Globally and Locally Consistent Image Completion

SATOSHI IIZUKA, Waseda University
EDGAR SIMO-SERRA, Waseda University
HIROSHI ISHIKAWA, Waseda University

Fig. 1. Image completion results by our approach. The masked area is shown in white. Our approach can generate novel fragments that are not present elsewhere in the image, such as needed for completing faces; this is not possible with patch-based methods.

We present a novel approach for image completion that results in images that are both locally and globally consistent. With a fully-convolutional neural network, we can complete images of arbitrary resolutions by filling-in missing regions of any shape. To train this image completion network to be consistent, we use global and local context discriminators that are trained to distinguish real images from completed ones. The global discriminator looks at the entire image to assess if it is coherent as a whole, while the local discriminator looks only at a small area centered at the completed region to ensure the local consistency of the generated patches. The image completion network is then trained to fool the both context discriminator networks, which requires it to generate images that are indistinguishable from real ones with regard to overall consistency as well as in details. We show that our approach can be used to complete a wide variety of scenes. Furthermore, in contrast with the patch-based approaches such as PatchMatch, our approach can generate fragments that do not appear elsewhere in the image, which allows us to naturally complete the images of objects with familiar and highly specific structures, such as faces.

CCS Concepts: • **Computing methodologies** → **Image processing**; *Neural networks*;

Additional Key Words and Phrases: image completion, convolutional neural network

This work was partially supported by JST ACT-I Grant Number JPMJPR16U3 and JST CREST Grant Number JPMJCR14D1.
© 2017 Copyright held by the owner/author(s). This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, https://doi.org/http://dx.doi.org/10.1145/3072959.3073659.

ACM Reference format:
Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. *ACM Trans. Graph.* 36, 4, Article 107 (July 2017), 13 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073659

1 INTRODUCTION

Image completion is a technique that allows filling-in target regions with alternative contents. This allows removing unwanted objects or generating occluded regions for image-based 3D reconstruction. Although many approaches have been proposed for image completion, such as patch-based image synthesis [Barnes et al. 2009; Darabi et al. 2012; Huang et al. 2014; Simakov et al. 2008; Wexler et al. 2007], it remains a challenging problem because it often requires high-level recognition of scenes. Not only is it necessary to complete textured patterns, it is also important to understand the anatomy of the scene and objects being completed. Based on this observation, in this work we consider both the local continuity and the global composition of the scene, in a single framework for image completion.

We leverage a fully convolutional network as the basis of our approach, and propose a novel architecture that results in both locally and globally consistent natural image completion. Our architecture is composed of three networks: a completion network, a global context discriminator, and a local context discriminator. The completion network is fully convolutional and used to complete

the image, while both the global and the local context discriminators are auxiliary networks used exclusively for training. These discriminators are used to determine whether or not an image has been completed consistently. The global discriminator takes the full image as input to recognize global consistency of the scene, while the local discriminator looks only at a small region around the completed area in order to judge the quality of more detailed appearance. During each training iteration, the discriminators are updated first so that they correctly distinguish between real and completed training images. Afterwards, the completion network is updated so that it fills the missing area well enough to fool the context discriminator networks. As we will show, using both the local and the global context discriminators is critical for obtaining realistic image completion.

We evaluate and compare our approach with existing methods on a large variety of scenes. We also show results on more challenging specific tasks, such as face completion, in which our approach can generate image fragments of objects such as eyes, noses, or mouths to realistically complete the faces. We evaluate the naturalness of this challenging face completion with a user study, where the difference between our results and real faces is indiscernible 77% of the time.

In summary, in this paper we present:

- a high performance network model that can complete arbitrary missing regions,
- a globally and locally consistent adversarial training approach for image completion, and
- results of applying our approach to specific datasets for more challenging image completion.

2 RELATED WORK

A variety of different approaches have been proposed for the image completion task. One of the more traditional approaches is that of diffusion-based image synthesis. This technique propagates the local image appearance around the target holes to fill them in. For example, the propagation can be performed based on the isophote direction field [Ballester et al. 2001; Bertalmio et al. 2000], or global image statistics based on the histograms of local features [Levin et al. 2003]. However, diffusion-based approaches, in general, can only fill small or narrow holes, such as scratches found commonly in old photographs.

In contrast to the diffusion-based techniques, patch-based approaches have been able to perform more complicated image completion that can fill large holes in natural images. Patch-based image completion was first proposed for texture synthesis [Efros and Leung 1999; Efros and Freeman 2001], in which texture patches are sampled from a source image and then pasted into a target image. This was later extended with image stitching [Kwatra et al. 2003] with graph cuts and texture generation [Kwatra et al. 2005] based on energy optimization. For image completion, several modifications such as optimal patch search have been proposed [Bertalmio et al. 2003; Criminisi et al. 2004; Drori et al. 2003]. In particular, Wexler *et al.* [2007] and Simakov *et al.* [2008] proposed a global-optimization-based method that can obtain more consistent fills. These techniques were later accelerated by a randomized patch search algorithm called PatchMatch [Barnes et al. 2009, 2010], which allows for real-time high-level image editing of images. Darabi *et al.* [2012] demonstrated improved image completion by integrating image gradients into the distance metric between patches. However, these methods depend on low-level features such as the sum of squared differences of patch pixel values, which are not effective to fill in holes on complicated structures. Furthermore, they are unable to generate novel objects not found in the source image, unlike our approach.

Table 1. Comparison of different approaches for completion. Patch-based approaches such as [Barnes et al. 2009] cannot generate new texture or objects and only look at local similarity without taking into account the semantics of the scene. The context encoder [Pathak et al. 2016] handles only images of small fixed size without maintaining local consistency with the surrounding region. In contrast, our method can complete images of any size, generating new texture and objects according to the local and global structures of the scenes.

	Patch-based	Context encoder	Ours
Image size	**Any**	Fixed	**Any**
Local Consistency	**Yes**	No	**Yes**
Semantics	No	**Yes**	**Yes**
Novel objects	No	**Yes**	**Yes**

To tackle the problem of generating large missing regions of structured scenes, there are some approaches that use structure guidance, which are generally specified manually, to preserve important underlying structures. This can be done by specifying points of interest [Drori et al. 2003], lines or curves [Barnes et al. 2009; Sun et al. 2005], and perspective distortion [Pavić et al. 2006]. Approaches for automatic estimation of the scene structure have also been proposed: utilizing the tensor-voting algorithm to smoothly connect curves across holes [Jia and Tang 2003]; exploiting structure-based priority for patch ordering [Criminisi et al. 2004], tile-based search space constraints [Kopf et al. 2012], statistics of patch offsets [He and Sun 2012], and regularity in perspective planar surfaces [Huang et al. 2014]. These approaches improve the quality of the image completion by preserving important structures. However, such guidances are based on the heuristic constraints of specific types of scenes and thus are limited to specific structures.

