EdgeConnect: Structure Guided Image Inpainting using Edge Prediction

Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi
University of Ontario Institute of Technology, Canada
{kamyar.nazeri, eric.ng, tony.joseph, faisal.qureshi, mehran.ebrahimi}@uoit.ca
http://www.ImagingLab.ca http://www.VCLab.ca

Abstract

In recent years, many deep learning techniques have been applied to the image inpainting problem: the task of filling incomplete regions of an image. However, these models struggle to recover and/or preserve image structure especially when significant portions of the image are missing. We propose a two-stage model that separates the inpainting problem into structure prediction and image completion. Similar to sketch art, our model first predicts the image structure of the missing region in the form of edge maps. Predicted edge maps are passed to the second stage to guide the inpainting process. We evaluate our model end-to-end over publicly available datasets CelebA, CelebHQ, Places2, and Paris StreetView on images up to a resolution of 512×512*. We demonstrate that this approach outperforms current state-of-the-art techniques quantitatively and qualitatively.*

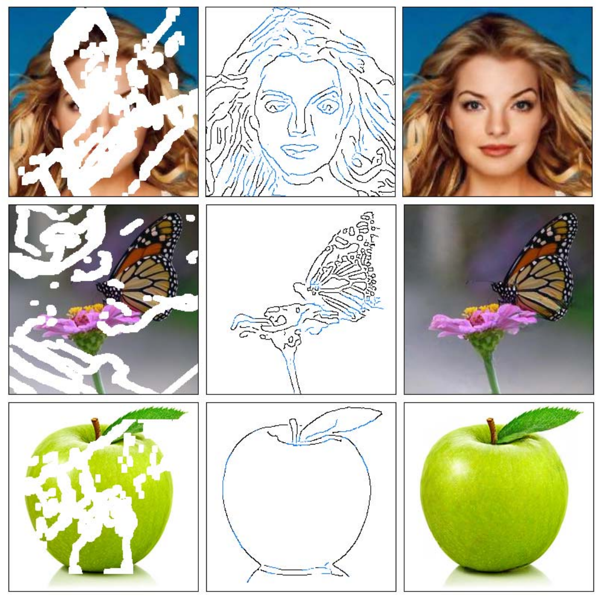

Figure 1: (Left) Input images with missing regions. The missing regions are depicted in white. (Center) Computed edge masks. Edges drawn in black are computed (for the available regions) using Canny edge detector; whereas edges shown in blue are hallucinated (for the missing regions) by the edge generator network. (Right) Image inpainting results of the proposed approach.

1. Introduction

Image inpainting, or image completion, involves filling in missing regions of an image. It is an important step in many image editing tasks. For example, it can be used to fill in empty regions after removing unwanted objects from an image. Filled regions must be perceptually plausible since humans have an uncanny ability to zero in on visual inconsistencies. The lack of fine structure in the filled region is a giveaway that something is amiss, especially when the rest of the image contain sharp details. The work presented in this paper is motivated by our observation that many existing inpainting techniques generate over-smoothed and/or blurry regions where detailed image structure is expected.

Like most computer vision problems, image inpainting predates the wide-spread use of deep learning techniques. Broadly speaking, traditional approaches for image inpainting can be divided into two groups: diffusion-based and patch-based. Diffusion-based methods propagate background data into the missing region by numerically solving corresponding partial differential equations (PDEs) that model desired behaviors [4, 15, 30, 2]. Patch-based methods, on the other hand, fill in missing regions with patches from a collection of source images that maximize patch similarity [8, 23].

More recently, deep learning approaches have found remarkable success at the task of image inpainting. These schemes fill the missing pixels using a learned data distribution. They are able to generate coherent structures in the missing regions, a feat that was nearly impossible for traditional techniques without significant user intervention. While these approaches are able to generate missing regions with meaningful structures, the generated regions are often blurry or suffer from artifacts, suggesting that these methods struggle to reconstruct high frequency information accurately.

