Semantically Consistent Image Completion with Fine-grained Details

Anonymous CVPR submission

Paper ID 2202

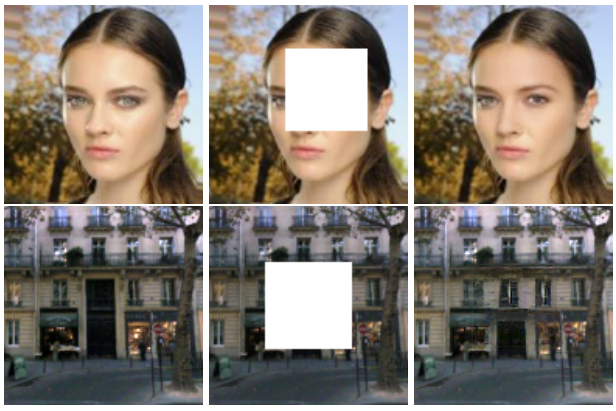
Abstract

Image completion has achieved significant progress due to advances in generative adversarial networks (GANs). Albeit natural-looking, the synthesized contents still lack details, especially for scenes with complex structures or images with large holes. This is because there exists a gap between low-level reconstruction loss and high-level adversarial loss. To address this issue, we introduce a perceptual network to provide mid-level guidance, which measures the semantical similarity between the synthesized and original contents in a similarity-enhanced space. We conduct a detailed analysis on the effects of different losses and different levels of perceptual features in image completion, showing that there exist complementarity between adversarial training and perceptual features. By combining them together, our model can achieve nearly seamless fusion results in an end-to-end manner. Moreover, we design an effective lightweight generator architecture, which can achieve effective image inpainting with far less parameters. Evaluated on CelebA Face and Paris StreetView dataset, our proposed method significantly outperforms existing methods.

1. Introduction

Image completion, also known as image inpainting, aims to restore the damaged images or fill in the missing parts of images with visually plausible contents (Figure 1). As a common image editing technique, it can also be used to remove unwanted objects. Traditional image completion methods [4, 9, 5, 3] utilize low-level information to synthesize missing regions, which fails to produce high-level image semantics. By stacking multiple layers, CNNs are able to capture some intrinsic hierarchical representations. They have been employed in this field and achieve great success [28, 16]. However, most of them are designed for completing small and narrow holes, such as removing text in images.

Recently proposed adversarial training is a powerful strategy to train an image inpainting model [22, 29, 18]. During training, two loss functions are widely used, the pixel-wise reconstruction loss and image-level adversarial loss. The former one helps the model reconstruct the original missing part by ensuring the pixel-wise identity (too low-level), while the latter aims at making the completed image natural-looking enough to fool the discriminator (too high-level). Intuitively, a mid-level guidance, which can encourage the restoration of differently-looking but semantically consistent contents, is lacking.

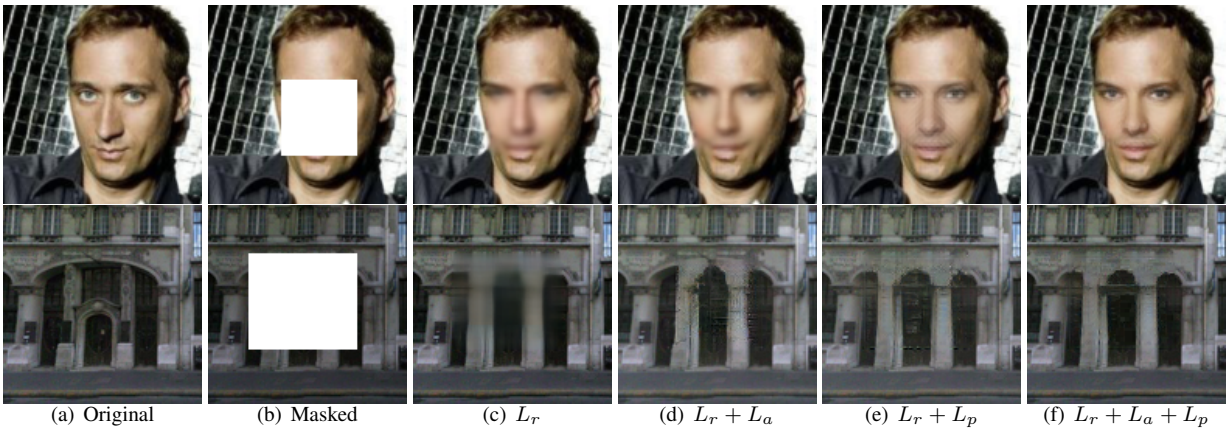


(a) original (b) masked (c) results

Figure 1. Image completion results by our method. Note that the synthesized contents are different from the original images, but the completion results still look natural and semantically consistent.