The obvious limitation of most existing patch-based approaches is that the synthesized texture only comes from the input image. This is a problem when a convincing completion requires textures that are not found in the input image. Hays and Efros [2007] proposed an image completion method using a large database of images. They first search for the image most similar to the input in the database, and then complete the image by cutting the corresponding regions from the matched image and pasting them into the holes. However, this assumes that the database contains an image similar to the input image, which may not be the case. This was also extended to the particular case in which images of exactly the same scene are included in the database of images [Whyte et al. 2009]. However, the assumption that the exact same scene is included limits the applicability greatly in comparison to general approaches.

Completion of human faces has also received attention as a particular application of inpainting. Mohammed *et al.* [2009] build a patch library using a dataset of faces and propose a global and local parametric model for face completion. Deng *et al.* [2011] use a

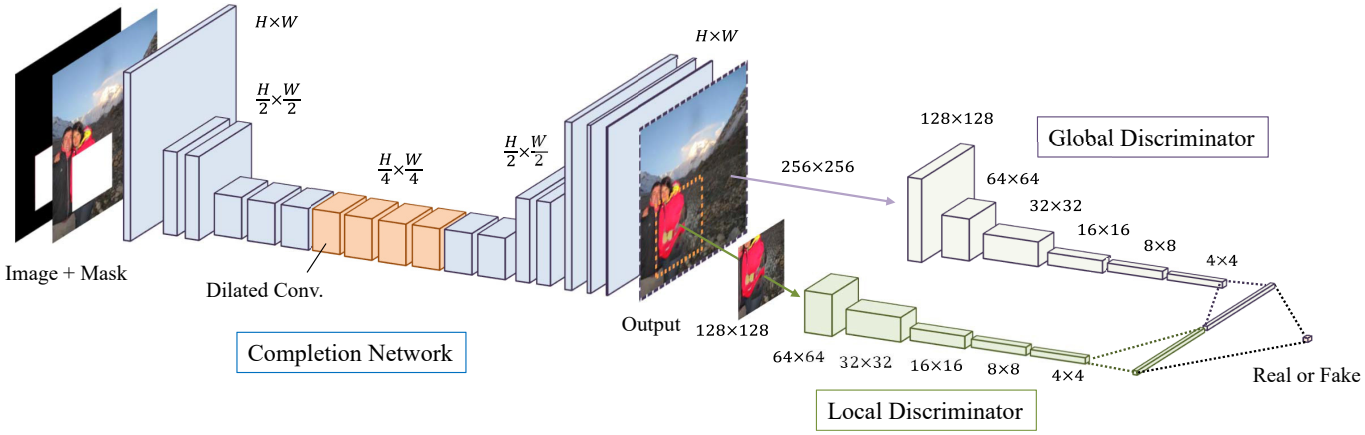


Fig. 2. Overview of our architecture for learning image completion. It consists of a completion network and two auxiliary context discriminator networks that are used only for training the completion network and are not used during the testing. The global discriminator network takes the entire image as input, while the local discriminator network takes only a small region around the completed area as input. Both discriminator networks are trained to determine if an image is real or completed by the completion network, while the completion network is trained to fool both discriminator networks.

spectral-graph-based algorithm for face image repairing. However, these approaches require aligned images for learning patches, and do not generalize to the arbitrary inpainting problem.

Convolutional Neural Networks (CNNs) have also been used for image completion. Initially, CNN-based image inpainting approaches were limited to very small and thin masks [Köhler et al. 2014; Ren et al. 2015; Xie et al. 2012]. Similar approaches have also been applied to MRI and PET images for completing missing data [Li et al. 2014]. More recently, and concurrently to this work, Yang *et al.* [2017] also proposed a CNN based optimization approach for inpainting. However, unlike our approach, this has an increased computation time due to having to optimize for every image.

We build upon the recently proposed Context Encoder (CE) [Pathak et al. 2016], that extended CNN-based inpainting to large masks, and proposed a context encoder to learn features by inpainting, based on Generative Adversarial Networks (GAN) [Goodfellow et al. 2014]. The original purpose of GAN is to train generative models using convolutional neural networks. These generator networks are trained by using an auxiliary network, called *discriminator*, which serves to distinguish whether an image is generated by a network or is real. The generator network is trained to fool the discriminator network, while the discriminator network is updated in parallel. By using a Mean Squared Error (MSE) loss in combination with a GAN loss, Pathak et al. [2016] were able to train an inpainting network to complete a 64×64 pixel area in the center of 128×128 pixel images, avoiding the blurring common with using only MSE losses. We extend their work to handle arbitrary resolutions by using a fully convolutional network, and significantly improve the visual quality by employing both a global and local discriminator.

One of the main issues of GAN is the instability during learning, which has led to numerous research on the topic [Radford et al. 2016; Salimans et al. 2016]. We avoid this issue by not training purely generative models and tuning the learning process to prioritize stability. Additionally, we have heavily optimized the architecture and the training procedure specifically for the image completion problem. In particular, we do not use a single discriminator but two: a global discriminator network and a local discriminator network. As we show, this proves critical in obtaining semantically and locally coherent image completion results.

Our approach can overcome the limitations of the existing approaches and realistically complete diverse scenes. A high-level comparison of different approaches can be seen in Table 1. On one hand, the patch-based approaches [Barnes et al. 2009, 2010; Darabi et al. 2012; Huang et al. 2014; Wexler et al. 2007] show high quality reconstructions for arbitrary image sizes and masks; however, they are unable to provide novel image fragments not found elsewhere in the image nor have a high level semantic understanding of the image: they only local at similarity on a local patch level. On the other hand, the context encoder-based approach [Pathak et al. 2016] can generate novel objects, but are limited to fixed low resolution images. Furthermore, the approach can lack local consistency as the continuity of the completed region with the surrounding area is not taken into account. Our approach can deal with arbitrary image sizes and masks, while being consistent with the image and able to generate novel objects.

3 APPROACH

Our approach is based on deep convolutional neural networks trained for the image completion task. A single completion network is used for the image completion. Two additional networks, the global and the local context discriminator networks, are used in order to train this network to realistically complete images. During the training, the discriminator networks are trained to determine whether or not an image has been completed, while the completion network is trained to fool them. Only by training all the three networks together is it possible for the completion network to realistically complete a diversity of images. An overview of this approach can be seen in Fig. 2.

3.1 Convolutional Neural Networks

Our approach is based on Convolutional Neural Networks [Fukushima 1988; LeCun et al. 1989]. These are a special variant of neural network based on using convolution operators that conserve the spatial structure of the input, generally consisting of images. These networks are formed by layers in which a bank of filters is convoluted with the input map to produce an output map which is further processed with a non-linear activation function, most often the Rectified Linear Unit (ReLU), defined as $\sigma(\cdot) = max(\cdot, 0)$ [Nair and Hinton 2010].