Then, how does one force an image inpainting network to generate fine details? Since image structure is well-represented in its edge mask, we show that it is possible to generate superior results by conditioning an image inpainting network on edges in the missing regions. Our approach of "lines first, color next" is partly inspired by our understanding of how artists work [14]. "*In line drawing, the lines not only delineate and define spaces and shapes; they also play a vital role in the composition*", says Betty Edwards, highlights the importance of sketches from an artistic viewpoint [13]. Edge recovery, we suppose, is an easier task than image completion. We propose a model that essentially decouples the recovery of high and low-frequency information of the inpainted region.

We divide the image inpainting process into a two-stages (Figure 1): edge generation and image completion. Edge generation is solely focused on hallucinating edges in the missing regions. The image completion then estimates RGB intensities of the region using hallucinated edges. Both stages follow an adversarial framework [19] to ensure that the hallucinated edges and the RGB pixel intensities are visually consistent. Losses based on deep features are in corporated into both networks to enforce perceptually realistic results.

We evaluate our proposed model on standard datasets CelebA [33], CelebHQ [27], Places2 [59], and Paris StreetView [9]. We compare the performance of our model against current state-of-the-art schemes. Furthermore, we provide results of experiments carried out to study the effects of edge information on the image inpainting task. Our paper makes the following contributions:

- An edge generator that approximates structural information by predicting edge data in missing regions of an image.
- We demonstrate that using image structure as *a priori* significantly improves inpainting results.

We show that our model can be used in some common image editing applications, such as object removal and scene generation. Our source code is available at:
`https://github.com/knazeri/edge-connect`

2. Related Work

Diffusion-based methods propagate neighboring information into the missing regions [4, 2]. [15] adapted the Mumford-Shah segmentation model for image inpainting by introducing Euler's Elastica. However, reconstruction is restricted to locally available information for these diffusion-based methods, and these methods fail to recover meaningful structures in the missing regions especially for cases with large missing regions. Structure guided diffusion-based methods have also been proposed such as [5, 47, 22].

Patch-based methods fill in missing regions (*i.e.*, targets) by copying information from similar regions (*i.e.*, sources) of the same image (or a collection of images). Source regions are often blended into the target regions to minimize discontinuities [8, 23]. These methods are computationally expensive since similarity scores must be computed for every target-source pair. PatchMatch [3] addressed this issue by using a fast nearest neighbor field algorithm. These methods, however, assume that the texture of the inpainted region can be found elsewhere in the image, which may not always hold. Consequently, these methods excel at recovering highly patterned regions such as background completion but struggle at reconstructing structure that are locally unique.

One of the first *deep learning* methods designed for image inpainting is context encoder [40], which uses an encoder-decoder architecture. The encoder maps an image with missing regions to a low-dimensional feature space, which the decoder uses to construct the output image. Due to the information bottleneck in the channel-wise fully connected layer, recovered regions of the output image often contain visual artifacts and exhibit blurriness. This was addressed by Iizuka *et al*. [24] by reducing the number of downsampling layers. To preserve the effective receptive field from the reduction of downsampling layers, the channel-wise fully connected layer was replaced by a series of dilated convolution layers [54] with varying dilation factors. However, the training time was increased significantly.[1] Yang *et al*. [52] uses a pre-trained VGG network [44] to improve the output of the context-encoder, by minimizing the feature difference of image background. This approach requires solving a multi-scale optimization problem iteratively, which noticeably increases computational cost during inference time. Liu *et al*. [31] introduced "partial convolution" for image inpainting, where convolution weights are normalized by the mask area of the window that the convolution filter currently resides over. This effectively prevents the convolution filters from capturing too many zeros when they traverse over the incomplete region.

Recently, several methods were introduced by providing additional information prior to inpainting. Yeh *et al*. [53] trains a GAN for image inpainting with uncorrupted data. During inference, back-propagation is employed for $1,500$ iterations to find the representation of the corrupted image on a uniform noise distribution. However, the model is slow during inference since back-propagation must be performed for every image it attempts to recover. Dolhansky and Ferrer [10] demonstrate the importance of exemplar information for inpainting. Their method is able to achieve both sharp and realistic inpainting results when filling in miss-

[1] Model by [24] required two months of training over four GPUs.

ing eye regions in frontal human face images. Contextual Attention [56] takes a two-step approach to the problem of image inpainting by first producing a coarse estimate of the missing region. The initial estimate is passed to a refinement network with an attention mechanism that searches for a collection of background patches with the highest similarity to the coarse estimate. [45] takes a similar approach and introduces a "patch-swap" layer which replaces each patch inside the missing region with the most similar patch on the boundary. SPG-Net [46] also follows a two-stage model which uses semantic segmentation labels to guide the inpainting process. Free-form inpainting method proposed in [55] is perhaps closest in spirit to our work, by using hand-drawn sketches to guide the inpainting process. Our method does away with hand-drawn sketches and instead learns to hallucinate edges in the missing regions.

