the attention maps of left and right adjacent pixels, the second and third row is the attention maps of up and down adjacent pixels. We see that the attention maps of two adjacent pixels are basically the same, and the correlation areas are not limited to the most relevant contextual areas, the weak correlation areas in attention maps are areas of concern for generated patches which are far from it, the strong correlation areas are areas of concern for both adjacent generated patches and

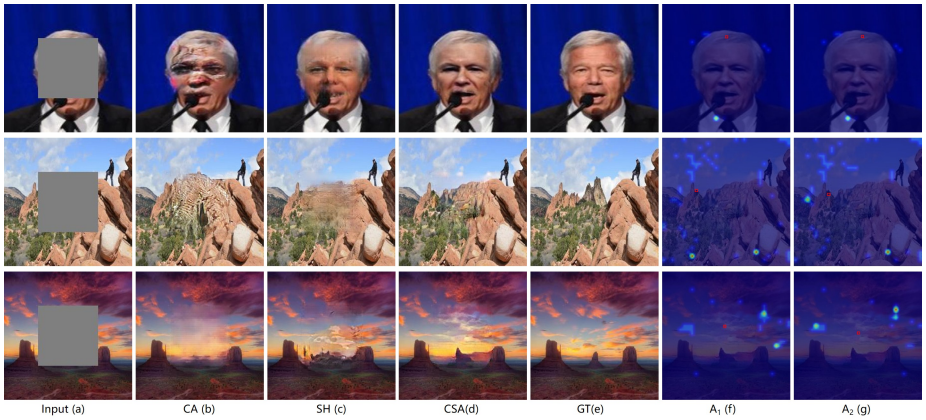

Figure 6. Qualitative comparisons in centering masks cases. The first row is the testing result on Celeba image and the others are the testing result on Places2 images.

Figure 7. Qualitative comparisons in irregular masks cases. The first row is the testing result on Celeba image and the others are the testing result on Paris StreetView images.

the most relevant contextual patch. These phenomena can prove that our approach is better at modeling the coherence of the generated content and enlarging the perception domain for each generated patch than other attention based model [40, 35].

4.2. Quantitative comparisons

We randomly select 500 images from Celeba validation dataset [24] and generate irregular and centering holes for each image to make comparisons. Following the CA [40], we use common evaluation metrics, i.e., L1, L2, PSNR, and SSIM to quantify the performance of the models. Table 1 and Table 2 list the evaluation results with centering mask and irregular masks respectively. It can be seen that our method outperforms all the other methods on these measurements with irregular mask or centering mask.

4.3. Ablation Study

Effect of CSA layer To investigate the effectiveness of CSA, we replace the CSA layer with a conventional 3×3 layer and the contextual attention layer [40] respectively to make a comparison. As shown in Fig 8(b), the mask part fails to restore reasonable content when we use conventional conv. Although contextual attention layer [40] can improve the performance compared to conventional convolution, the inpainting results are still lack of fine texture details and the pixels are not consistent with the background(see Fig 8(c)). Compared with them, our method performs better (see Fig 8(d)). This illustrates the fact that the global semantic structure and local coherency are constructed by the CSA layer.

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
CA	2.64	0.47	0.882	23.93
SH	1.97	0.28	0.926	26.38
CSA	**1.83**	**0.27**	**0.931**	**26.54**

Table 1. Comparison results over Celeba with centering hole between CA [40], SH [35], and Ours. $^-$Lower is better. $^+$Higher is better

Effect of CSA layer at different positions Too deep or

	Mask	PC	GC	CSA
$L_1^-(\%)$	10-20%	1.00	1.00	**0.72**
	20-30%	1.46	1.40	**0.94**
	30-40%	2.97	2.62	**2.18**
	40-50%	4.01	3.26	**2.85**
$L_2^-(\%)$	10-20%	0.12	0.08	**0.04**
	20-30%	0.19	0.12	**0.07**
	30-40%	0.58	0.44	**0.37**
	40-50%	0.76	0.50	**0.44**
$PSNR^+$	10-20%	31.13	31.67	**34.69**
	20-30%	29.10	29.83	**32.58**
	30-40%	23.46	24.48	**25.32**
	40-50%	22.11	23.36	**24.14**
$SSIM^+$	10-20%	0.970	0.977	**0.989**
	20-30%	0.956	0.964	**0.982**
	30-40%	0.897	0.910	**0.926**
	40-50%	0.839	0.860	**0.883**

Table 2. Comparison results over Celeba with irregular mask between PC [23], GC [39], and Ours. $^-$Lower is better. $^+$Higher is better

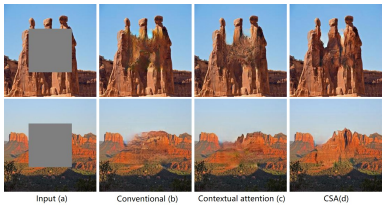

Figure 8. The effect of CSA layer. (b), (c) are results of our model which replace the CSA layer with the conventional layer and the CA layer [40] respectively.

too shallow positions of CSA layer may cause loss of information details or increase calculation time overhead. Fig 9 shows the results of the CSA layer at the 2nd, 3rd, and 4th down-sample positions of refinement network. When the CSA layer is placed on the 2nd position with 64×64 size (See Fig 9(b)), our model performances well but it takes more time to process an image. When the CSA layer is placed on the 4th position with 16×16 size (See Fig 9(c)), our model becomes very efficient but tends to generate the result with coarse details. By performing the CSA layer in the 3rd position with 32×32 size, better tradeoff between efficiency (i.e., 0.82 seconds per image) and performance can be obtained by our model (See Fig 9(d)).