Instead of using only the standard convolutional layers, we also employ a variant called the dilated convolution layers [Yu and Koltun 2016], which allow increasing the area each layer can use as input. This is done without increasing the number of learnable weights by spreading the convolution kernel across the input map. More specifically, if one 2D layer is a C-channel $h \times w$ map and the next layer is a C'-channel $h' \times w'$ map, the dilated convolution operator can be written for each pixel as:

$$\mathbf{y}_{u,v} = \sigma\left(\mathbf{b} + \sum_{i=-k'_{\mathrm{h}}}^{k'_{\mathrm{h}}} \sum_{j=-k'_{\mathrm{w}}}^{k'_{\mathrm{w}}} W_{k'_{\mathrm{h}}+i,k'_{\mathrm{w}}+j}\, \mathbf{x}_{u+\eta i,v+\eta j}\right),$$
$$k'_{\mathrm{h}} = \frac{k_{\mathrm{h}}-1}{2}, \qquad k'_{\mathrm{w}} = \frac{k_{\mathrm{w}}-1}{2}, \tag{1}$$

where k_{w} and k_{h} are the kernel width and height (odd numbers), respectively, η is the dilation factor, $\mathbf{x}_{u,v} \in \mathbb{R}^C$ and $\mathbf{y}_{u,v} \in \mathbb{R}^{C'}$ are the pixel component of the input and the output of the layer, $\sigma(\cdot)$ is a component-wise non-linear transfer function, $W_{s,t}$ are C'-by-C matrices of the kernel, and $\mathbf{b} \in \mathbb{R}^{C'}$ is the layer bias vector. With $\eta = 1$ the equation becomes the standard convolution operation.

These networks are then trained to minimize a loss function with back-propagation [Rumelhart et al. 1986], and are trained by using datasets which consist of input and output pairs. The loss function usually tries to minimize the distance between the network output and the corresponding output pair in the dataset.

3.2 Completion Network

The completion network is based on a fully convolutional network. An overview of the network model architecture can be seen in Table 2. The input of the completion network is an RGB image with a binary channel that indicates the image completion mask (1 for a pixel to be completed), and the output is an RGB image. As we do not wish any change in areas other than the completion regions, the output pixels outside of the completion regions are restored to the input RGB values. The general architecture follows an encoder-decoder structure, which allows reducing the memory usage and computational time by initially decreasing the resolution before further processing the image. Afterwards, the output is restored to the original resolution using deconvolution layers [Long et al. 2015], which consist of convolutional layers with fractional strides. Unlike other architectures that use many pooling layers to decrease the resolution, our network model only decreases the resolution twice, using strided convolutions to 1/4 of the original size, which is important to generate non-blurred texture in the missing regions.

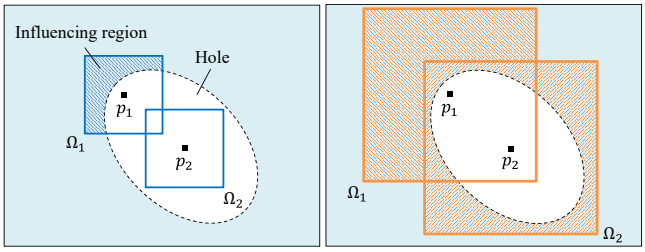


Fig. 3. Importance of spatial support. In order to be able to complete large regions, the spatial support used to compute an output pixel must include pixels outside of the hole. On the left, the pixel p_1 is computed from the influencing region in the spatial support Ω_1, while the pixel p_2 cannot be calculated since the supporting area Ω_2 does not contain any information outside of the hole. However, on the right side, the spatial support is larger than the hole, allowing the completion of the center pixels.

Dilated convolutional layers [Yu and Koltun 2016] are also used in the mid-layers (Eq. (1) with $\eta > 1$). Dilated convolutions use kernels that are spread out, allowing to compute each output pixel with a much larger input area, while still using the same amount of parameters and computational power. This is important for the image completion task, as the context is critical for realism. By using dilated convolutions at lower resolutions, the model can effectively "see" a larger area of the input image when computing each output pixel than with standard convolutional layers. The resulting network model computes each output pixel under the influence of a 307×307-pixel region of the input image. Without using dilated convolutions, it would only use a 99×99-pixel region, not allowing the completion of holes larger than 99×99 pixels, as depicted in Fig. 3.

3.3 Context Discriminators

A global context discriminator network and a local context discriminator network have the objective of discerning whether an image is real or has been completed. The networks are based on convolutional neural networks that compress the images into small feature vectors. Outputs of the networks are fused together by a concatenation layer that predicts a continuous value corresponding to the probability of the image being real. An overview of the networks can be seen in Table 3.

The global context discriminator takes as an input the entire image rescaled to 256×256 pixels. It consists of six convolutional layers and a single fully-connected layer that outputs a single 1024-dimensional vector. All the convolutional layers employ a stride of 2×2 pixels to decrease the image resolution while increasing the number of output filters. In contrast with the completion network, all convolutions use 5×5 kernels.

The local context discriminator follows the same pattern, except that the input is a 128×128-pixel image patch centered around the completed region. (Note that, at the training time, there is always a single completed region. The trained completion network can, however, fill-in any number of holes at the same time.) In the case the image is not a completed image, a random patch of the image is selected, as there is no completed region to center it on. As the initial input resolution is half of the global discriminator, the first layer used in the global discriminator is not necessary. The output

Table 2. Architecture of the image completion network. After each convolution layer, except the last one, there is a Rectified Linear Unit (ReLU) layer. The output layer consists of a convolutional layer with a sigmoid function instead of a ReLU layer to normalize the output to the [0, 1] range. "Outputs" refers to the number of output channels for the output of the layer.

Type	Kernel	Dilation (η)	Stride	Outputs
conv.	5×5	1	1×1	64
conv.	3×3	1	2×2	128
conv.	3×3	1	1×1	128
conv.	3×3	1	2×2	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
dilated conv.	3×3	2	1×1	256
dilated conv.	3×3	4	1×1	256
dilated conv.	3×3	8	1×1	256
dilated conv.	3×3	16	1×1	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
deconv.	4×4	1	$1/2 \times 1/2$	128
conv.	3×3	1	1×1	128
deconv.	4×4	1	$1/2 \times 1/2$	64
conv.	3×3	1	1×1	32
output	3×3	1	1×1	3

is a 1024-dimensional vector representing the local context around the completed region.

Finally, the outputs of the global and the local discriminators are concatenated together into a single 2048-dimensional vector, which is then processed by a single fully-connected layer, to output a continuous value. A sigmoid transfer function is used so that this value is in the $[0, 1]$ range and represents the probability that the image is real, rather than completed.

3.4 Training

Let $C(x, M_c)$ denote the completion network in a functional form, with x the input image and M_c the completion region mask that is the same size as the input image. The binary mask M_c takes the value 1 inside regions to be filled-in and 0 elsewhere. As a preprocessing, C overwrites the completion region of the training input image x by a constant color, which is the mean pixel value of the training dataset, before putting it into the network. Similarly, $D(x, M_d)$ denotes the combined context discriminators in a functional form.

In order to train the network to complete the input image realistically, two loss functions are jointly used: a weighted Mean Squared Error (MSE) loss for training stability, and a Generative Adversarial Network (GAN) [Goodfellow et al. 2014] loss to improve the realism of the results. Using the mixture of the two loss functions allows the stable training of the high performance network model, and has been used for image completion [Pathak et al. 2016], and concurrently with this work, for various image-to-image translation problems [Isola et al. 2017]. Training is done with backpropagation [Rumelhart et al. 1986].