2.1. Image-to-Edges vs. Edges-to-Image

The inpainting technique proposed in this paper subsumes two disparate computer vision problems: Image-to-Edges and Edges-to-Image. There is a large body of literature that addresses "Image-to-Edges" problems [6, 11, 29, 32]. Canny edge detector, an early scheme for constructing edge maps, for example, is roughly 30 years old [7]. Dollár and Zitnikc [12] use *structured learning* [37] on random decision forests to predict local edge masks. Holistically-nested Edge Detection (HED) [51] is a fully convolutional network that learns edge information based on its importance as a feature of the overall image. In our work, we train on edge maps computed using Canny edge detector. We explain this in detail in Section 4.1 and Section 5.3.

Traditional "Edges-to-Image" methods typically follow a bag-of-words approach, where image content is constructed through a pre-defined set of keywords. These methods, however, are unable to accurately construct fine-grained details especially near object boundaries. Scribbler [43] is a learning-based model where images are generated using line sketches as the input. The results of their work possess an art-like quality, where color distribution of the generated result is guided by the use of color in the input sketch. Isola *et al*. [25] proposed a conditional GAN framework [35], called pix2pix, for image-to-image translation problems. This scheme can use available edge information as *a priori*. CycleGAN [60] extends this framework and finds a reverse mapping back to the original data distribution. This approach yields superior results since the aim is to learn the inverse of the forward mapping.

3. EdgeConnect

We propose an image inpainting network that consists of two stages: 1) edge generator, and 2) image completion network (Figure 2). Both stages follow an adversarial model [19], *i.e*. each stage consists of a generator/discriminator pair. Let G_1 and D_1 be the generator and discriminator for the edge generator, and G_2 and D_2 be the generator and discriminator for the image completion network, respectively. To simplify notation, we will use these symbols also to represent the function mappings of their respective networks.

Our generators follow an architecture similar to the method proposed by Johnson *et al*. [26], which has achieved impressive results for style transfer, super-resolution [42, 18], and image-to-image translation [60]. Specifically, the generators consist of encoders that down-sample twice, followed by eight residual blocks [20] and decoders that up-sample images back to the original size. Dilated convolutions with a dilation factor of eight are used instead of regular convolutions in the residual layers to increase receptive field in subsequent layers. For discriminators, we use a 70×70 PatchGAN [25, 60] architecture, which determines whether or not overlapping image patches of size 70×70 are real. We use instance normalization [48] across all layers of the network[2].

3.1. Edge Generator

Let $\mathbf{I}_{gt}$ be ground truth images. Their edge map and grayscale counterpart will be denoted by $\mathbf{C}_{gt}$ and $\mathbf{I}_{gray}$, respectively. In the edge generator, we use the masked grayscale image $\tilde{\mathbf{I}}_{gray} = \mathbf{I}_{gray} \odot (\mathbf{1} - \mathbf{M})$ as the input, its edge map $\tilde{\mathbf{C}}_{gt} = \mathbf{C}_{gt} \odot (\mathbf{1} - \mathbf{M})$, and image mask $\mathbf{M}$ as a pre-condition (1 for the missing region, 0 for background). Here, $\odot$ denotes the Hadamard product. The generator predicts the edge map for the masked region

$$\mathbf{C}_{pred} = G_1\left(\tilde{\mathbf{I}}_{gray}, \tilde{\mathbf{C}}_{gt}, \mathbf{M}\right). \tag{1}$$

