Generative Image Inpainting with Contextual Attention

Jiahui Yu[1] Zhe Lin[2] Jimei Yang[2] Xiaohui Shen[2] Xin Lu[2] Thomas S. Huang[1]

[1]University of Illinois at Urbana-Champaign
[2]Adobe Research

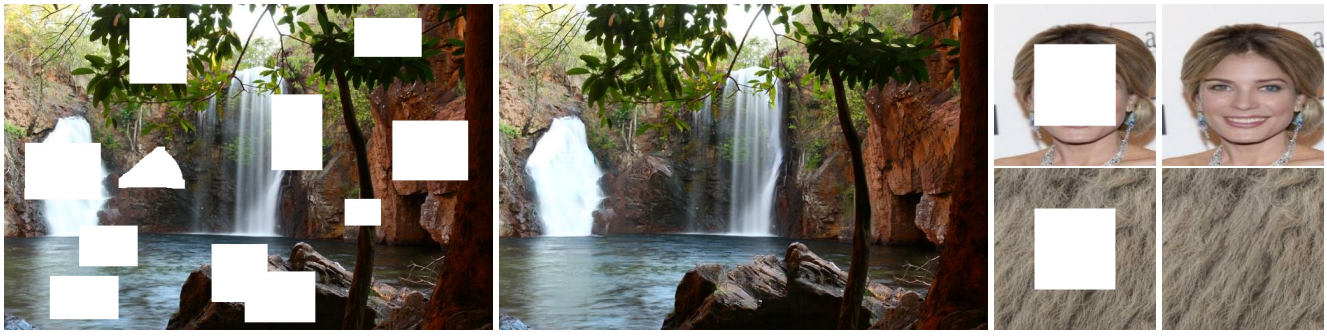

Figure 1: Example inpainting results of our method on images of natural scene, face and texture. Missing regions are shown in white. In each pair, the left is input image and right is the direct output of our trained generative neural networks without any post-processing.

Abstract

Recent deep learning based approaches have shown promising results for the challenging task of inpainting large missing regions in an image. These methods can generate visually plausible image structures and textures, but often create distorted structures or blurry textures inconsistent with surrounding areas. This is mainly due to ineffectiveness of convolutional neural networks in explicitly borrowing or copying information from distant spatial locations. On the other hand, traditional texture and patch synthesis approaches are particularly suitable when it needs to borrow textures from the surrounding regions. Motivated by these observations, we propose a new deep generative model-based approach which can not only synthesize novel image structures but also explicitly utilize surrounding image features as references during network training to make better predictions. The model is a feed-forward, fully convolutional neural network which can process images with multiple holes at arbitrary locations and with variable sizes during the test time. Experiments on multiple datasets including faces (CelebA, CelebA-HQ), textures (DTD) and natural images (ImageNet, Places2) demonstrate that our proposed approach generates higher-quality inpainting results than existing ones. Code, demo and models are available at: `https://github.com/JiahuiYu/generative_inpainting`.

1. Introduction

Filling missing pixels of an image, often referred as image inpainting or completion, is an important task in computer vision. It has many applications in photo editing, image-based rendering and computational photography [3, 25, 30, 31, 36, 41]. The core challenge of image inpainting lies in synthesizing visually realistic and semantically plausible pixels for the missing regions that are coherent with existing ones.

Early works [3, 14] attempted to solve the problem using ideas similar to texture synthesis [10, 11], i.e. by matching and copying background patches into holes starting from low-resolution to high-resolution or propagating from hole boundaries. These approaches work well especially in background inpainting tasks, and are widely deployed in practical applications [3]. However, as they assume missing patches can be found somewhere in background regions, they cannot hallucinate novel image contents for challenging cases where inpainting regions involve complex, non-repetitive structures (e.g. faces, objects). Moreover, these methods are not able to capture high-level semantics.

Rapid progress in deep convolutional neural networks (CNN) and generative adversarial networks (GAN) [12] inspired recent works [17, 27, 32, 41] to formulate inpainting as a conditional image generation problem where high-level recognition and low-level pixel synthesis are formulated into a convolutional encoder-decoder network,

jointly trained with adversarial networks to encourage the coherency between generated and existing pixels. These works are shown to generate plausible new contents in highly structured images, such as faces, objects and scenes.

Unfortunately, these CNN-based methods often create boundary artifacts, distorted structures and blurry textures inconsistent with surrounding areas. We found that this is likely due to ineffectiveness of convolutional neural networks in modeling long-term correlations between distant contextual information and the hole regions. For example, to allow a pixel being influenced by the content of 64 pixels away, it requires at least 6 layers of 3×3 convolutions with dilation factor 2 or equivalent [17, 42]. Nevertheless, a dilated convolution samples features from a regular and symmetric grid and thus may not be able to weigh the features of interest over the others. Note that a recent work [40] attempts to address the appearance discrepancy by optimizing texture similarities between generated patches and the matched patches in known regions. Although improving the visual quality, this method is being dragged by hundreds of gradient descent iterations and costs minutes to process an image with resolution 512×512 on GPUs.

We present a unified feed-forward generative network with a novel contextual attention layer for image inpainting. Our proposed network consists of two stages. The first stage is a simple dilated convolutional network trained with reconstruction loss to rough out the missing contents. The contextual attention is integrated in the second stage. The core idea of contextual attention is to use the features of known patches as convolutional filters to process the generated patches. It is designed and implemented with convolution for matching generated patches with known contextual patches, channel-wise softmax to weigh relevant patches and deconvolution to reconstruct the generated patches with contextual patches. The contextual attention module also has spatial propagation layer to encourage spatial coherency of attention. In order to allow the network to hallucinate novel contents, we have another convolutional pathway in parallel with the contextual attention pathway. The two pathways are aggregated and fed into single decoder to obtain the final output. The whole network is trained end to end with reconstruction losses and two Wasserstein GAN losses [1, 13], where one critic looks at the global image while the other looks at the local patch of the missing region.

Experiments on multiple datasets including faces, textures and natural images demonstrate that the proposed approach generates higher-quality inpainting results than existing ones. Example results are shown in Figure 1.

Our contributions are summarized as follows:

- We propose a novel contextual attention layer to explicitly attend on related feature patches at distant spatial locations.
- We introduce several techniques including inpainting network enhancements, global and local WGANs [13] and spatially discounted reconstruction loss to improve the training stability and speed based on the current the state-of-the-art generative image inpainting network [17]. As a result, we are able to train the network in a week instead of two months.
- Our unified feed-forward generative network achieves high-quality inpainting results on a variety of challenging datasets including CelebA faces [28], CelebA-HQ faces [22], DTD textures [6], ImageNet [34] and Places2 [43].

2. Related Work

2.1. Image Inpainting

Existing works for image inpainting can be mainly divided into two groups. The first group represents traditional diffusion-based or patch-based methods with low-level features. The second group attempts to solve the inpainting problem by a learning-based approach, e.g. training deep convolutional neural networks to predict pixels for the missing regions.