Table 3. Architectures of the discriminators used in our network model. Fully-Connected (FC) layers refer to the standard neural network layers. The output layer consists of a filly-connected layer with a sigmoid transfer layer that outputs the probability that an input image came from real images rather than the completion network.

(a) Local Discriminator

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

(b) Global Discriminator

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

(c) Concatenation layer

Type	Kernel	Stride	Outputs
concat.	-	-	2048
FC	-	-	1

In order to stabilize the training, a weighted MSE loss considering the completion region mask is used [Pathak et al. 2016]. The MSE loss is defined by:

$$L(x, M_c) = \| M_c \odot (C(x, M_c) - x) \|^2 \ , \tag{2}$$

where $\odot$ is the pixelwise multiplication and $\| \cdot \|$ is the Euclidean norm.

The context discriminator networks also work as a kind of loss, sometimes called the GAN loss [Goodfellow et al. 2014]. This is the crucial part of training in our approach, and involves turning the standard optimization of a neural network into a min-max optimization problem in which at each iteration the discriminator networks are jointly updated with the completion network. For our completion and context discriminator networks, the optimization becomes:

$$\min_C \max_D \ \mathbb{E}[\ \log D(x, M_d) + \log(1 - D(C(x, M_c), M_c) \ ] \ , \tag{3}$$

where M_d is a random mask, M_c is the input mask, and the expectation value is just the average over the training images x.

By combining the two loss functions, the optimization becomes:

$$\begin{aligned} \min_C \max_D \ \mathbb{E}[\ & L(x, M_c) + \log D(x, M_d) \\ & + \alpha \log(1 - D(C(x, M_c), M_c)) \ ], \end{aligned} \tag{4}$$

where α is a weighing hyper parameter.

During the course of the optimization, the completion and the discriminator networks written here as C and D change, which actually means that the weights and the biases of the networks change. Let us denote the parameters of the completion network C by θ_C. In the standard stochastic gradient descent, the above min-max optimization then means that, for training C, we take the gradient of the loss function with respect to θ_C and update the

Algorithm 1 Training procedure of the image completion network.

while iterations $t < T_{train}$ **do**
  Sample a minibatch of images x from training data.
  Generate masks M_c with random holes for each image x in the minibatch.
  if $t < T_C$ **then**
    Update the completion network C with the weighted MSE loss (Eq. (2)) using (x, M_c).
  else
    Generate masks M_d with random holes for each image x in the minibatch.
    Update the discriminators D with the binary cross entropy loss with both $(C(x, M_c), M_c)$ and (x, M_d).
    if $t > T_C + T_D$ **then**
      Update the completion network C with the joint loss gradients (Eq. (5)) using (x, M_c), and D.
    end if
  end if
end while

parameters so that the value of the loss function decreases. The gradient is:

$$\mathbb{E}[\ \nabla_{\theta_C} L(x, M_c) + \alpha \nabla_{\theta_C} \log(1 - D(C(x, M_c), M_c))\ ]\ . \qquad (5)$$

In practice, we take a more fine-grained control, such as initially keeping the norm of the MSE loss gradient roughly the same order of magnitude as the norm of the discriminator gradient. This helps stabilize the learning.

We also update the discriminator networks D similarly, except we take update in the opposite direction so that the loss increases. Note that here D consists of the local and the global context discriminators. So the flow of the gradient in backpropagation initially splits into the two networks and then merge into the completion network.

In optimization, we use the ADADELTA algorithm [Zeiler 2012], which sets a learning rate for each weight in the network automatically.

3.5 Stable Training

During the training, the context discriminators are trained to distinguish fake from real images, while the completion network is trained to deceive the discriminators. As the optimization consists of jointly minimizing and maximizing conflicting objectives, it is not very stable. Unlike other approaches that focus on image generation [Salimans et al. 2016], our method does not generate images from noise. That helps the training process to be initially more stable. However, since the image completion task itself is very challenging, much care has to be still taken in order to train the networks to convergence.

An overview of the general training procedure can be seen in Algorithm 1. The training is split into three phases: first, the completion network is trained with the MSE loss from Eq. (2) for T_C iterations. Afterwards, the completion network is fixed and the discriminators are trained from scratch for T_D iterations. Finally, both the completion network and content discriminators are trained jointly until the end of training. The pretraining of the completion and the discriminator networks has proved critical for successful training.

In order to facilitate the propagation of gradients through the network, during training we use the batch normalization layers [Ioffe and Szegedy 2015] after all convolutional layers except for the last layers of both the completion and the discriminator networks. This normalizes the output of each layer using output statistics that are updated online. During testing, they can be integrated into the preceding convolutional layer, so as not to add computational burden.

Training is done by resizing images so that the smallest edge is a random value in the [256, 384] pixel range. Afterwards, a random 256×256-pixel patch is extracted and used as the input image. For the mask, we generate a random hole in the [96, 128] pixel range and fill it with the mean pixel value of the training dataset. Note that the aspect ratio of this hole can vary as the width and height are drawn separately. The input for the global context discriminator is the full 256×256-pixel image, and for the local context discriminator the input is a 128×128-pixel patch centered around the completed region (or a random area for real samples).

Simple post-processing. Although our network model can plausibly fill missing regions, sometimes the generated area has subtle color inconsistencies with the surrounding regions. To avoid this, we perform simple post-processing by blending the completed region with the color of the surrounding pixels. In particular, we employ the fast marching method [Telea 2004], followed by Poisson image blending [Pérez et al. 2003].

Table 4. Analysis of computation time of our model. We notice a significant speedup when using the GPU that drives computation times down to under a second even for large input images.

Image Size	Pixels	CPU (s)	GPU (s)	Speedup
512×512	409,600	2.286	0.141	16.2×
768×768	589,824	4.933	0.312	15.8×
1024×1024	1,048,576	8.262	0.561	14.7×

4 RESULTS

We train our model using 8,097,967 training images taken from the Places2 dataset [Zhou et al. 2016]. This dataset includes images of a diversity of scenes and was originally meant for scene classification. We set the weighting hyper-parameter to $\alpha = 0.0004$, and train using a batch size of 96 images. The completion network is trained for $T_C = 90,000$ iterations; then the discriminator is trained for $T_D = 10,000$ iterations; and finally both are jointly trained to reach the total of $T_{train} = 500,000$ iterations. The entire training procedure takes roughly 2 months on a single machine equipped with four K80 GPUs.

We evaluate our model using images from a wide variety of scenes not used in the training data, and compare with the existing approaches, demonstrating the performance of our method. Unless otherwise mentioned, our models are trained on the Places2 dataset.