To close the gap, we propose to utilize a pre-trained CNN (termed as *perceptual network*) in our framework to transform the original image into a suitable feature space, where semantical similarity is enhanced instead of pixel-wise appearance. In this space, differently-looking but semantically consistent contents will get closer. By emphasizing the similarity between the contents and the original contents on the feature level, an additional loss is proposed, termed as *Perceptual Loss*. By minimizing perceptual loss, the model gets more encouragement on generating differently-looking but semantically consistent contents.

In practice, we find that adversarial training and perceptual features are complementary to each other, which is consistent with our analysis that they provide different-level



(a) Original (b) Masked (c) L_r (d) $L_r + L_a$ (e) $L_r + L_p$ (f) $L_r + L_a + L_p$

Figure 2. Our image completion results under different settings. (c) Using reconstruction loss L_r only, completion results are very blurry; (d) Only adding adversarial loss L_a, completion results are a litter sharper and natural, but the synthesized contents still lack details; (e) Only adding perceptual loss L_p, results contain more details, but have clear artifacts and are discontinuous on the boundary (such as color difference); (f) The combination of them produce semantically consistent results with details. (f) shows our final results without any post-precessing procedures.

semantic supervision. Adversarial training helps to make the completed image natural and enables the synthesized contents continuous on the boundary, but the synthesized contents usually lack details (Figure 2(d)). Perceptual features can guide the generator to synthesize fine-grained details. However, they bring artifacts and the synthesized contents are discontinuous with the surrounding regions on the boundary of missing regions (Figure 2(e)). Thus, by combining them together, our proposed method can produce consistent contents with fine-grained details (Figure 2(f)). Since different perceptual features have different effects on contents synthesis, we further make a detailed analysis and comparison of different perceptual features (Figure 4).

Besides, most existing learning-based image completion approaches leverage extremely large models (e.g., number of parameters >100M) to produce reasonable results, which does not fit for real application. Thus, we design a novel generator structure based on fully convolutional network (FCN), which can achieve better performance with far less parameters (number of parameters <10M).

Our contributions can be summarized as follows. We propose to employ supervision of multiple levels for image completion: reconstruction loss (low-level), perceptual loss (mid-level) and adversarial loss (high-level). Based on the detailed analysis and comparison on the effects of different loss functions and different levels of perceptual features, we show that there exists complementarity between adversarial training and perceptual features. Besides, a novel lightweight FCN-based generator structure is designed to produce better inpainting results with far less parameters. Experiments on CelebA Face and Paris StreetView demonstrates the superiority of out proposed method over the existing methods.

2. Related Work

2.1. Adversarial Training

Generative Adversarial Network (GAN) [11] is a minimax two-player game, which utilizes adversarial training to train the generator and discriminator alternatively and has shown powerful ability to generate natural and high-quality images. Denton *et al*. [6] applies a Laplacian pyramid GAN framework to generate high-resolution images in a coarse-to-fine manner. Radford *et al*. [24] proposes deep convolutional GAN (DCGAN), which generates high-quality images in many datasets. Arjovsky *et al*. [2] analyzes the causes of instability theoretically and puts forward Wasserstein GAN.

Although GAN is firstly proposed for image generation, the idea of adversarial training is actually general, which has been widely applied to various research fields, such as image-to-image translation [13] and image suer-resolution [17, 27]. Different from these existing works, we apply adversarial training to generate natural image completion results and ensure the continuity between the synthesized contents and the surrounding regions.