Traditional diffusion or patch-based approaches such as [2, 4, 10, 11] typically use variational algorithms or patch similarity to propagate information from the background regions to the holes. These methods work well for stationary textures but are limited for non-stationary data such as natural images. Simakov et al. [36] propose a bidirectional patch similarity-based scheme to better model non-stationary visual data for re-targeting and inpainting applications. However, dense computation of patch similarity [36] is a very expensive operation, which prohibits practical applications of such method. In order to address the challenge, a fast nearest neighbor field algorithm called PatchMatch [3] has been proposed which has shown significant practical values for image editing applications including inpainting.

Recently, deep learning and GAN-based approaches have emerged as a promising paradigm for image inpainting. Initial efforts [23, 39] train convolutional neural networks for denoising and inpainting of small regions. Context Encoders [32] firstly train deep neural networks for inpainting large holes. It is trained to complete center region of 64×64 in a 128×128 image, with both ℓ_2 pixel-wise reconstruction loss and generative adversarial loss as the objective function. More recently, Iizuka et al. [17] improve it by introducing both global and local discriminators as adversarial losses. The global discriminator assesses if completed image is coherent as a whole, while the local discriminator focus on a small area centered at the generated region to enforce the local consistency. In addition, Iizuka et al. [17] use dilated convolutions in inpainting network to

replace channel-wise fully connected layer adopted in Context Encoders, both techinics are proposed for increasing receptive fields of output neurons. Meanwhile, there have been several studies focusing on generative face inpainting. Yeh et al. [41] search for the closest encoding in latent space of the corrupted image and decode to get completed image. Li et al. [27] introduce additional face parsing loss for face completion. However, these methods typically require post processing steps such as image blending operation to enforce color coherency near the hole boundaries.

Several works [37, 40] follow ideas from image stylization [5, 26] to formulate the inpainting as an optimization problem. For example, Yang et al. [40] propose a multi-scale neural patch synthesis approach based on joint optimization of image content and texture constraints, which not only preserves contextual structures but also produces high-frequency details by matching and adapting patches with the most similar mid-layer feature correlations of a deep classification network. This approach shows promising visual results but is very slow due to the optimization process.

2.2. Attention Modeling

There have been many studies on learning spatial attention in deep convolutional neural networks. Here, we select to review a few representative ones related to the proposed contextual attention model. Jaderberg et al. [19] firstly propose a parametric spatial attention module called spatial transformer network (STN) for object classification tasks. The model has a localization module to predict parameters of global affine transformation to warp features. However, this model assumes a global transformation so is not suitable for modeling patch-wise attention. Zhou et al. [44] introduce an appearance flow to predict offset vectors specifying which pixels in the input view should be moved to reconstruct the target view for novel view synthesis. This method is shown to be effective for matching related views of the same objects but is not effective in predicting a flow field from the background region to the hole, according to our experiments. Recently, Dai et al. [8] and Jeon et al. [20] propose to learn spatially attentive or active convolutional kernels. These methods can potentially better leverage information to deform the convolutional kernel shape during training but may still be limited when we need to borrow exact features from the background.

3. Improved Generative Inpainting Network

We first construct our baseline generative image inpainting network by reproducing and making several improvements to the recent state-of-the-art inpainting model [17] which has shown promising visual results for inpainting images of faces, building facades and natural images.

Coarse-to-fine network architecture The network architecture of our improved model is shown in Figure 2. We follow the same input and output configurations as in [17] for training and inference, i.e. the generator network takes an image with white pixels filled in the holes and a binary mask indicating the hole regions as input pairs, and outputs the final completed image. We pair the input with a corresponding binary mask to handle holes with variable sizes, shapes and locations. The input to the network is a 256×256 image with a rectangle missing region sampled randomly during training, and the trained model can take an image of different sizes with multiple holes in it.

In image inpainting tasks, the size of the receptive fields should be sufficiently large, and Iizuka et al. [17] adopt dilated convolution for that purpose. To further enlarge the receptive fields and stabilize training, we introduce a two-stage coarse-to-fine network architecture where the first network makes an initial coarse prediction, and the second network takes the coarse prediction as inputs and predict refined results. The coarse network is trained with the reconstruction loss explicitly, while the refinement network is trained with the reconstruction as well as GAN losses. Intuitively, the refinement network sees a more complete scene than the original image with missing regions, so its encoder can learn better feature representation than the coarse network. This two-stage network architecture is similar in spirits to residual learning [15] or deep supervision [24].

Also, our inpainting network is designed in a thin and deep scheme for efficiency purpose and has fewer parameters than the one in [17]. In terms of layer implementations, we use mirror padding for all convolution layers and remove batch normalization layers [18] (which we found deteriorates color coherence). Also, we use ELUs [7] as activation functions instead of ReLU in [17], and clip the output filter values instead of using $tanh$ or $sigmoid$ functions. In addition, we found separating global and local feature representations for GAN training works better than feature concatenation in [17]. More details can be found in the supplementary materials.

Global and local Wasserstein GANs Different from previous generative inpainting networks [17, 27, 32] which rely on DCGAN [33] for adversarial supervision, we propose to use a modified version of WGAN-GP [1, 13]. We attach the WGAN-GP loss to both global and local outputs of the second-stage refinement network to enforce global and local consistency, inspired by [17]. WGAN-GP loss is well-known to outperform existing GAN losses for image generation tasks, and it works well when combined with ℓ_1 reconstruction loss as they both use the ℓ_1 distance metric.

Specifically, WGAN uses the *Earth-Mover* distance (a.k.a. *Wasserstein-1*) distance $W(\mathbb{P}_r, \mathbb{P}_g)$ for comparing the generated and real data distributions. Its objective function is constructed by applying the *Kantorovich-Rubinstein*

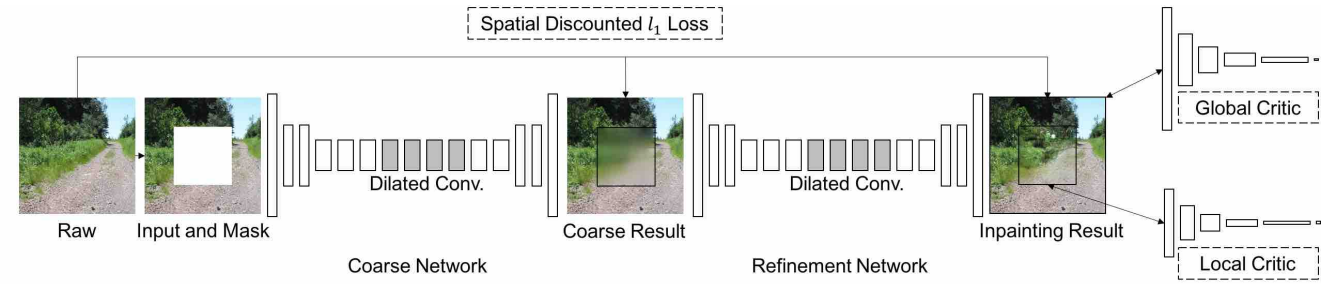

Figure 2: Overview of our improved generative inpainting framework. The coarse network is trained with reconstruction loss explicitly, while the refinement network is trained with reconstruction loss, global and local WGAN-GP adversarial loss.

duality:

$$\min_G \max_{D \in \mathcal{D}} E_{\mathbf{x} \sim \mathbb{P}_r}[D(\mathbf{x})] - E_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}})],$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions and $\mathbb{P}_g$ is the model distribution implicitly defined by $\tilde{\mathbf{x}} = G(\mathbf{z})$. $\mathbf{z}$ is the input to the generator.