Computational time. Processing time of image completion depends on the resolution of the input image, not on the size of the region to be completed. Table 4 shows the computation time for

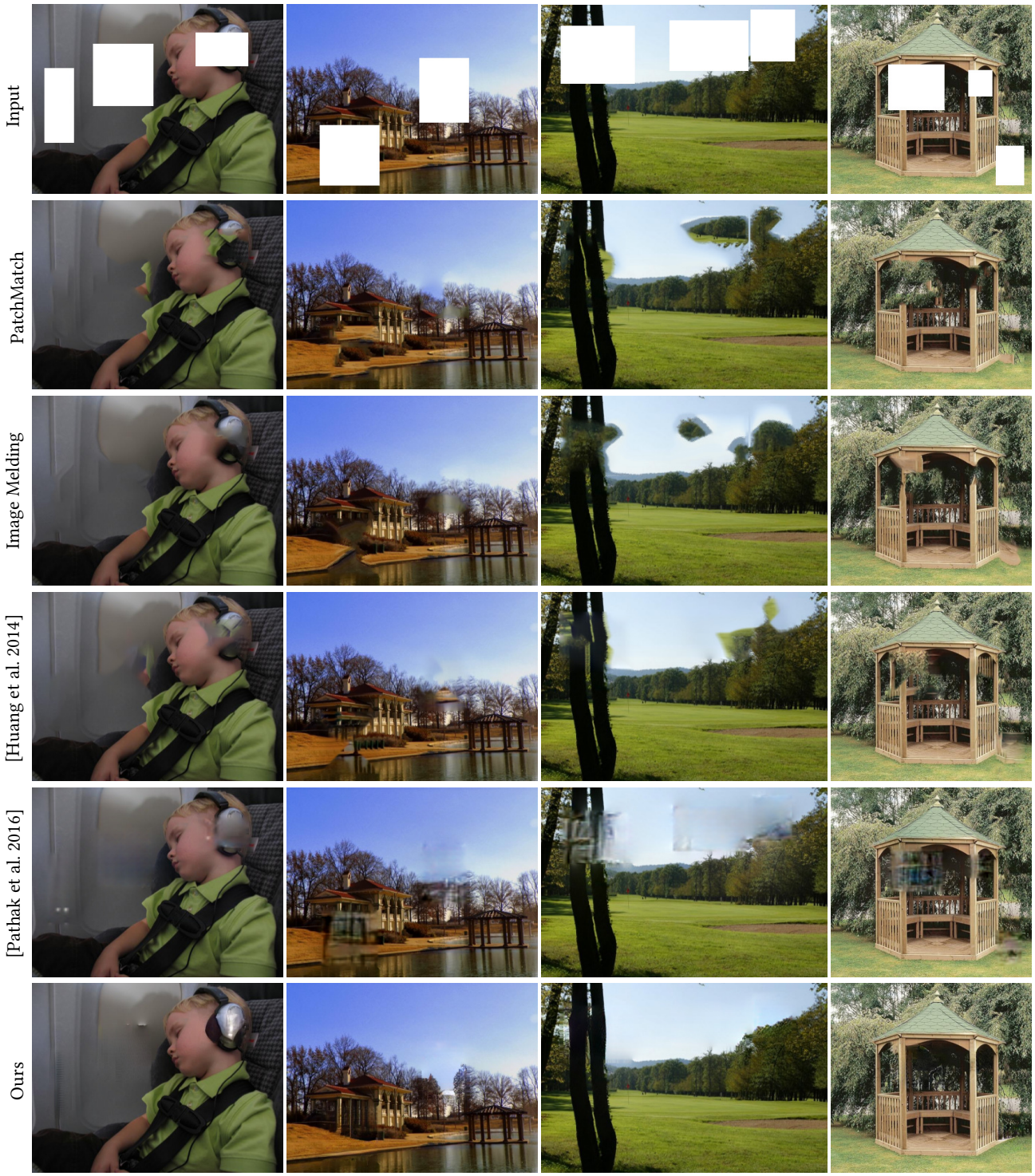


Fig. 4. Comparisons with existing works. We compare with Photoshop Content Aware Fill (PatchMatch), Image Melding, [Huang et al. 2014], and [Pathak et al. 2016] on a diverse set of scenes from the Places2 validation set using random masks. For the comparison, we have retrained the model of [Pathak et al. 2016] on the Places2 dataset for arbitrary region completion. Furthermore, we use the same post-processing as used for our results. We can see that, while PatchMatch and Image Melding generate locally consistent patches extracted from other parts of the image, they are not globally consistent with the other parts of the scene, *e.g.*, trees float in mid-air. The approach of [Pathak et al. 2016] can inpaint novel regions, but the inpainted region tends to be easy to identify, even with our post-processing. Our approach, designed to be both locally and globally consistent, results in much more natural scenes.

several resolutions. We evaluate both on CPU and GPU using an Intel Core i7-5960X CPU @ 3.00 GHz with 8 cores and NVIDIA GeForce TITAN X GPU. Even large images can be processed in under a second using a GPU.

4.1 Comparison with Existing Work

We evaluate our approach on both the general arbitrary region completion, and the center region completion task of [Pathak et al. 2016].

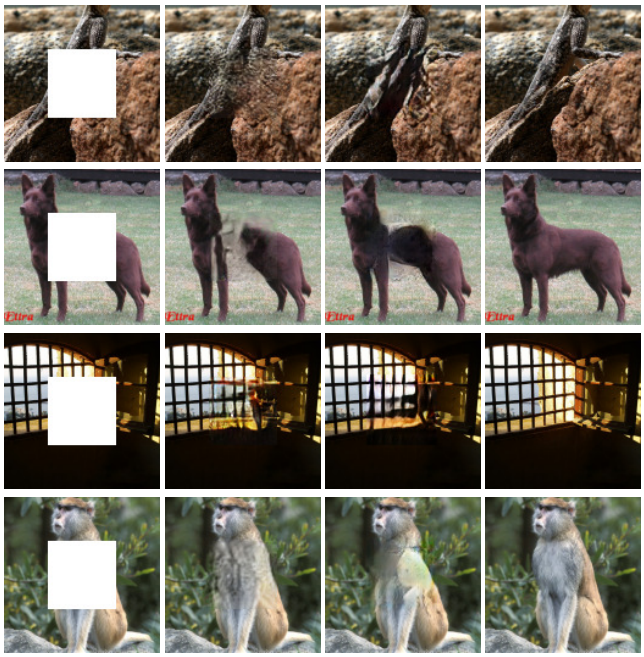
4.1.1 Arbitrary Region Completion. We compare our results with Photoshop Content Aware Fill that uses PatchMatch [Barnes et al. 2009], Image Melding [Darabi et al. 2012], [Huang et al. 2014], and [Pathak et al. 2016]. For the comparison, we have retrained the model of [Pathak et al. 2016] on the Places2 dataset for arbitrary masks for the same number of epochs as our model, and use the best performing model obtained during training. We evaluate it by resizing the images to its fixed input size, processing, resizing back to the original size, and restoring the pixels outside of the mask. Furthermore, we use the same post-processing as our approach, which is essential for obtaining results.

Results are shown in Fig. 4. The patch-based approaches are unable to generate novel objects in the image, unlike our approach. Furthermore, while they are able to complete with locally consistent image patches, they are not necessarily globally consistent with the scene, *e.g.*, objects may appear in mid-air or in the middle of other objects. The model of [Pathak et al. 2016] results in blurred and easy to identify areas, even with our post-processing. Our approach is explicitly trained to be both locally and globally consistent, leading to much more natural image completion.