We use $\mathbf{C}_{gt}$ and $\mathbf{C}_{pred}$ conditioned on $\mathbf{I}_{gray}$ as inputs of the discriminator that predicts whether or not an edge map is real. The network is trained with an objective comprised of the hinge variant of GAN loss [36] and feature-matching loss [49]

$$\mathcal{J}_{G_1} = \lambda_{G_1}\mathcal{L}_{G_1} + \lambda_{FM}\mathcal{L}_{FM} \tag{2}$$

where λ_{G_1} and λ_{FM} are regularization parameters. The hinge losses over the generator and discriminator are defined as

$$\mathcal{L}_{G_1} = -\mathbb{E}_{\mathbf{I}_{gray}}\left[D_1(\mathbf{C}_{pred}, \mathbf{I}_{gray})\right], \tag{3}$$

$$\begin{aligned}\mathcal{L}_{D_1} = \mathbb{E}_{(\mathbf{C}_{gt}, \mathbf{I}_{gray})}\left[\max(0, 1 - D_1(\mathbf{C}_{gt}, \mathbf{I}_{gray}))\right] \\ + \mathbb{E}_{\mathbf{I}_{gray}}\left[\max(0, 1 + D_1(\mathbf{C}_{pred}, \mathbf{I}_{gray}))\right].\end{aligned} \tag{4}$$

The feature-matching loss $\mathcal{L}_{FM}$ compares the activation maps in the intermediate layers of the discriminator. This stabilizes the training process by forcing the generator to

[2] The details of our architecture are in the supplementary material.

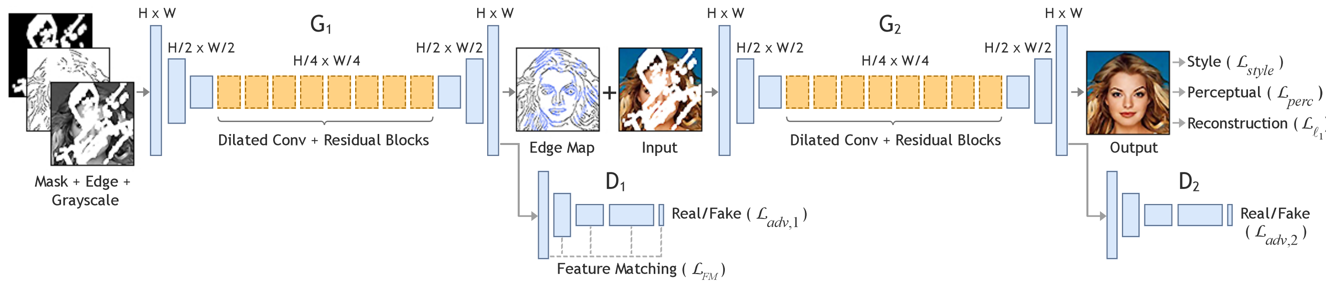

Figure 2: Summary of our proposed method. Incomplete grayscale image and edge map, and mask are the inputs of G_1 to predict the full edge map. Predicted edge map and incomplete color image are passed to G_2 to perform the inpainting task.

produce results with representations that are similar to real images. This is similar to perceptual loss [26, 17, 16], where activation maps are compared with those from the pre-trained VGG network. However, since the VGG network is not trained to produce edge information, it fails to capture the result that we seek in the initial stage. The feature matching loss $\mathcal{L}_{FM}$ is defined as

$$\mathcal{L}_{FM} = \mathbb{E}\left[\sum_i \frac{1}{N_i} \left\| D_1^{(i)}(\mathbf{C}_{gt}) - D_1^{(i)}(\mathbf{C}_{pred}) \right\|_1 \right], \quad (5)$$

where, N_i is the number of elements in the i'th activation layer, and $D_1^{(i)}$ is the activation in the i'th layer of the discriminator. Spectral normalization (SN) [36] further stabilizes training by scaling down weight matrices by their respective largest singular values, effectively restricting the Lipschitz constant of the network to one. Although this was originally proposed to be used only on the discriminator, recent works [57, 38] suggest that generator can also benefit from SN by suppressing sudden changes of parameter and gradient values. Therefore, we apply SN to both generator and discriminator. Spectral normalization was chosen over Wasserstein GAN (WGAN) [1], as we found that WGAN was several times slower in our early tests. For our experiments, we choose $\lambda_{G_1} = 1$ and $\lambda_{FM} = 10$.