Gulrajani et al. [13] proposed an improved version of WGAN with a gradient penalty term

$$\lambda E_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}}(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2,$$

where $\hat{\mathbf{x}}$ is sampled from the straight line between points sampled from distribution $\mathbb{P}_g$ and $\mathbb{P}_r$. The reason is that the gradient of D^* at all points $\hat{\mathbf{x}} = (1-t)\mathbf{x} + t\tilde{\mathbf{x}}$ on the straight line should point directly towards current sample $\tilde{\mathbf{x}}$, meaning $\nabla_{\hat{\mathbf{x}}} D^*(\hat{\mathbf{x}}) = \frac{\tilde{\mathbf{x}} - \hat{\mathbf{x}}}{\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|}$.

For image inpainting, we only try to predict hole regions, thus the gradient penalty should be applied only to pixels inside the holes. This can be implemented with multiplication of gradients and input mask $\mathbf{m}$ as follows:

$$\lambda E_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}}(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}) \odot (\mathbf{1} - \mathbf{m})\|_2 - 1)^2,$$

where the mask value is 0 for missing pixels and 1 for elsewhere. λ is set to 10 in all experiments.

We use a weighted sum of pixel-wise ℓ_1 loss (instead of mean-square-error as in [17]) and WGAN adversarial losses. Note that in primal space, *Wasserstein-1* distance in WGAN is based on ℓ_1 ground distance:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} E_{(\mathbf{x},\mathbf{y}) \sim \gamma}[\|\mathbf{x} - \mathbf{y}\|],$$

where $\prod(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. Intuitively, the pixel-wise reconstruction loss directly regresses holes to the current ground truth image, while WGANs implicitly learn to match potentially correct images and train the generator with adversarial gradients. As both losses measure pixel-wise ℓ_1 distances, the combined loss is easier to train and makes the optimization process stabler.

Spatially discounted reconstruction loss Inpainting problems involve hallucination of pixels, so it could have many plausible solutions for any given context. In challenging cases, a plausible completed image can have patches or pixels that are very different from those in the original image. As we use the original image as the only ground truth to compute a reconstruction loss, strong enforcement of reconstruction loss in those pixels may mislead the training process of convolutional network.

Intuitively, missing pixels near the hole boundaries have much less ambiguity than those pixels closer to the center of the hole. This is similar to the issue observed in reinforcement learning. When long-term rewards have large variations during sampling, people use temporal discounted rewards over sampled trajectories [38]. Inspired by this, we introduce spatially discounted reconstruction loss using a weight mask $\mathbf{M}$. The weight of each pixel in the mask is computed as γ^l, where l is the distance of the pixel to the nearest known pixel. γ is set to 0.99 in all experiments.

Similar weighting ideas are also explored in [32, 41]. Importance weighted context loss, proposed in [41], is spatially weighted by the ratio of uncorrupted pixels within a fixed window (e.g. 7×7). Pathak et al. [32] predict a slightly larger patch with higher loss weighting (×10) in the border area. For inpainting large hole, the proposed discounted loss is more effective for improving the visual quality. We use discounted ℓ_1 reconstruction loss in our implementation.

With all the above improvements, our baseline generative inpainting model converges much faster than [17] and result in more accurate inpainting results. For Places2 [43], we reduce the training time from 11,520 GPU-hours (K80) reported by [17] to 120 GPU-hours (GTX 1080) which is almost 100× speedup. Moreover, the post-processing step (image blending) [17] is no longer necessary.

4. Image Inpainting with Contextual Attention

Convolutional neural networks process image features with local convolutional kernel layer by layer thus are not

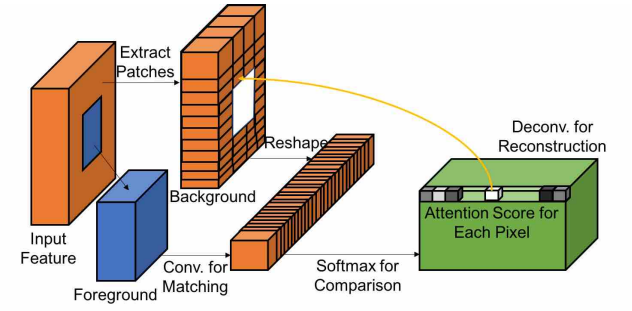

Figure 3: Illustration of the contextual attention layer. Firstly we use convolution to compute matching score of foreground patches with background patches (as convolutional filters). Then we apply softmax to compare and get attention score for each pixel. Finally we reconstruct foreground patches with background patches by performing deconvolution on attention score. The contextual attention layer is differentiable and fully-convolutional.

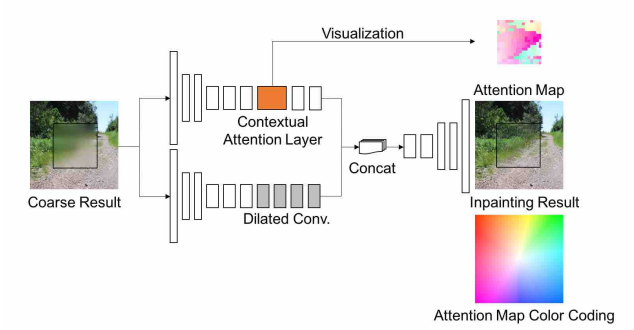

Figure 4: Based on coarse result from the first encoder-decoder network, two parallel encoders are introduced and then merged to single decoder to get inpainting result. For visualization of attention map, color indicates relative location of the most interested background patch for each pixel in foreground. For examples, white (center of color coding map) means the pixel attends on itself, pink on bottom-left, green means on top-right.

effective for borrowing features from distant spatial locations. To overcome the limitation, we consider attention mechanism and introduce a novel contextual attention layer in the deep generative network. In this section, we first discuss details of the contextual attention layer, and then address how we integrate it into our unified inpainting network.

4.1. Contextual Attention

The contextual attention layer learns where to borrow or copy feature information from known background patches to generate missing patches. It is differentiable, thus can be trained in deep models, and fully-convolutional, which allows testing on arbitrary resolutions.