2.2. Perceptual Features

Perceptual features, which are extracted from pre-trained networks (e.g., VGG16 [26]), have been employed in many computer vision tasks. Gatys *et al*. [10] applies perceptual features to neural style transfer, which recombines one image's content and another image's style for high-level image synthesis. Johnson *et al*. [14] trains a feed-forward neural network to solve the optimization problem and improves its speed to achieve real-time style transfer. Ledig *et al*. [17] make use of perceptual features for realistic image super-

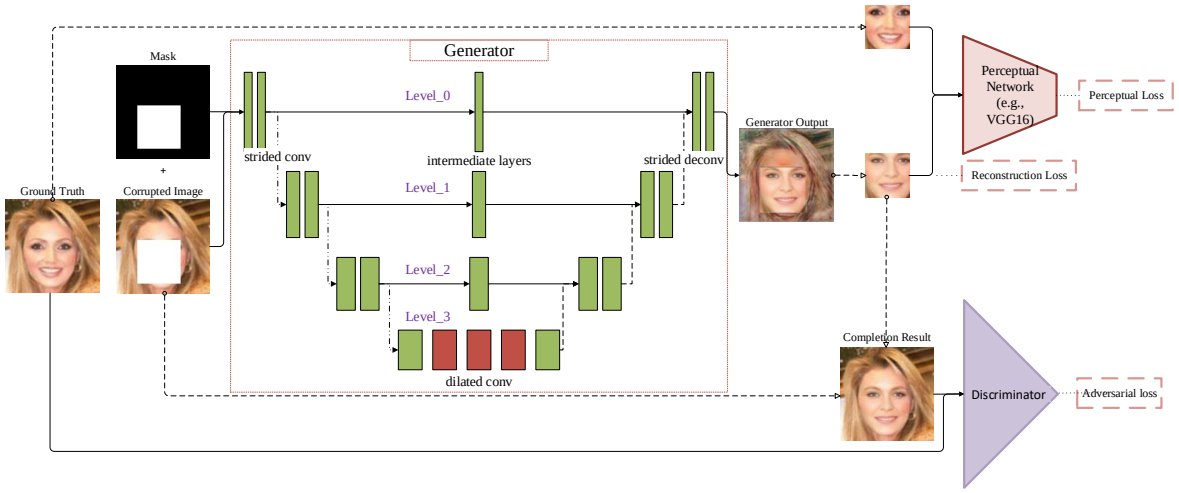


Figure 3. Framework Overview. The generator in our framework can have different number of levels, and it contains three levels in this example. Higher resolution images need more levels to capture larger receptive fields. The discriminator helps make the completion results natural as a whole and enables the boundary continuity. The perceptual network can capture different semantic-level features, which guides the generator to synthesize detailed contents. Only the generator is needed during testing stage.

resolution. Since perceptual features can capture both low-level and high-level semantic information, Dosovitskiy *et al*. [8] utilizes it to generate natural images. We show that perceptual features can also be used for generating details in image completion.

2.3. Image Completion

Traditional image completion algorithms can be broadly divided into two categories. The first category is diffusion-based [4, 9], which propagates low-level features from surrounding regions to the missing holes. However, this category cannot synthesize pleasing contents for textured regions or large holes. The second category is patch-based, which searches similar patches from the same image or database [5, 3]. Methods in this category only make use of low-level features to ensure local similarity and will fail if similar patches do not exist in the database.

Recently, Context Encoder [22] produces reasonable semantic image inpainting results for the first time. It trains an auto-encoder neural network to predict the missing regions with the combination of L_2 loss and adversarial loss. Yeh *et al*. [29] considers image completion as an image generation problem, which leverages a pre-trained DCGAN [24] model and tries to find the closest vector in the latent space. However, it needs many iterations before finding a proper latent vector during testing stage. Moreover, pre-trained GAN models do not always exist. Li *et al*. [18] proposes a generative approach for face completion, which utilizes two adversarial losses and can generate realistic face completion results. However, it cannot work well on misaligned face images and the need for a face scene parsing network restricts its application in other scenes. These large hole-filling approaches only take advantage of adversarial training, thus the inpainting results usually lack high-frequency details and have obvious color difference along mask boundaries, needing some post-processing methods such as Poisson blending [23] to eliminate. Our algorithm can synthesize semantically consistent contents with details and requires no post-processing procedures.

3. Proposed Method

Figure 3 shows the overall architecture of our framework. The generator is used to synthesize the missing contents. The discriminator makes the completed image natural as a whole and enables the boundary continuity between the synthesized contents and the surrounding regions. The perceptual network can capture mid-level semantic features, which guide the generator to synthesize detailed contents.

3.1. Generator

The input of our generator is the corrupted image plus a mask indicating which pixels are missing. Reconstruction loss and perceptual loss are only applied to the masked regions, because we do not care about the unmasked regions of generator output ('Generator Output' in Figure 3). Even so, the generator output is an image of the same resolution as original images, which enables our algorithm to handle the cases of random mask shape and location.