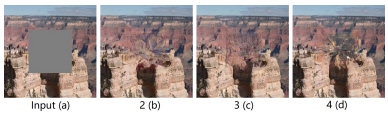

Figure 9. The results of CSA layer on three down-sample positions of refinement network: 2nd, 3rd, and 4th.

Effect of consistency loss We conduct further experiment to evaluate the effect of consistency loss. We add and drop out the consistency loss L_c to train the inpainting model. Fig 10 shows the comparison results. It can be seen that, without the consistency loss, the center of the hole regions present distorted structure, which may be caused by training instability and misunderstanding of image semantic [See Fig 10(b)]. The consistency loss helps to deal with these issues [See Fig 10(c)].

Figure 10. The effect of consistency loss. (b), (c) are results of our model without or with consistency loss

Effect of feature patch discriminator As shown in Fig 11(b), when we only use the patch discriminator, the result performances distorted structure. Then we add the conventional feature discriminator [25], however the generated content still seems blurry (See Fig 11(c)). Finally, by performing the feature patch discriminator, fine details and reasonable structure can be obtained (See Fig 11(d)). Moreover, the feature patch discriminator processes each image for 0.2 seconds faster than the conventional one [25].

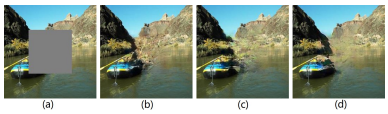

Figure 11. The effect of feature patch discriminator. Given the input (a), (b), (c) and (d) are the results when we use patch discriminator, patch and SRFeat feature discriminators [25], patch and feature patch discriminators, respectively.

5. Conclusion

In this paper, we proposed a fined deep generative model based approach which designed a novel Coherent Semantic Attention layer to learn the relationship between features of missing region in image inpainting task. The consistency loss is introduced to enhance the CSA layer learning ability for ground truth feature distribution and training stability. Moreover, a feature patch discriminator is joined into our model to achieve better predictions. Experiments have verified the effectiveness of our proposed methods. In future, we plan to extend the method to other tasks, such as style transfer and single image super-resolution.

6. Acknowledgements

This work was supported partly by National Natural Science Foundation of China under Grant No. 61702176 and Hunan Provincial Natural Science Foundation of China under Grant No.2017JJ3038.

References

[1] Efros Alexei A and Leung Thomas K. Texture synthesis by nonparametric sampling. *ICECCS*, 2001.

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

[3] 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.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm forstructural image editing. *ACM Transactions on Graphics*, 28, 2009.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. *SIGGRAPH*, 2000.

[6] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on image processing*, 12(8):882–889, 2003.

[7] Cristian Canton Ferrer Brian Dolhansky. Eye in-painting with exemplar generative adversarial networks. *CVPR*, 2018.

[8] Doersch Carl, Singh Saurabh, Gupta Abhinav, Sivic Josef, and Efros Alexei A. What makes paris look like paris? *ACM Transactions on graphics*, 31(4), 2012.

[9] Antonio Criminisi, Patrick Pērez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[10] Kingma Diederik and Ba Jimmy. Adam: A method for stochastic optimization. *ICLR*, 2015.

[11] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. *SIGGRAPH*, 2001.

[12] SELIM ESEDOGLU and JIANHONG SHEN. Digital inpainting based on the mumfordcshah euler image model. *European Journal of Applied Mathematics*, 13(4):353–370, 2002.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *NIPS*, 2014.

[14] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. *ACM Transactions on graphics*, 33(4), 2014.

[15] Drori Iddo, Cohen-Or Daniel, and Yeshurun Hezy. Fragment-based image completion. *ACM Transactions on graphics*, 22:303–312, 2003.

[16] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4), 2017.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[18] Weickert Joachim. Coherence-enhancing diffusion filtering. *International journal of computer vision*, 31(2):111–127, 1999.

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

[20] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv: 1807.00734*, 2018.

[21] Chuan Li and Wand Michael. Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV*, 2016.

[22] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. *CVPR*, 2017.

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

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015.

[25] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. Srfeat: Single image super-resolution with feature discrimination. *ECCV*, 2018.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016.

[27] RaymondYeh, ChenChen, TeckYianLim, Mark Hasegawa-Johnson, and Minh N.Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.

[28] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. *CVPR*, 2008.

[29] Darabi Soheil, Shechtman Eli, Barnes Connelly, Dan B Goldman, and Sen Pradeep. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics*, 31(4), 2012.

[30] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and CC Jay. Contextual-based image inpainting: Infer, match, and translate. *ECCV*, 2018.

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

[32] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-YeungShum. Image completion with structure propagation. *ACM Transactions on Graphics*, 24(3):861–868, 2005.

[33] Marta Wilczkowiak, Gabriel J. Brostow, Ben Tordoff, and Roberto Cipolla. Hole filling through photomontage. *BMVC*, 2005.

[34] Zongben Xu and Jian Sun. Image inpainting by patch propagation using patch sparsity. *IEEE transactions on image processing*, 19(5):1153–1165, 2010.

[35] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. *ECCV*, 2018.

[36] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Lik. High-resolution image inpainting using multi-scale neural patch synthesis. *CVPR*, 2017.

[37] Chao Yang, Yuhang Song, Xiaofeng Liu, Qingming Tang, and C C Jay Kuo. Image inpainting using block-wise procedural training with annealed adversarial counterpart. *arXiv preprint arXiv: 1803.08943*, 2018.

[38] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Hasegawa johnson Mark, and Minh N Do. Semantic image inpainting with deep generative models. *CVPR*, 2017.

[39] 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.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *CVPR*, 2018.

[41] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. Semantic image inpainting with progressive generative networks. *MM*, 2018.

[42] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Guided image inpainting: Replacing an image region by pulling content from another image. *arXiv preprint arXiv: 1803.08435*, 2018.

[43] 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*, 2017.