Match and attend We consider the problem where we want to match features of missing pixels (foreground) to surroundings (background). As shown in Figure 3, we first extract patches (3×3) in background and reshape them as convolutional filters. To match foreground patches $\{f_{x,y}\}$ with backgrounds ones $\{b_{x',y'}\}$, we measure with normalized inner product (cosine similarity)

$$s_{x,y,x',y'} = \langle \frac{f_{x,y}}{||f_{x,y}||}, \frac{b_{x',y'}}{||b_{x',y'}||} \rangle,$$

where $s_{x,y,x',y'}$ represents similarity of patch centered in background (x', y') and foreground (x, y). Then we weigh the similarity with scaled softmax along $x'y'$-dimension to get attention score for each pixel $s^*_{x,y,x',y'} = \mathit{softmax}_{x',y'}(\lambda s_{x,y,x',y'})$, where λ is a constant value. This is efficiently implemented as convolution and channel-wise softmax. Finally, we reuse extracted patches $\{b_{x',y'}\}$ as deconvolutional filters to reconstruct foregrounds. Values of overlapped pixels are averaged.

Attention propagation We further encourage coherency of attention by propagation (fusion). The idea of coherency is that a shift in foreground patch is likely corresponding to an equal shift in background patch for attention. For example, $s^*_{x,y,x',y'}$ usually have close value with $s^*_{x+1,y,x'+1,y'}$. To model and encourage coherency of attention maps, we do a left-right propagation followed by a top-down propagation with kernel size of k. Take left-right propagation as an example, we get new attention score with:

$$\hat{s}_{x,y,x',y'} = \sum_{i \in \{-k,\dots,k\}} s^*_{x+i,y,x'+i,y'}.$$

The propagation is efficiently implemented as convolution with identity matrix as kernels. Attention propagation significantly improves inpainting results in testing and enriches gradients in training.

Memory efficiency Assuming that a 64×64 region is missing in a 128×128 feature map, then the number of convolutional filters extracted from backgrounds is 12,288. This may cause memory overhead for GPUs. To overcome this issue, we introduce two options: 1) extracting background patches with strides to reduce the number of filters and 2) downscaling resolution of foreground inputs before convolution and upscaling attention map after propagation.

4.2. Unified Inpainting Network

To integrate attention module, we introduce two parallel encoders as shown in Figure 4 based on Figure 2. The bottom encoder specifically focuses on hallucinating contents with layer-by-layer (dilated) convolution, while the top one tries to attend on background features of interest. Output features from two encoders are aggregated and fed into a

single decoder to obtain the final output. To interpret contextual attention, we visualize it in a way shown in Figure 4. We use color to indicate the relative location of the most interested background patch for each foreground pixel. For examples, white (center of color coding map) means the pixel attends on itself, pink on bottom-left, green on top-right. The offset value is scaled differently for different images to best visualize the most interesting range.

For training, given a raw image $\mathbf{x}$, we sample a binary image mask $\mathbf{m}$ at a random location. Input image $\mathbf{z}$ is corrupted from the raw image as $\mathbf{z} = \mathbf{x} \odot \mathbf{m}$. Inpainting network G takes concatenation of $\mathbf{z}$ and $\mathbf{m}$ as input, and output predicted image $\mathbf{x}' = G(\mathbf{z}, \mathbf{m})$ with the same size as input. Pasting the masked region of $\mathbf{x}'$ to input image, we get the inpainting output $\tilde{\mathbf{x}} = \mathbf{z} + \mathbf{x}' \odot (\mathbf{1} - \mathbf{m})$. Image values of input and output are linearly scaled to $[-1, 1]$ in all experiments. Training procedure is shown in Algorithm 1.

Algorithm 1 Training of our proposed framework.

while G has not converged **do**
  for $i = 1, ..., 5$ **do**
    Sample batch images $\mathbf{x}$ from training data;
    Generate random masks $\mathbf{m}$ for $\mathbf{x}$;
    Construct inputs $\mathbf{z} \leftarrow \mathbf{x} \odot \mathbf{m}$;
    Get predictions $\tilde{\mathbf{x}} \leftarrow \mathbf{z} + G(\mathbf{z}, \mathbf{m}) \odot (\mathbf{1} - \mathbf{m})$;
    Sample $t \sim U[0, 1]$ and $\hat{\mathbf{x}} \leftarrow (1 - t)\mathbf{x} + t\tilde{\mathbf{x}}$;
    Update two critics with $\mathbf{x}$, $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$;
  end for
  Sample batch images $\mathbf{x}$ from training data;
  Generate random masks $\mathbf{m}$ for $\mathbf{x}$;
  Update inpainting network G with spatial discounted ℓ_1 loss and two adversarial critic losses;
end while

5. Experiments

We evaluate the proposed inpainting model on four datasets including Places2 [43], CelebA faces [28], CelebA-HQ faces [22], DTD textures [6] and ImageNet [34].

Qualitative comparisons First, we show in Figure 5 that our baseline model generates comparable inpainting results with the previous state-of-the-art [17] by comparing our output result and result copied from their main paper. Note that no post-processing step is performed for our baseline model, while image blending is applied in result of [17].

Next we use the most challenging Places2 dataset to evaluate our full model with contextual attention by comparing to our baseline two-stage model which is extended from the previous state-of-the-art [17]. For training, we use images of resolution 256×256 with largest hole size 128×128 described in Section 4.2. Both methods are based on fully-convolutional neural networks thus can fill in multiple holes on images of different resolutions. Visual comparisons on a variety of complex scenes from the validation set are shown in Figure 6. Those test images are all with size 512×680 for consistency of testing. All the results reported are direct outputs from the trained models without using any post-processing. For each example, we also visualize latent attention map for our model in the last column (color coding is explained in Section 4.2).

As shown in the figure, our full model with contextual attention can leverage the surrounding textures and structures and consequently generates more realistic results with much less artifacts than the baseline model. Visualizations of attention maps reveal that our method is aware of contextual image structures and can adaptively borrow information from surrounding areas to help the synthesis and generation.

In Figure 7, we also show some example results and attention maps of our full model trained on CelebA, DTD and ImageNet. Due to space limitation, we include more results for these datasets in the supplementary material.

Quantitative comparisons Like other image generation tasks, image inpainting lacks good quantitative evaluation metrics. Inception score [35] introduced for evaluating GAN models is not a good metric for evaluating image inpainting methods as inpainting mostly focuses on background filling (e.g. object removal case), not on its ability to generate a variety classes of objects.

Evaluation metrics in terms of reconstruction errors are also not perfect as there are many possible solutions different from the original image content. Nevertheless, we report our evaluation in terms of mean ℓ_1 error, mean ℓ_2 error, peak signal-to-noise ratio (PSNR) and total variation (TV) loss on validation set on Places2 just for reference in Table 1. As shown in the table, learning-based methods perform better in terms of ℓ_1, ℓ_2 errors and PSNR, while methods directly copying raw image patches have lower total variation loss.

Method	ℓ_1 loss	ℓ_2 loss	PSNR	TV loss
PatchMatch [3]	16.1%	3.9%	16.62	**25.0%**
Baseline model	9.4%	2.4%	18.15	25.7%
Our method	**8.6%**	**2.1%**	**18.91**	25.3%

Table 1: Results of mean ℓ_1 eror, mean ℓ_2 error, PSNR and TV loss on validation set on Places2 for reference.