3.2. Image Completion Network

The image completion network uses the incomplete color image $\tilde{\mathbf{I}}_{gt} = \mathbf{I}_{gt} \odot (\mathbf{1} - \mathbf{M})$ as input, conditioned using a composite edge map $\mathbf{C}_{comp}$. The composite edge map is constructed by combining the background region of ground truth edges with generated edges in the corrupted region from the previous stage, *i.e.* $\mathbf{C}_{comp} = \mathbf{C}_{gt} \odot (\mathbf{1} - \mathbf{M}) + \mathbf{C}_{pred} \odot \mathbf{M}$. The network returns a color image $\mathbf{I}_{pred}$, with missing regions filled in, that has the same resolution as the input image:

$$\mathbf{I}_{pred} = G_2\left(\tilde{\mathbf{I}}_{gt}, \mathbf{C}_{comp}\right). \quad (6)$$

This is trained over a joint loss that consists of an ℓ_1 loss, hinge loss, perceptual loss, and style loss. To ensure proper scaling, the ℓ_1 loss is normalized by the mask size. The hinge loss is similar to 3, 4:

$$\mathcal{L}_{G_2} = -\mathbb{E}_{\mathbf{C}_{comp}}\left[D_2(\mathbf{I}_{pred}, \mathbf{C}_{comp})\right], \quad (7)$$

$$\begin{aligned}\mathcal{L}_{D_2} = &\ \mathbb{E}_{(\mathbf{I}_{gt}, \mathbf{C}_{comp})}\left[\max(0, 1 - D_2(\mathbf{I}_{gt}, \mathbf{C}_{comp}))\right] \\ &+ \mathbb{E}_{\mathbf{C}_{comp}}\left[\max(0, 1 + D_2(\mathbf{I}_{pred}, \mathbf{C}_{comp}))\right]. \end{aligned} \quad (8)$$

We include the two losses proposed in [17, 26] commonly known as perceptual loss $\mathcal{L}_{perc}$ and style loss $\mathcal{L}_{style}$. As the name suggests, $\mathcal{L}_{perc}$ penalizes results that are not perceptually similar to labels by defining a distance measure between activation maps of a pre-trained network. Perceptual loss is defined as

$$\mathcal{L}_{perc} = \mathbb{E}\left[\sum_i \frac{1}{N_i} \left\|\phi_i(\mathbf{I}_{gt}) - \phi_i(\mathbf{I}_{pred})\right\|_1\right], \quad (9)$$

where ϕ_i is the activation map in the i'th layer of a pre-trained network. For our work, ϕ_i corresponds to activation maps from layers `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` and `relu5_1` of the VGG-19 network pre-trained on the ImageNet dataset [41]. These activation maps are also used to compute style loss which measures the differences between covariances of the activation maps. Given feature maps of sizes $N_i = C_j \times H_j \times W_j$, style loss is computed by

$$\mathcal{L}_{style} = \mathbb{E}\left[\sum_j \|G_j^{\phi}(\tilde{\mathbf{I}}_{pred}) - G_j^{\phi}(\tilde{\mathbf{I}}_{gt})\|_1\right], \quad (10)$$

where G_j^{ϕ} is a $C_j \times C_j$ Gram matrix constructed from activation maps ϕ_j [17]. We choose to use style loss as it was shown by Sajjadi *et al.* [42] to be an effective tool to combat "checkerboard" artifacts caused by transpose convolution layers [39]. Our overall loss function is

$$\mathcal{J}_{G_2} = \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{G_2}\mathcal{L}_{G_2} + \lambda_p\mathcal{L}_{perc} + \lambda_s\mathcal{L}_{style}. \quad (11)$$

For our experiments, we choose $\lambda_{\ell_1} = 1$, $\lambda_{G_2} = \lambda_p = 0.1$, and $\lambda_s = 250$. We noticed that the training time increases significantly if spectral normalization is included. We believe this is due to the network becoming too restrictive with the increased number of terms in the loss function. Therefore we choose to exclude spectral normalization from the image completion network.

4. Experiments

4.1. Edge Information and Image Masks

To train G_1, we generate training labels (*i.e*. edge maps) using Canny edge detector. The sensitivity of Canny edge detector is controlled by the standard deviation of the Gaussian smoothing filter σ. For our tests, we empirically found that $\sigma \approx 2$ yields the best results (Figure 4). In Section 5.3, we investigate the effect of the quality of edge maps on the overall image completion.