4.1.2 Center Region Completion. We also compare with the Context Encoder (CE) [Pathak et al. 2016] on their provided 128×128-pixel test images, taken from ImageNet [Deng et al. 2009], with the fixed 64×64-pixel inpainting masks in the center of the image. For a fair comparison, we train our model using their training data, which consists of a subset of 100K images of ImageNet. We also do not perform post-processing for the results of our model. Results are shown in Fig. 5. For the center region completion task, the results of CE are significantly better than in the general arbitrary region completion case. We note that, while the CE approach is specialized to inpaint images of this size and fixed holes, our model is capable of arbitrary region completion at any resolution.

4.2 Global and Local Consistency

We investigate the influence of the global and the local context discriminators by training models that only use one of them and comparing with the full approach. We show the results in Fig. 6. We can see that, when the local discriminator is not used (b)(c), the result is completion by large blurred areas. On the other hand, while using only the local discriminator (d) results in locally more realistic textures, without the global discriminator it still lacks global consistency. By using both the global and the local discriminators, we can achieve results that are both locally and globally consistent.



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

Fig. 5. Comparison with the Context Encoder (CE) [Pathak et al. 2016] on images taken from the ImageNet validation set for center region completion. All images are resized to 128×128 pixels and the center 64×64 pixel region is completed. Both CE and Ours are trained on the same 100K subset of training images of ImageNet to complete the fixed center masks.

4.3 Effect of Post-Processing and Training Data

We show the effect of our simple post-processing in Fig. 7. We can see how this simple post-processing can be used to make the inpainted area blend better into the global image.

We also look at the effect of the dataset used for training our model. In particular, we compare models trained on Places2 [Zhou et al. 2016] and ImageNet [Deng et al. 2009]. The Places2 dataset consists of roughly 8 million images of scenes, while the ImageNet dataset focuses on classification on objects and only has 1 million images. Results are shown in Fig. 8. Although results are fairly similar, the results of the model trained on Places2 gives better performance in a wide diversity of scenarios, and is the primary model we use unless stated otherwise.

4.4 Object Removal

One of the main motivations of image completion is being able to remove unwanted objects in images. We show examples of object removal in Fig. 9. The results of our approach are natural and it becomes nearly impossible to identify where an object has been removed.

4.5 Faces and Facades

Although our model can generate various texture or objects to complete missing regions in general images, fine-tuning the model using a specific dataset can achieve even better results for more concrete and complicated image completion tasks. In particular, we

(a) Input (b) Weighted MSE (c) Global D (d) Local D (e) Full method

Fig. 6. Comparison of training with different discriminator configurations. We show the results of models trained with different discriminator configurations: (b) Weighted MSE (no discriminators), (c) using Weighted MSE and only a global discriminator, (d) using Weighted MSE and only a local discriminator, and (e) using Weighted MSE and both the global and the local discriminator.

(a) Input (b) Output (c) Post-processed

Fig. 7. Effect of our simple post-processing.

(a) Input (b) Ours (ImageNet) (c) Ours (Places2)

Fig. 8. Results of training with different datasets evaluated on public domain images taken from Flickr. In particular, we compare a model trained on the ImageNet dataset with one trained on the Places2 dataset.

consider both the CelebFaces Attributes Dataset (CelebA) [Liu et al. 2015], and the CMP Facade dataset [Radim Tyleček 2013], which consist of 202,599 and 606 images, respectively. For both datasets, we use the image completion network trained on the Places2 dataset and further train it on the new data. To adapt to new data, we initially train the context discriminator from scratch, then both the context discriminator and the completion network are trained together.

(a) Original (b) Input (c) Output

Fig. 9. Examples of object removal by our approach.

For the CelebA dataset, we train using 200,000 images and leave 2,599 images for evaluation. As the dataset has images of 178×218 pixels, we slightly adapt the training approach: instead of using 256×256-pixel image patches for training, we use 160×160-pixel image patches. We randomly generate holes in the [48, 96]-pixel range and thus modify the input of the local discriminator to be 96×96 pixels instead of 128×128 pixels. Finally, we remove a layer from the global context discriminator and adapt the fully-connected layers of both the global and the local context discriminators to the new training resolutions.

For the CMP Facade dataset, we train using 550 images and use 56 images for evaluation. The training procedure is the same as for the Places2 dataset, except that the completion network is initialized with the one trained on the Places2 dataset, instead of being trained with the MSE loss for T_C iterations.

We show the results in Fig. 10. We can see that our approach can realistically complete faces despite very large occluded areas. Note that patch-based approaches are unable to complete faces, as it requires the algorithm to generate novel objects such as eyes, noses, and mouths that are not already part of the image. We also see that

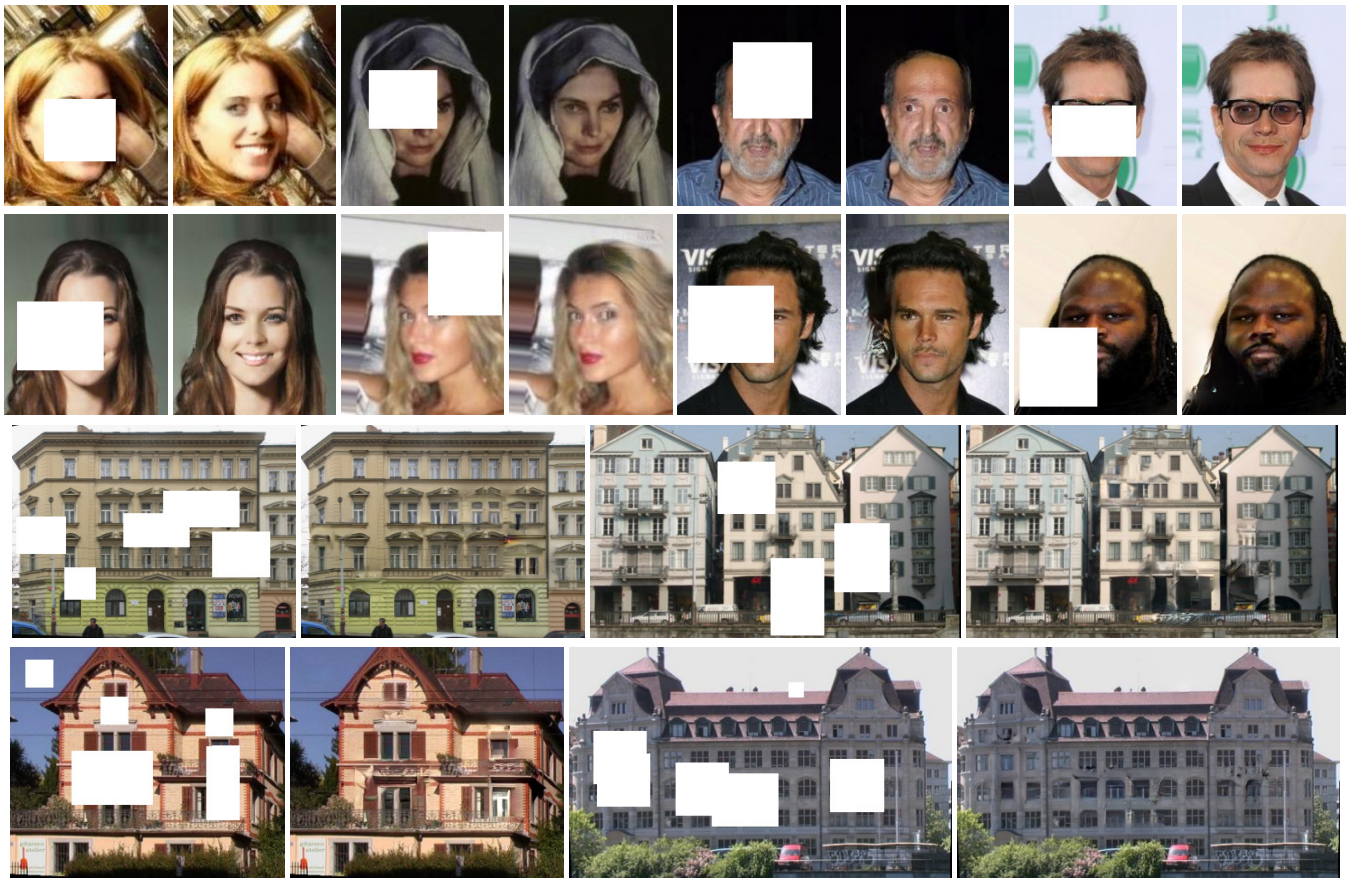
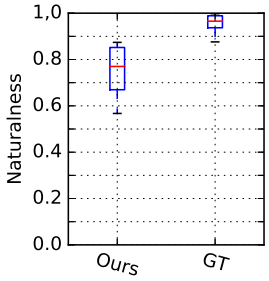