Our full model has a total of **2.9M** parameters, which is roughly half of model proposed in [17]. Models are implemented on TensorFlow v1.3, CUDNN v6.0, CUDA v8.0, and run on hardware with CPU Intel(R) Xeon(R) CPU E5-2697 v3 (2.60GHz) and GPU GTX 1080 Ti. Our full model runs at **0.2** seconds per frame on GPU and **1.5** seconds per frame on CPU for images of resolution $\mathbf{512 \times 512}$ on average.

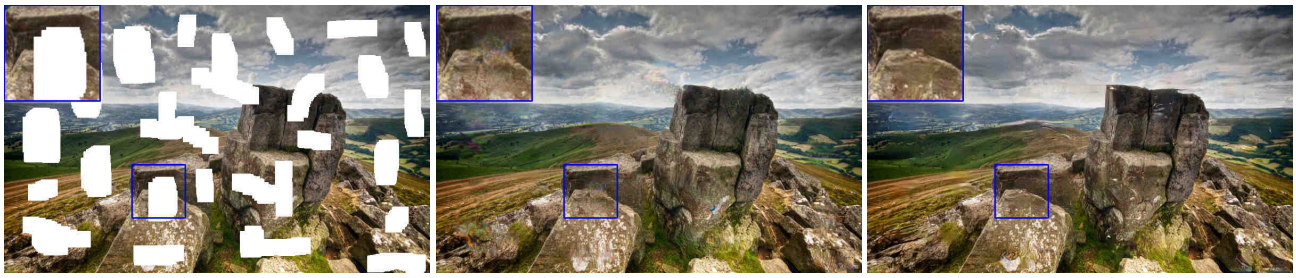

Figure 5: Comparison of our baseline model with Iizuka et al. [17]. From left to right, we show the input image, result copied from main paper of work [17], and result of our baseline model. Note that no post-processing step is performed for our baseline model, while image blending is applied for the result of [17]. Best viewed with zoom-in.

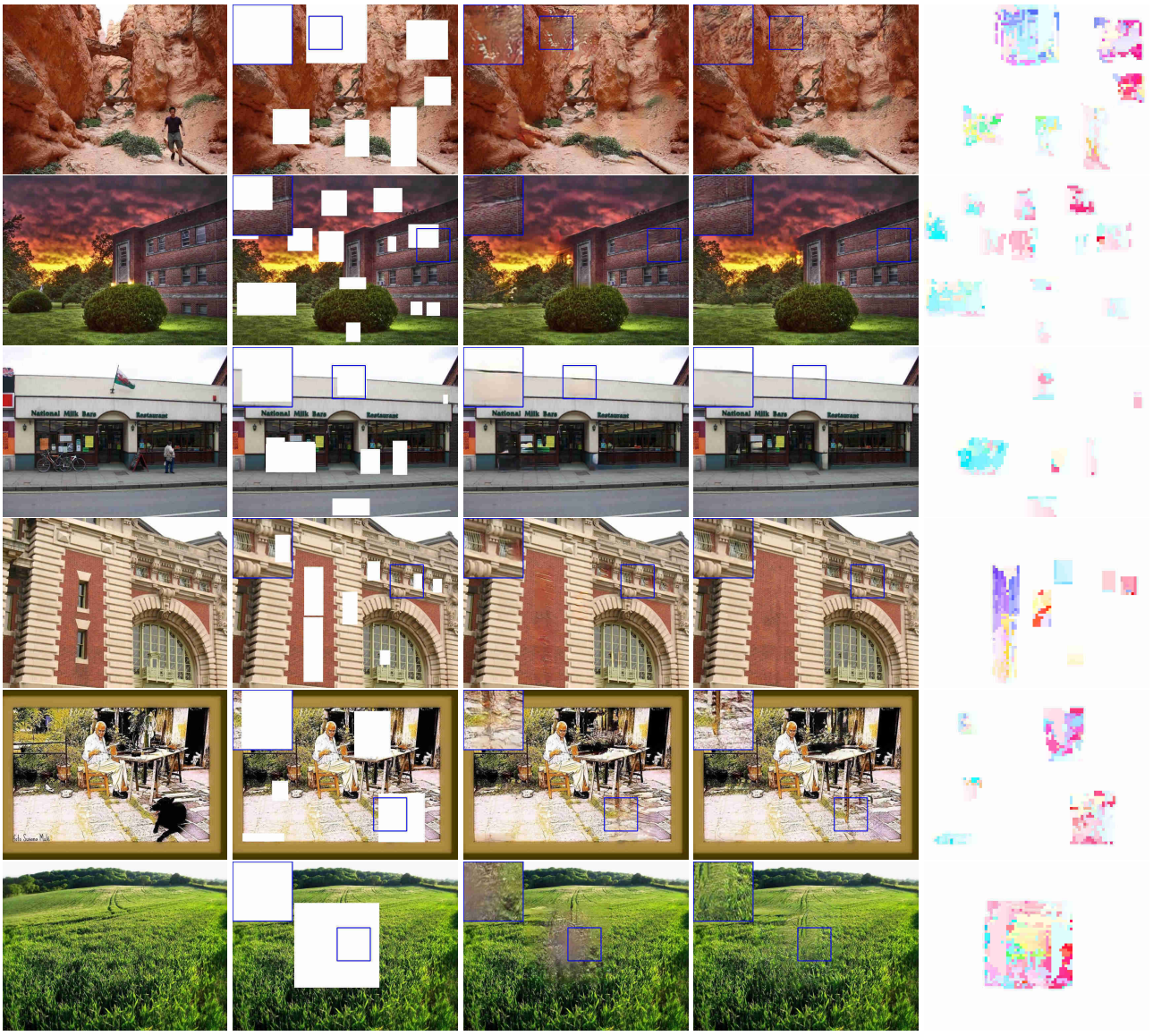

Figure 6: Qualitative results and comparisons to the baseline model. We show from left to right the original image, input image, result of our baseline model, result and attention map (upscaled $4\times$) of our full model. Best viewed with zoom-in.

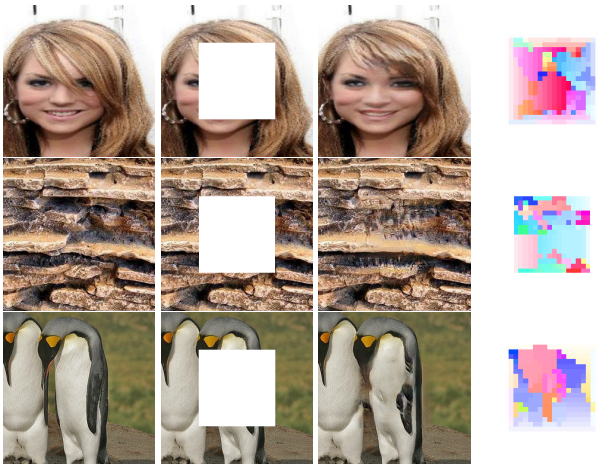

Figure 7: Sample results of our model on CelebA faces, DTD textures and ImageNet from top to bottom. Each row, from left to right, shows original image, input image, result and attention map (upscaled $4\times$), respectively.