Our generator structure is based on an FCN, which makes use of two good properties of convolution neural networks: translation invariance and parameter sharing. The former is essentially helpful for image completion of arbitrary mask location, and the latter can reduce the number

(a) Masked (b) $L_r + L_{p_1}$ (c) $L_r + L_{p_2}$ (d) $L_r + L_{p_3}$ (e) $L_r + L_{p_4}$ (f) $L_r + L_{p_5}$ (g) $L_r + \{L_{p_i}\}$

Figure 4. The effect of different perceptual features. L_{p_1} is computed based on the output of $conv1\_2$ in VGG16. Similarily, from (c) to (f), we use perceptual features of $conv2\_2$, $conv3\_2$, $conv4\_2$, $conv5\_2$. $\{L_{p_i}\}$ means the combination of several perceptual losses. (b)(c) produce blurry results, meaning that $conv1\_2$ and $conv2\_2$ only capture low-level features. (d)(e)(f) contain more details because they capture some semantic-level representations. (g) is the combination of several semantic-level features and produces the best results.

of parameters tremendously. However, since convolution layers are locally connected, some input pixels cannot influence output pixels if the receptive field is small (e.g., receptive field size $<$ hole size).

Stacking many layers or downsampling can enlarge receptive field. However, the former increases the number of parameters and computation cost, and the latter decreases image restoration performance [21]. To address the above issues, we employ dilated convolution layers [30] to enlarge receptive field without increasing the number of parameters. As shown in Figure 3, we call each downsampling as a $level$, and stack several dilated convolution layers in the last level. Besides, we propose to add intermediate layers at different levels to combine multi-scale information. With the large receptive field and multi-scale information fusion, our generator structure can achieve comparable performance with far less parameters (e.g., set max channel number as 128).

In our implementation, we use 3×3 kernel size in all convolution layers, 4×4 kernel size with stride 2 in deconvolution layers [20]. Apart from the last layer, we add batch normalization (BN) [12] and ReLU activation function after convolution. Different generator structures are compared in the experiment part (Figure 7), which demonstrates the superiority of our generator.

3.2. Discriminator

Reconstruction loss tends to average all of the details, thus the generator can only capture coarse information about the missing regions. Therefore the synthesized contents look blurry (Figure 2(c)). Besides, the generator only optimize the masked regions, thus the synthesized content can be discontinuous with surroundings.

Adversarial training trains discriminator and generator alternatively, which is known to be able to generate natural and realistic images. Therefore we employ a discriminator to distinguish between the original image and the completed image. The unmasked regions are replaced by the original image ('Completion Result' in Figure 3). This is essential and intuitive because the raw generator output ('Generator Output' in Figure 3) is not natural at all and discontinuous on the boundary. The effect of original image context can be seen in Figure 8(c) and 8(d).

As shown in Figure 2(c), 2(d) and Figure 2(e), 2(f), adding adversarial training can indeed make the result more natural and continuous on the boundary between the synthesized contents and surrounding regions.

Our discriminator structure is similar to DCGAN, using BN and Leakly ReLU activation function after each convolution layer.

3.3. Perceptual Network

Discriminator can help produce shaper results, but as shown in Figure 2(d), the synthesized contents still lack details. This is because discriminator only enforces the completion results to look natural as a whole and ignores some essential information for detailed contents synthesis. Perceptual features can capture different semantic-levels features, and by employing semantical similarity between the contents and the original contents on the feature level, the model gets more encouragement on generating semantically consistent contents with fine-grained details.

However, only adding perceptual loss is not a good choice as well, which will lead to artifacts (Figure 2(e)). The reason is that there may exist many unnatural contents sharing the same representation in feature space. That is, similar in feature space does not ensure natural in image space. Therefore, we use both discriminator and perceptual network in our proposed framework.

(a) original (b) masked (c) CAF (d) GFC (e) GFC_post (f) Ours($L_r + L_a$) (g) Ours(final)

Figure 5. Comparison with Content Aware Fill (CAF), Generative Face Completion (GFC) and GFC with post-processing (GFC_post) on CelebA. Our approach can produce more natural results, such as smile in the second row, and can synthesize more details, such as eyes in first row, teeth in second row, glasses in third row and hair in last row.