Fig. 10. Faces and Facades. We also apply our model to more specific datasets such as human faces and building facades by fine-tuning on different datasets. In the first two rows we show results of a model trained on the CelebA dataset, while the last two rows show results of a model trained on the CMP Facade dataset. For all results we use random inpainting masks.



Method	Naturalness (median)
GT	96.5%
Ours	77.0%

Fig. 11. Result of our user study evaluating the naturalness of the image completion on the CelebA dataset. The numbers are the percentage of the images that are deemed to be real by 10 different users for the Ground Truth (GT) and the result of the completion by our approach.

our approach can complete various types of facades in a way that they are both locally and globally coherent.

4.6 User Study

We perform a user study using the validation set of the CelebA dataset for the challenging face completion task and show results in Fig. 11. We ask 10 users to evaluate the naturalness of the completion. The users are only shown either the full completed image or a random image from the dataset, and asked to guess if the image is an actual image from the dataset or a completed one. The figure shows the percentage of the images that are deemed to be real. That is, 77.0% of the completed images by our approach is thought to be real. For comparison, the real images are correctly categorized 96.5% of the time. This highlights the realism of the resulting image completion by our approach.

4.7 Additional Results

We show additional results for our approach on the Places2 validation set in Fig. 12. Our approach can complete a wide diversity of scenes such as mountain ranges, close ups of walls, and libraries. Furthermore, the results look natural even when large sections of the image are completed.

4.8 Limitations and Discussion

Although our model can handle various images of any sizes with arbitrary holes, significantly large holes cannot be filled in due to the spatial support of the model as discussed in Section 3.2. By changing the model architecture to include more dilated convolutions it is possible to push this limit. Note that this limitation refers strictly to square masks, *e.g.*, wide areas can still be completed as long as they

Fig. 12. Additional image completion results by our approach using randomly generated masks.

are not too tall: information from above and below will be used to complete the image. This is especially limiting in the case of image extrapolation, in which the inpainting mask is at the border of the image. Figure 13-left shows such an example, which are from the [Hays and Efros 2007] dataset. Not only is the missing area very large relative to the image, but information from only one side of the area is available. Figure 13-right shows another failure case due to a large inpainting region. We note that in this case, [Hays and Efros 2007] also fails to realistically inpaint the mask. Approaches like [Hays and Efros 2007], which leverage sizable databases to copy and paste large parts of images, work well if the database contains an image significantly similar to the input. Indeed, for such approaches, extrapolation is easier than inpainting, since there are less to match at the boundary. Note that, in the output by [Hays and Efros 2007],

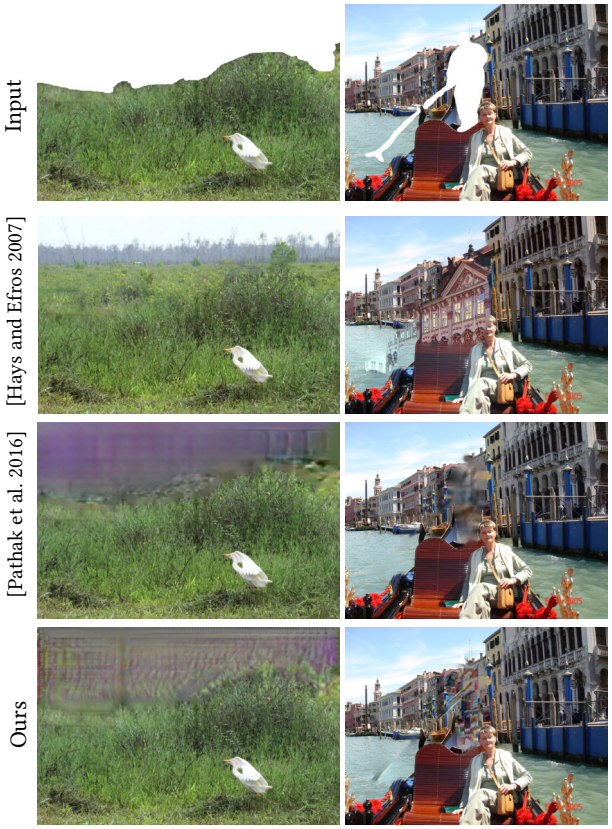


Fig. 13. Failure cases from the dataset of [Hays and Efros 2007]. For the comparison, we have retrained the model of [Pathak et al. 2016] on the Places2 dataset for arbitrary regions. The image on the left corresponds to a case of image extrapolation, *i.e.*, the inpainting mask lies on the boundary of the image. Out of the 51 images in this dataset, 32 have masks that correspond to image extrapolation.