5.1. Ablation study

Figure 8: We show input image, result and attention map using three different attention modules: spatial transformer network (left), appearance flow (middle), our contextual attention (right).

Contextual attention *vs*. spatial transformer network and appearance flow We investigate the effectiveness of contextual attention comparing to other spatial attention modules including appearance flow [44] and spatial transformer network [19] for image inpainting. For appearance flow [44], we train on the same framework except that the contextual attention layer is replaced with a convolution layer to directly predict 2-D pixel offsets as attention. As shown in Figure 8, for a very different test image pair, appearance flow returns very similar attention maps, meaning that the network may stuck in a bad local minima. To improve results of appearance flow, we also investigated ideas of multiple attention aggregation and patch-based attention. None of these ideas work well enough to improve the inpainting results. Also, we show the results with the spatial transformer network [19] as attention in our framework in Figure 8. As shown in the figure, STN-based attention does not work well for inpainting as its global affine transformation is too coarse.

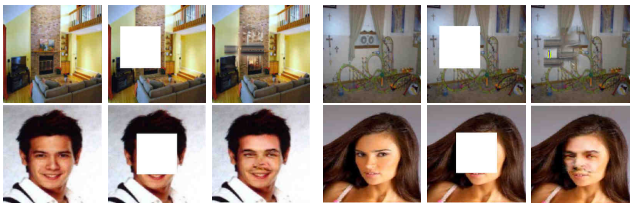

Figure 9: Inpainting results of the model trained with DCGAN on Places2 (top) and CelebA (bottom) when modes collapse.

Choice of the GAN loss for image inpainting Our inpainting framework benefits greatly from the WGAN-GP loss as validated by its learning curves and faster/stabler convergence behaviors. The same model trained with DCGAN sometimes collapses to limited modes for the inpainting task, as shown in Figure 9. We also experimented with LSGAN [29], and the results were worse.

Essential reconstruction loss We also performed testing if we could drop out the ℓ_1 reconstruction loss and purely rely on the adversarial loss (i.e. improved WGANs) to generate good results. To draw a conclusion, we train our inpainting model without ℓ_1 reconstruction loss in the refinement network. Our conclusion is that the pixel-wise reconstruction loss, although tends to make the result blurry, is an essential ingredient for image inpainting. The reconstruction loss is helpful in capturing content structures and serves as a powerful regularization term for training GANs.

Perceptual loss, style loss and total variation loss We have not found perceptual loss (reconstruction loss on VGG features), style loss (squared Frobenius norm of Gram matrix computed on the VGG features) [21] and total variation (TV) loss bring noticeable improvements for image inpainting in our framework, thus are not used.

6. Conclusion

We proposed a coarse-to-fine generative image inpainting framework and introduced our baseline model as well as full model with a novel contextual attention module. We showed that the contextual attention module significantly improves image inpainting results by learning feature representations for explicitly matching and attending to relevant background patches. As a future work, we plan to extend the method to very high-resolution inpainting applications using ideas similar to progressive growing of GANs [22]. The proposed inpainting framework and contextual attention module can also be applied on conditional image generation, image editing and computational photography tasks including image-based rendering, image super-resolution, guided editing and many others.

References

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

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2009)*, 2009.

[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.

[5] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016.

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017.

[9] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012)*, 2012.

[10] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[11] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[12] 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.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[14] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*. ACM, 2007.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Image completion using planar structure guidance. *ACM Transactions on Graphics (TOG)*, 33(4):129, 2014.

[17] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[18] 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.

[19] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[20] Y. Jeon and J. Kim. Active convolution: Learning the shape of convolution for image classification. *arXiv preprint arXiv:1703.09076*, 2017.

[21] 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. Springer, 2016.

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[23] 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.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

[25] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. *Computer Vision-ECCV 2004*, pages 377–389, 2004.

[26] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

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

[28] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[29] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[30] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[31] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. *arXiv preprint arXiv:1703.02921*, 2017.

[32] 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.

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

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[35] 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.

[36] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[37] X. Snelgrove. High-resolution multi-scale neural texture synthesis. In *SIGGRAPH ASIA 2017 Technical Briefs*. ACM, 2017.

[38] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[39] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014.

[40] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. *arXiv preprint arXiv:1611.09969*, 2016.

[41] 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.

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

[43] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[44] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016.

A. More Results on CelebA, CelebA-HQ, DTD and ImageNet

CelebA-HQ [22] We show results from our full model trained on CelebA-HQ dataset in Figure 10. Note that the original image resolution of CelebA-HQ dataset is 1024×1024. We resize image to 256×256 for both training and evaluation.

CelebA [28] We show more results from our full model trained on CelebA dataset in Figure 11. Note that the original image resolution of CelebA dataset is 218×178. We resize image to 315×256 and do a random crop of size 256×256 to make face landmarks roughly unaligned for both training and evaluation.

ImageNet [34] We show more results from our full model trained on ImageNet dataset in Figure 12.

DTD textures [6] We show more results from our full model trained on DTD dataset in Figure 13.

B. Comparisons with More Methods

We show more results for qualitative comparisons with more methods including Photoshop Content-Aware Fill [3], Image Melding [9] and StructCompletion [16] in Figure 14 and 15. For all these methods, we use default hyper-parameter settings.

C. More Visualization with Case Study

In addition to attention map visualization, we visualize which parts in the input image are being attended for pixels in holes. To do so, we highlight the regions that have the maximum attention score and overlay them to input image. As shown in Figure 16, the visualization results given holes in different locations demonstrate the effectiveness of our proposed contextual attention to borrow information at distant spatial locations.

D. Network Architectures

In addition to Section 3, we report more details of our network architectures. For simplicity, we denote them with K (kernel size), D (dilation), S (stride size) and C (channel number).

Inpainting network Inpainting network has two encoder-decoder architecture stacked together, with each encoder-decoder of network architecture:

K5S1C32 - K3S2C64 - K3S1C64 - K3S2C128 - K3S1C128 - K3S1C128 - K3D2S1C128 - K3D4S1C128 - K3D8S1C128 - K3D16S1C128 - K3S1C128 - K3S1C128 - resize ($2\times$) - K3S1C64 - K3S1C64 - resize ($2\times$) - K3S1C32 - K3S1C16 - K3S1C3 - clip.

Local WGAN-GP critic We use Leaky ReLU with $\alpha = 0.2$ as activation function for WGAN-GP critics.

K5S2C64 - K5S2C128 - K5S2C256 - K5S2C512 - fully-connected to 1.

Global WGAN-GP critic K5S2C64 - K5S2C128 - K5S2C256 - K5S2C256 - fully-connected to 1.

Contextual attention branch K5S1C32 - K3S2C64 - K3S1C64 - K3S2C128 - K3S1C128 - K3S1C128 - contextual attention layer - K3S1C128 - K3S1C128 - concat.