3.4. Loss Functions

We train the model with a special designed hybrid loss L, which is the combination of reconstruction loss L_r, adversarial loss L_g and perceptual loss L_p. As shown in Figure 3, they correspond to generator, discriminator and perceptual network respectively. The overall loss function is defined as follow:

$$L = L_r + \lambda_1 L_g + \lambda_2 L_p, \tag{1}$$

where λ_1 and λ_2 are hyperparameters that balance the contribution of different losses.

We denote I_{gt}, I_c, I_g as the original, corrupted and generated image respectively. Let M be the binary mask, which has value 1 if the corresponding pixel needs to be completed and 0 otherwise.

Reconstruction Loss. We utilize a smooth loss function L_{smooth} to make an elegant compromise between L_1 and L_2, which is defined as:

$$L_{smooth}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{2}$$

Our reconstruction loss is defined on the masked regions:

$$L_r = L_{smooth}(I_g \odot M - I_{gt} \odot M), \tag{3}$$

where $\odot$ denotes the element-wise multiplication.

Adversarial Loss. Adversarial training trains generator and discriminator alternatively. The discriminator is trained to distinguish whether the corrupted image is completed naturally or not, while the generator tries to fool the discriminator. The image completion result is the combination of the masked regions of I_g and unmasked regions of I_{gt}:

$$I_{completion} = M \odot I_g + (1 - M) \odot I_{gt}, \tag{4}$$

therefore we can obtain the adversarial loss as follows:

$$L_a^D = -(\mathbb{E}[\log D(I_{gt})] + \mathbb{E}[\log(1 - D(I_{completion})]), \tag{5}$$

$$L_a^G = -\mathbb{E}[\log D(I_{completion})]. \tag{6}$$

To stabilize the adversarial training, we adopt one-sided label smoothing strategy [25], which replaces the label 0 and 1 for a classifier with smoothed values. We add smoothed values on $L^D adv$, using a random values between 0 and 0.2 to replace label 0, another random values between 0.8 and 1 to replace label 1.

Perceptual Loss. Perceptual loss is based on perceptual network and different layers can capture different levels of features. Let $\{\phi_l\}$ be a collection of layers in the pre-trained perceptual network. The Perceptual loss function is defined as follows:

$$L_p = \sum_l \alpha_l L_{smooth}(\phi_l(I_{gt} \odot M) - \phi_l(G \odot M)), \tag{7}$$

(a) original (b) masked (c) CAF (d) CE (e) Ours($L_r + L_a$) (f) Ours(final)

Figure 6. Comparison with Content Aware Fill (CAF), Context Encoder (CE) on Paris StreetView. Our method can handle different kinds of masks, but CE only works for specific masks used during training stage. Thus here we follow the original paper of CE and compare inpainting results on the central holes.

where α_l is the weighted hyperparameter of layer l.

4. Experiments

In this section, we first introduce datasets and our experimental settings. Then we conduct four main experiments, which demonstrate the effectiveness of our image completion framework: comparison of different generator structures, effect of original image context, analysis and comparison of different levels of perceptual features, qualitatively and quantitatively evaluation of our image completion results. Finally, we show some real-word applications, such as object removal.

4.1. Datasets

We evaluate our method on CelebA Face dataset [19] and Paris StreetView dataset [7]. CelebA contains 202,599 face images. We randomly choose 200,000 for training, the other 2,599 for testing. Paris StreetView contains 15,000 street scenes in Paris. As in Context Encoder [22], 14,900 images are used for training and 100 for testing.

4.2. Experimental Settings

We implement our model on TensorFlow [1] and use Adam [15] for optimization.

Data Preprocessing. For CelebA, we randomly crop a 160 $\times$ 160 region and resize it to 128 $\times$ 128. For Paris StreetView, we first resize the image so that the smaller dimension is with size 128 and then 128 $\times$ 128 is cropped. We also conduct data augmentation such as random flipping, shift. We randomly choose mask size ranging from 48 to 80 and the mask location is totally random. Such random setting enables our model to complete image with arbitrary mask size and location. After choosing the mask size and location, we shift and rescale the pixel value from [0, 255] to [-1, 1] and fill in the masked regions with 0. Finally, the corrupted image and the mask are concatenated as input.
Training Procedure. Instead of training the model with the joint loss all together directly, we add them gradually. First, we train the generator with reconstruction loss to synthesize coarse contents. Then adversarial loss and perceptual loss are added to further refine the results. We set learning rate with polynomial decay from 10^{-3} to 10^{-6}.