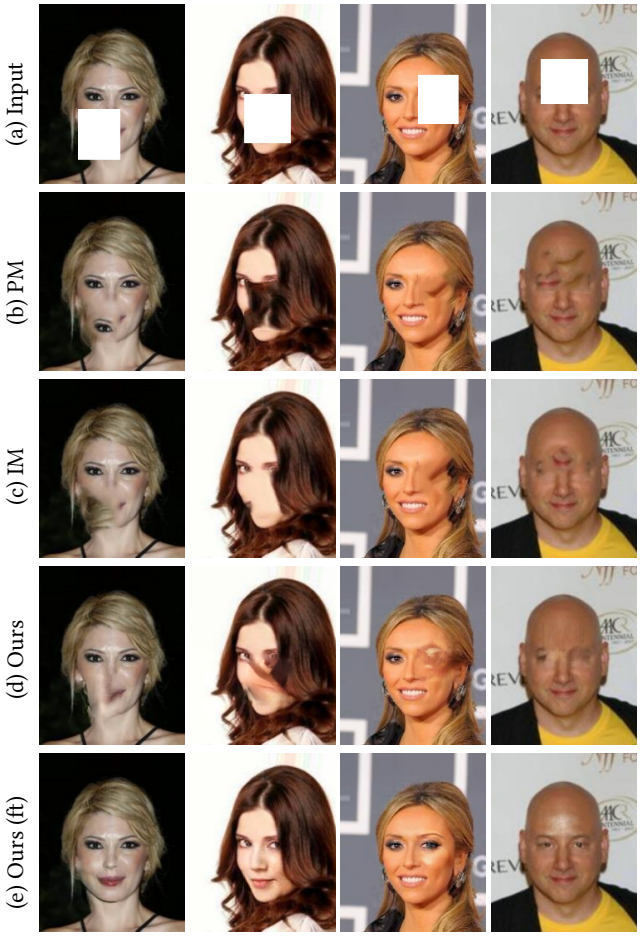


Fig. 14. Comparison with the PatchMatch (PM) and Image Melding (IM) on the CelebA dataset. We provide results for our general model (Ours), and our model fine-tuned for faces (Ours (ft)). Patch-based approaches are unable to generate novel objects in the scene leading to unnatural results.

parts of the original image outside of the mask are modified by fitting the image patch from the database.

The main advantage of our approach over standard techniques such as PatchMatch lies in the fact that our approach can generate novel objects that do not appear in the image. While this may not be necessary for certain outdoor scenes when parts of the image can be used for image completion, for other cases such as completing faces, it becomes critical, as without being able to generate noses, eyes, mouths, etc., the completion will fail as shown in Fig. 14.

Some examples of failure cases can be seen in Fig. 15. In general, the most common failure case is when a heavily structured object, *e.g.*, a person or an animal, is partially masked. In the left image, we can see that the model prioritizes reconstructing the mountains and trees in the background over the heads of the girls, leading to a poor completion. In the right image, our approach fails to complete the head of a stuffed monkey. We do not, however, that structured textures do get completed as shown in Fig. 12.

5 CONCLUSION

We have presented a novel approach for image completion that produces locally and globally consistent image completions based on convolutional neural networks. We have shown that, by using global and local context discriminators, it is possible to train models to produce realistic image completion. Unlike the patch-based approaches, our approach can generate novel objects that do not appear elsewhere in the image. We have provided in-depth comparisons with existing approaches and show realistic image completion for a large variety of scenes. Furthermore, we also use our approach to complete images of faces and show in a user study that our generated faces are indistinguishable from real faces 77% of the time.

REFERENCES

Coloma Ballester, Marcelo Bertalmío, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing* 10, 8 (2001), 1200–1211.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 28, 3 (2009), 24:1–24:11.

Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. 2010. The Generalized Patchmatch Correspondence Algorithm. In *European Conference on Computer Vision*. 29–43.

Fig. 15. Failure cases for our approach where our model is unable to complete heavily structured objects such as people.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image Inpainting. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*. 417–424.

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

A. Criminisi, P. Perez, and K. Toyama. 2004. Region Filling and Object Removal by Exemplar-based Image Inpainting. *IEEE Transactions on Image Processing* 13, 9 (2004), 1200–1212.

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. 2012. Image Melding: Combining Inconsistent Images using Patch-based Synthesis. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31, 4, Article 82 (2012), 82:1–82:10 pages.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Yue Deng, Qionghai Dai, and Zengke Zhang. 2011. Graph Laplace for occluded face completion and recognition. *IEEE Transactions on Image Processing* 20, 8 (2011), 2329–2338.

Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. 2003. Fragment-based Image Completion. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3 (2003), 303–312.

Alexei Efros and Thomas Leung. 1999. Texture Synthesis by Non-parametric Sampling. In *International Conference on Computer Vision*. 1033–1038.

Alexei A. Efros and William T. Freeman. 2001. Image Quilting for Texture Synthesis and Transfer. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*. 341–346.

Kunihiko Fukushima. 1988. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks* 1, 2 (1988), 119–130.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Conference on Neural Information Processing Systems*. 2672–2680.

James Hays and Alexei A. Efros. 2007. Scene Completion Using Millions of Photographs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 26, 3, Article 4 (2007).

Kaiming He and Jian Sun. 2012. Statistics of Patch Offsets for Image Completion. In *European Conference on Computer Vision*. 16–29.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image Completion Using Planar Structure Guidance. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4, Article 129 (2014), 10 pages.

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. (2017).

Jiaya Jia and Chi-Keung Tang. 2003. Image repairing: robust image synthesis by adaptive ND tensor voting. In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1. 643–650.

Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. 2014. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*.

Johannes Kopf, Wolf Kienzle, Steven Drucker, and Sing Bing Kang. 2012. Quality Prediction for Image Completion. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31, 6, Article 131 (2012), 8 pages.

Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. 2005. Texture Optimization for Example-based Synthesis. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24, 3 (July 2005), 795–802.

Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3 (July 2003), 277–286.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.

Anat Levin, Assaf Zomet, and Yair Weiss. 2003. Learning How to Inpaint from Global Image Statistics. In *International Conference on Computer Vision*. 305–312.

Rongjian Li, Wenlu Zhang, Heung-Il Suk, Li Wang, Jiang Li, Dinggang Shen, and Shuiwang Ji. 2014. Deep learning based imaging data completion for improved brain disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 305–312.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision*.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Umar Mohammed, Simon JD Prince, and Jan Kautz. 2009. Visio-lization: generating novel facial images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 28, 3 (2009), 57.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*. 807–814.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Darko Pavić, Volker Schönefeld, and Leif Kobbelt. 2006. Interactive image completion with perspective correction. *The Visual Computer* 22, 9 (2006), 671–681.

Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson Image Editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3 (July 2003), 313–318.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations*.

Radim Šára Radim Tyleček. 2013. Spatial Pattern Templates for Recognition of Objects with Regular Structure. In *German Conference on Pattern Recognition*. Saarbrucken, Germany.

Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. 2015. Shepard Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. 1986. Learning representations by back-propagating errors. In *Nature*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Conference on Neural Information Processing Systems*.

Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. 2008. Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–8.

Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. 2005. Image Completion with Structure Propagation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24, 3 (July 2005), 861–868. DOI: https://doi.org/10.1145/1073204.1073274

Alexandru Telea. 2004. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools* 9, 1 (2004), 23–34.

Yonatan Wexler, Eli Shechtman, and Michal Irani. 2007. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (2007), 463–476.

Oliver Whyte, Josef Sivic, and Andrew Zisserman. 2009. Get Out of my Picture! Internet-based Inpainting. In *British Machine Vision Conference*.

Junyuan Xie, Linli Xu, and Enhong Chen. 2012. Image Denoising and Inpainting with Deep Neural Networks. In *Conference on Neural Information Processing Systems*. 341–349.

Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. 2017. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations*.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012).

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Antonio Torralba, and Aude Oliva. 2016. Places: An Image Database for Deep Scene Understanding. *CoRR* abs/1610.02055 (2016).