Figure 10: More inpainting results of our full model with contextual attention on CelebA-HQ faces. Each triad, from left to right, shows original image, input masked image and result image. All input images are masked from validation set (training and validation split is provided in released code). All results are direct outputs from same trained model without post-processing.

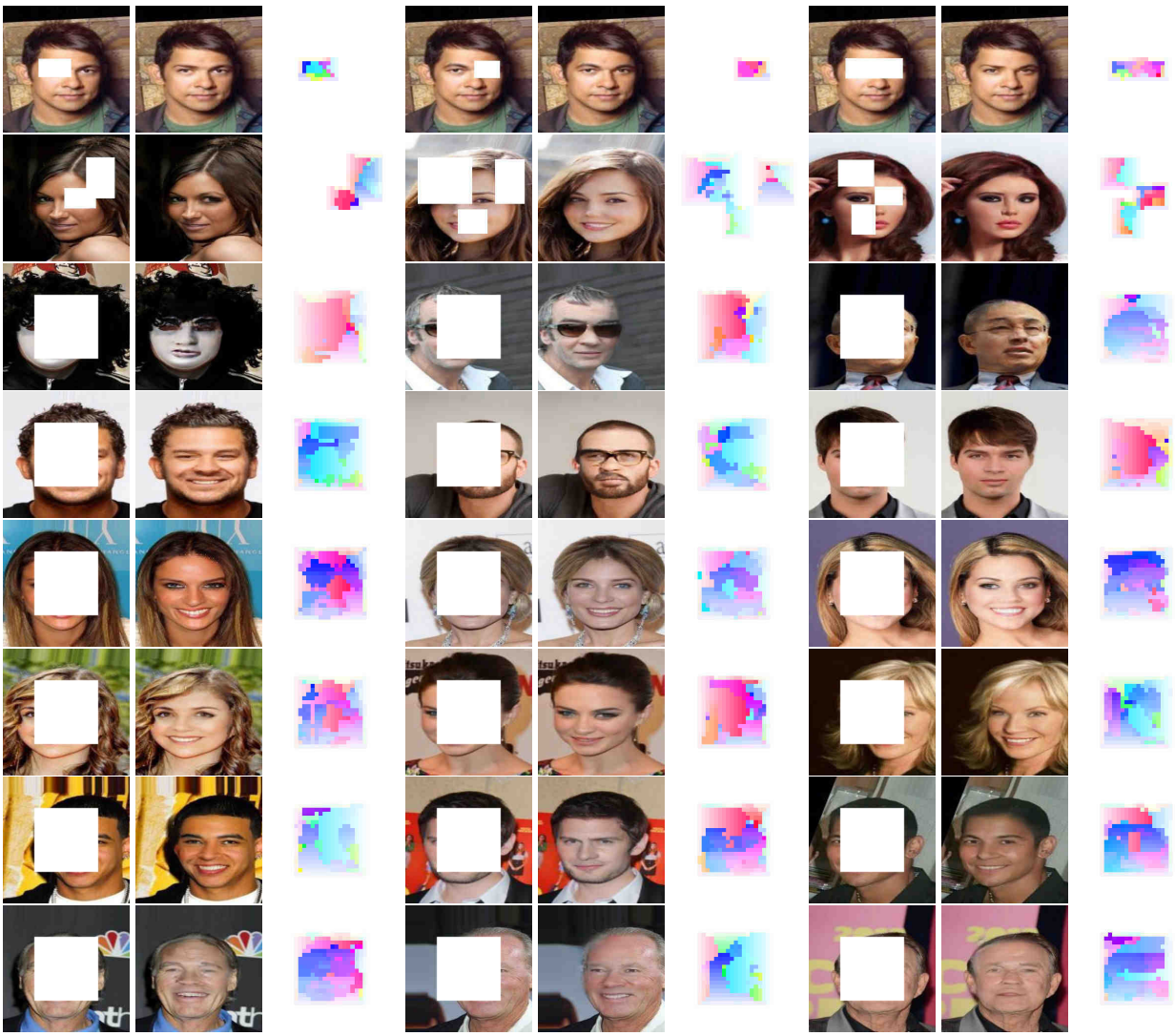

Figure 11: More inpainting results of our full model with contextual attention on CelebA faces. Each triad, from left to right, shows input image, result and attention map (upscaled $4\times$). All input images are masked from validation set (face identities are NOT overlapped between training set and validation set). All results are direct outputs from same trained model without post-processing.

Figure 12: More inpainting results of our full model with contextual attention on ImageNet. Each triad, from left to right, shows input image, result and attention map (upscaled $4\times$). All input images are masked from validation set. All results are direct outputs from same trained model without post-processing.

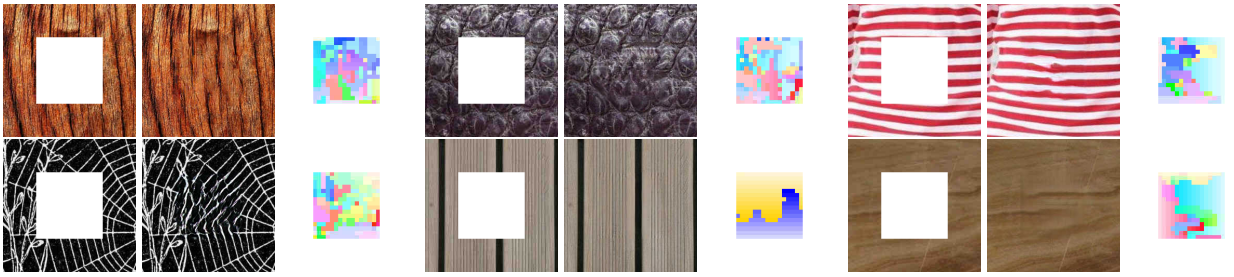

Figure 13: More inpainting results of our full model with contextual attention on DTD textures. Each triad, from left to right, shows input image, result and attention map (upscaled $4\times$). All input images are masked from validation set. All results are direct outputs from same trained model without post-processing.