Hyperparameter. During training, it is extremely important to fine-tune the hyperparameters so that we can make a good balance among different losses. In backpropagation, it is gradient values that make sense rather than loss values. Thus we propose a strategy to effectively select proper hyperparameters, which first trains the model for several steps to gain their gradients, then sets hyperpa-

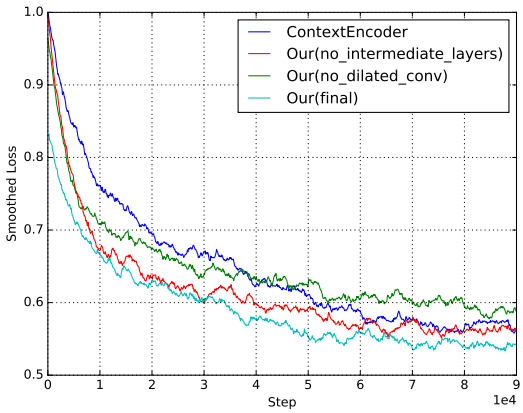


Figure 7. Training loss curves of different generator structures. We train each model to regress the missing central region with reconstruction loss only on Paris StreetView.

(a) original (b) masked (c) no replace (d) replace

Figure 8. The effect of replacing unmasked regions with original pixels.

rameters according to their relative magnitude. Choosing hyperparameters based on gradients enables us to know the exact effects of different losses.

4.3. Comparison of Generator Structures

As shown in Figure 7, dilated convolution layers can improve performance a lot, because it helps increase receptive field, letting generator see a 'larger' area. Intermediate layers combine multi-scale information, and the connection between low layers and high layers can ease the problem of gradient vanish, thus we also gain performance improvement. Compared with the generator of Context Encoder, our generator achieves better results with less parameters, demonstrating the superiority of our generator structure.

4.4. Effect of Original Image Context

The aim of image completion is to synthesize contents that fit the original image context well. We conduct an experiment to show the effect of image context, which employs two different input settings for the input of discriminator: replace unmasked regions with original pixels or no replace. As shown in Figure 8, when utilizing the original image context, the synthesized contents are more continuous on the boundary and the image completion results are more natural.

4.5. Analysis and Comparison of Perceptual Features

Before using perceptual features, we need to know the exact effects of different levels of perceptual features. We make detailed comparison and analysis. As shown in Figure 4, when using perceptual features extracted from $conv1\_2$ and $conv2\_2$, the inpainting results are still blurry. This is because $conv1\_2$ and $conv2\_2$ only capture low-level information without capturing some characteristics of structures and shapes. However, (d)(e)(f) have clear face and building structures, showing that higher layers such as $conv3\_2$, $conv4\_2$ and $conv5\_2$ can capture some semantic information which helps generate details. To combine the benefits of them, we use the ReLU output of layers $conv3\_2$, $conv4\_2$ and $conv5\_2$ in VGG16 as perceptual loss in our experiments, which achieves the best performance Figure 4(f).

4.6. Qualitative Comparisons

We first compare our method with Photoshop Content Aware Fill (CAF, based on PatchMatch [3]) and Generative Face Completion (GFC, [18]) on CelebA (Figure 5). We do not compare with Context Encoder (CE, [22]) here, because CE can only work on the same masks used in training, while our evaluations here focus on cases with random mask size and location. Traditional methods such as CAF, cannot handle highly specific and complex structures such as faces, because every face has different structure and copy-and-paste strategy will not work. GFC can synthesize some key components such as nose, but it will fail if the faces are not aligned, such as the first row and the last row. Moreover, as shown in Figure 5(d) and 5(e), the synthesized components is not consistent with surrounding regions, which needs some post-processing methods to eliminate. By contrast, our method is not sensitive to position and face does not need to be aligned, because both the mask size and location is random during our training, and our network is a FCN, which has the property of translation invariance.

We then compare our method with CAF and CE on Paris StreetView (Figure 6). Our method can work well on any mask location. Since CE can only work on specific masks, we compare the results on the central region. CAF can work

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

Figure 9. Object removal examples on CelebA and Paris Street.

well on texture structures, such as the third row in Figure 6, but it does not capture the semantic information. Compared with CE, our results are consistent on the boundary and have fine-grained details.