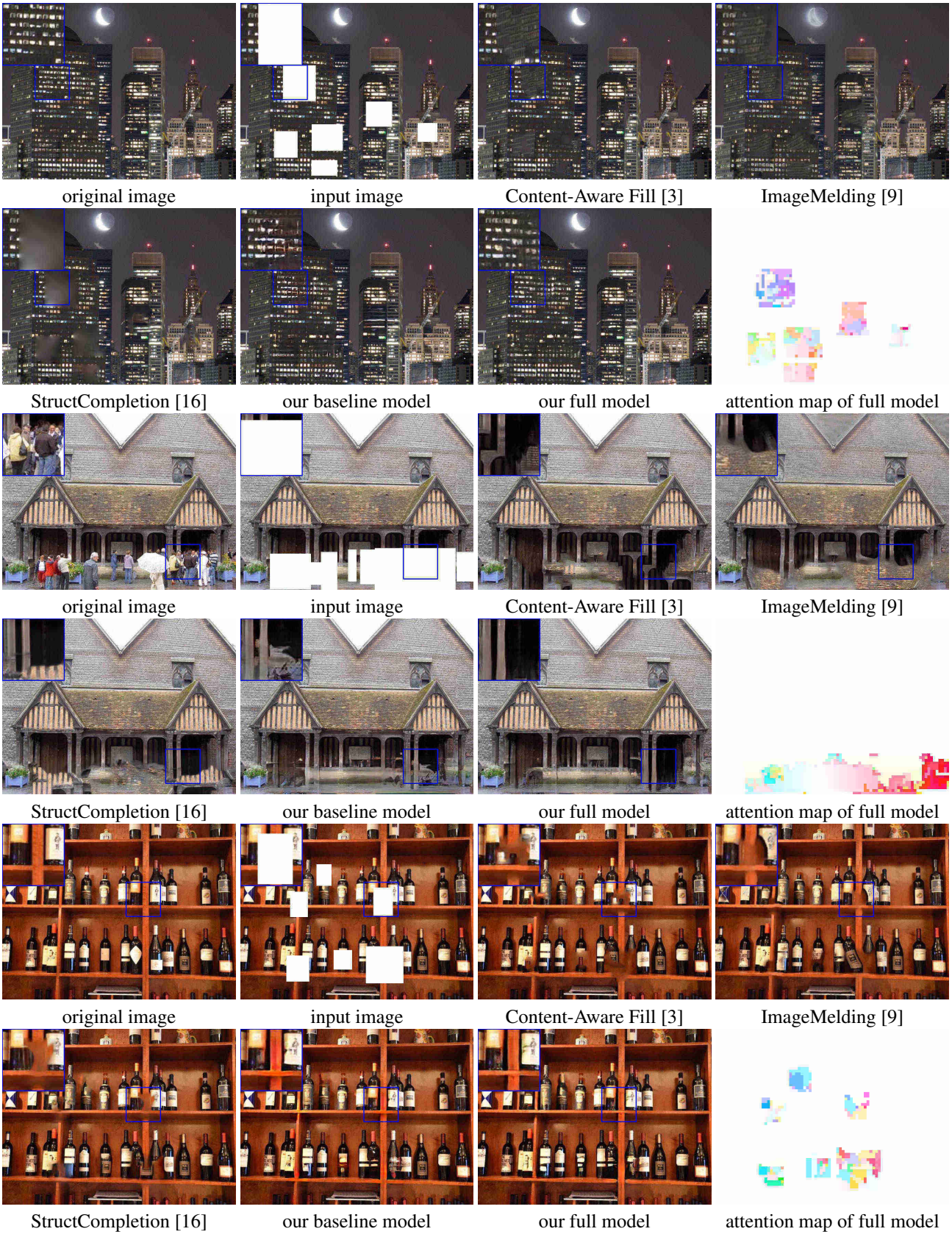

Figure 14: More qualitative results and comparisons. All input images are masked from validation set. All our results are direct outputs from same trained model without post-processing. Best viewed with zoom-in.

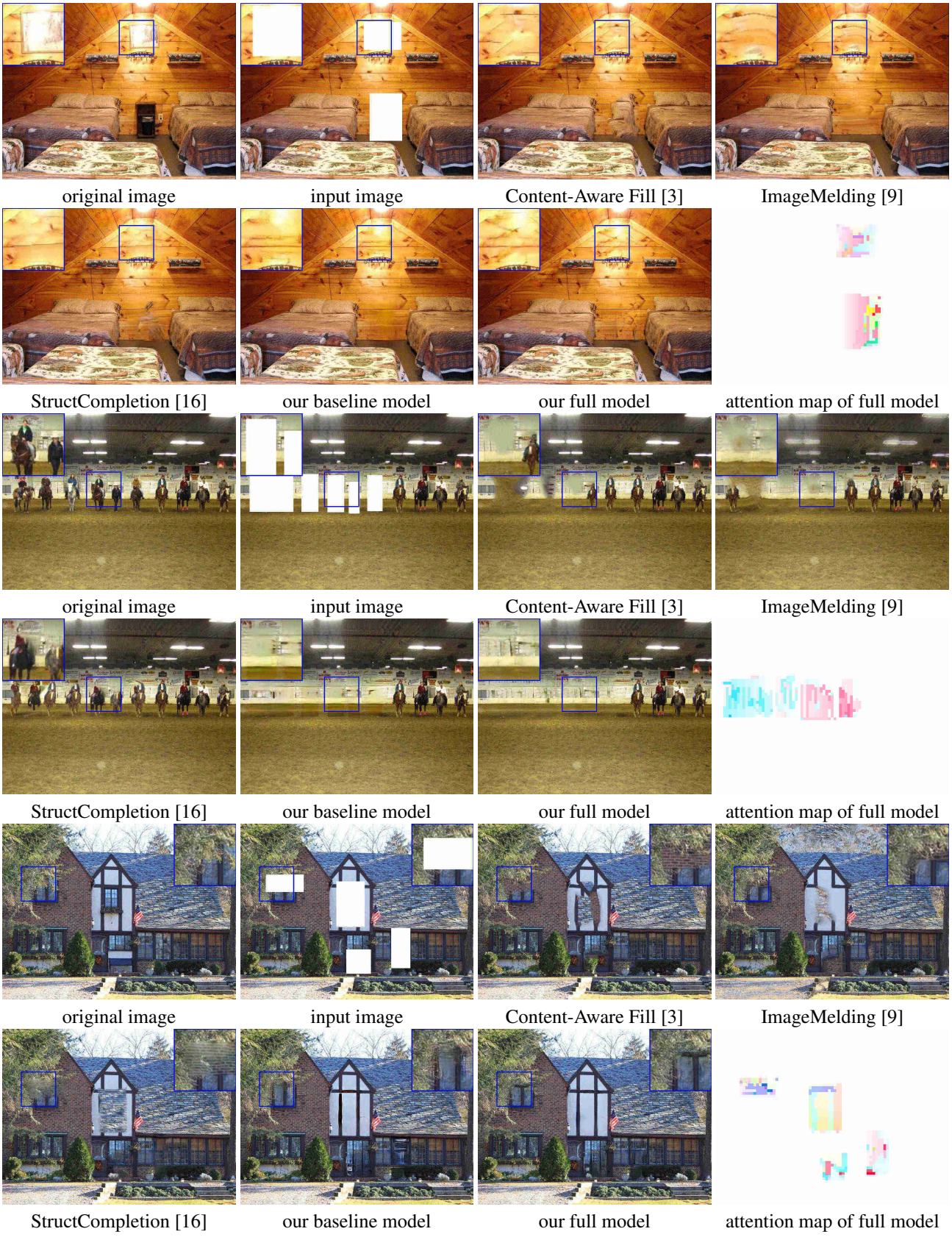

Figure 15: More qualitative results and comparisons. All input images are masked from validation set. All our results are direct outputs from same trained model without post-processing. Best viewed with zoom-in.

Figure 16: Visualization (highlighted regions) on which parts in input image are attended. Each triad, from left to right, shows input image, result and attention visualization.