Figure 5(d), 5(f) and Figure 6(d), 6(e) are all trained with reconstruction loss and adversarial loss. The results show the superiority of our network structure, which can synthesize more consistent and realistic contents. Comparison of Figure 5(f), 5(g) and Figure 2(e), 2(f) describe the advantages of perceptual loss, which can help synthesis details.

4.7. Quantitative Comparisons

We compare our method with baseline methods Context Encoder [22] on Paris StreetView dataset. As shown in Table 1, our method can achieve better results. Our method has nearly no performance decrease in the random setting, while Context Encoder decreases a lot. The results show that our method is capable for image completion with random mask location.

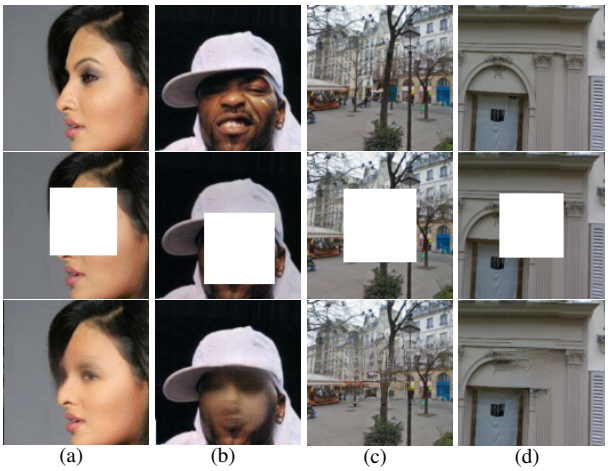
Compared with the results of $L_r + L_a$, our final results has comparable performance, because semantic inpainting is not for recovering the original images, but try to fill in the missing regions with realistic contents, as shown in Figure 5, Figure 6 and Figure 9. Therefore the quantitative evaluation may not be an effective metric.

4.8. Object Removal

Our algorithm can also be employed to remove unwanted objects, some examples are shown in Figure 9. The red object in the nose of 9(a), the beard of 9(b), the people in 9(c), the brown door of 9(d) are removed, but the results still look natural. In Figure 9(d), there are red banners on top of the two rooms in the right corner, therefore the synthesized center door also has a red banner, which shows the reasoning ability of our algorithm.

Table 1. Comparisons with Context Encoder [22] on Paris StreetView Dataset. Mask size is 56×56, up/down are results of center/random region completion.

Methods	Mean L_1 Loss	Mean L_2 Loss	PSNR
Context Encoder	0.1487	0.0545	20.4878dB
	0.2205	0.1005	17.3369dB
Ours($L_r + L_a$)	0.1320	0.0456	21.3919dB
	0.1338	0.0472	21.3863dB
Ours(final)	0.1310	0.0436	21.5338dB
	0.1334	0.0460	21.5059dB



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

Figure 10. Failure cases of our approach.

4.9. Limitations

Although our approach could synthesize semantically consistent results with fine-grained details and generate novel contents that do not appear in the dataset, it could fail if there are few or no scenes in the dataset. Figure 10 shows several failure cases. CelebA has very few profile images, so our algorithm does not handle Figure 10(a) well. Figure 10(b) has a hat and the whole face is masked with nearly no pixels of skin color left. The algorithm does not find that the tree in Figure 10(c) is crooked and does not utilize the symmetry property to complete arch structure of Figure 10(d), because most images in Paris StreetView are buildings and streets.

5. Conclusion

In this paper, we present a novel approach for semantically consistent image completion. We make a detailed analysis and comparison on the effects of different losses and different levels of perceptual features and show that there exist complementarity between adversarial training and perceptual features. We design a novel lightweight generator structure and show the superiority of our method on two benchmark datasets.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 6

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24–1, 2009. 1, 3, 7

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 1, 3

[5] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003. 1, 3

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1486–1494, 2015. 2

[7] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012. 6

[8] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016. 3

[9] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005. 1, 3

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 2

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 4

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 2

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016. 2

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[16] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014. 1

[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016. 2

[18] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. *arXiv preprint arXiv:1704.05838*, 2017. 1, 3, 7

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 6

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 4

[21] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*, pages 2802–2810, 2016. 4

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 1, 3, 6, 7, 8

[23] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 3

[24] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 3

[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 5

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[27] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016. 2

[28] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012. 1

[29] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016. 1, 3

[30] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 4