For our experiments, we use two types of image masks: regular and irregular. Regular masks are square masks of fixed size (25% of total image pixels) centered at a random location within the image. We obtain irregular masks from the work of Liu *et al*. [31]. Irregular masks are classified based on their sizes relative to the entire image in increments of 10% (*e.g*. 0-10%, 10-20%, *etc*.). All bins are divided into two batches of 1,750 and 250 masks for training and testing purposes respectively. Once separated, masks are augmented by introducing four rotations $(0°, 90°, 180°, 270°)$ and a horizontal reflection for each mask. This ensures that augmented variations of masks were not shared between the training and testing sets.

4.2. Training Setup and Strategy

Our proposed model is implemented in PyTorch. The network is trained with 256×256 images with batch size of eight to obtain results for quantitative comparisons with existing methods. The model is optimized using Adam optimizer [28] with $\beta_1 = 0$ and $\beta_2 = 0.9$. Generators G_1, G_2 are trained separately using Canny edges with learning rate 10^{-4} until the losses plateau. We lower the learning rate to 10^{-5} and continue to train G_1 and G_2 until convergence. Finally, we freeze training on G_1 while continue to train G_2. For visual comparisons presented in this paper, our model was trained with 512×512 images using pre-trained weights from the 256×256 model with the same hyper-parameters.

5. Results

Our proposed model is evaluated on the datasets CelebA [33], CelebHQ [27], Places2 [59], and Paris StreetView [9]. For the baseline, we use our image completion network only (no edge data, G_2 only). Results of the full model are compared against the the baseline and current state-of-the-art methods both qualitatively and quantitatively.

5.1. Qualitative Comparison

Figure 3 shows a sample of images generated by our model. For visualization purposes, we reverse the colors of $\mathbf{C}_{comp}$. Our model is able to generate photo-realistic results with a large fraction of image structures remaining intact. Furthermore, by including style loss, the inpainted images lack any "checkerboard" artifacts in the generated results [31]. As importantly, the inpainted images exhibit minimal blurriness. We conjecture that providing edge information alleviates the burden of preserving structure from the network. Thus it only needed to learn color distribution.

5.2. Quantitative Comparison

Numerical Metrics Since existing models were evaluated using 256×256, we evaluated our model trained on images of the same resolution to ensure fair comparisons. The performance of our model was measured using the following metrics: 1) relative ℓ_1; 2) structural similarity index (SSIM) [50], with a window size of 11; 3) peak signal-to-noise ratio (PSNR); and 4) Fréchet Inception Distance (FID) [21]. Since relative ℓ_1, SSIM, and PSNR assume pixel-wise independence, these metrics may assign favorable scores to perceptually inaccurate results. Recent works [58, 57, 10] have shown that FID serves as the preferred metric for human perception. Note that since FID is a dissimilarity measure between high-level features, it may not reflect low-level color consistencies that attribute to visual quality. While FID may not be the ideal metric to measure inpainting quality, we believe the combination of the listed metrics provided a better picture of inpainting performance. The results over Places2 dataset are reported in Table 1. Note that these statistics are based on the synthesized image which mostly comprises of the ground truth image. Therefore our reported FID values are lower than other generative models reported in [34]. Statistics for competing techniques are obtained using their respective pre-trained weights, where available[3,4], and are calculated over 10,000 random images in the test set. The full model of Partial Convolution (PConv) is not available at the time of writing. We implemented PConv based on the guidelines in [31] using the PConv layer that is publicly available.[5]

Visual Turing Tests We evaluate our results by performing *yes-no tasks* (Y-N) and *just noticeable differences* (JND). For Y-N, a single image was randomly sampled from either ground truth images, or images generated by our model. Participants were asked whether the sampled image was real or not. For JND, we asked participants to select the more realistic image from pairs of real and generated images. For both tests, two seconds were given for each image set(s). The tests were performed over 300 images for each model and mask size. Each image was shown 10 times in total. The results are summarized in Table 2.

[3] https://github.com/JiahuiYu/generative_inpainting
[4] https://github.com/satoshiiizuka/siggraph2017_inpainting
[5] https://github.com/NVIDIA/partialconv

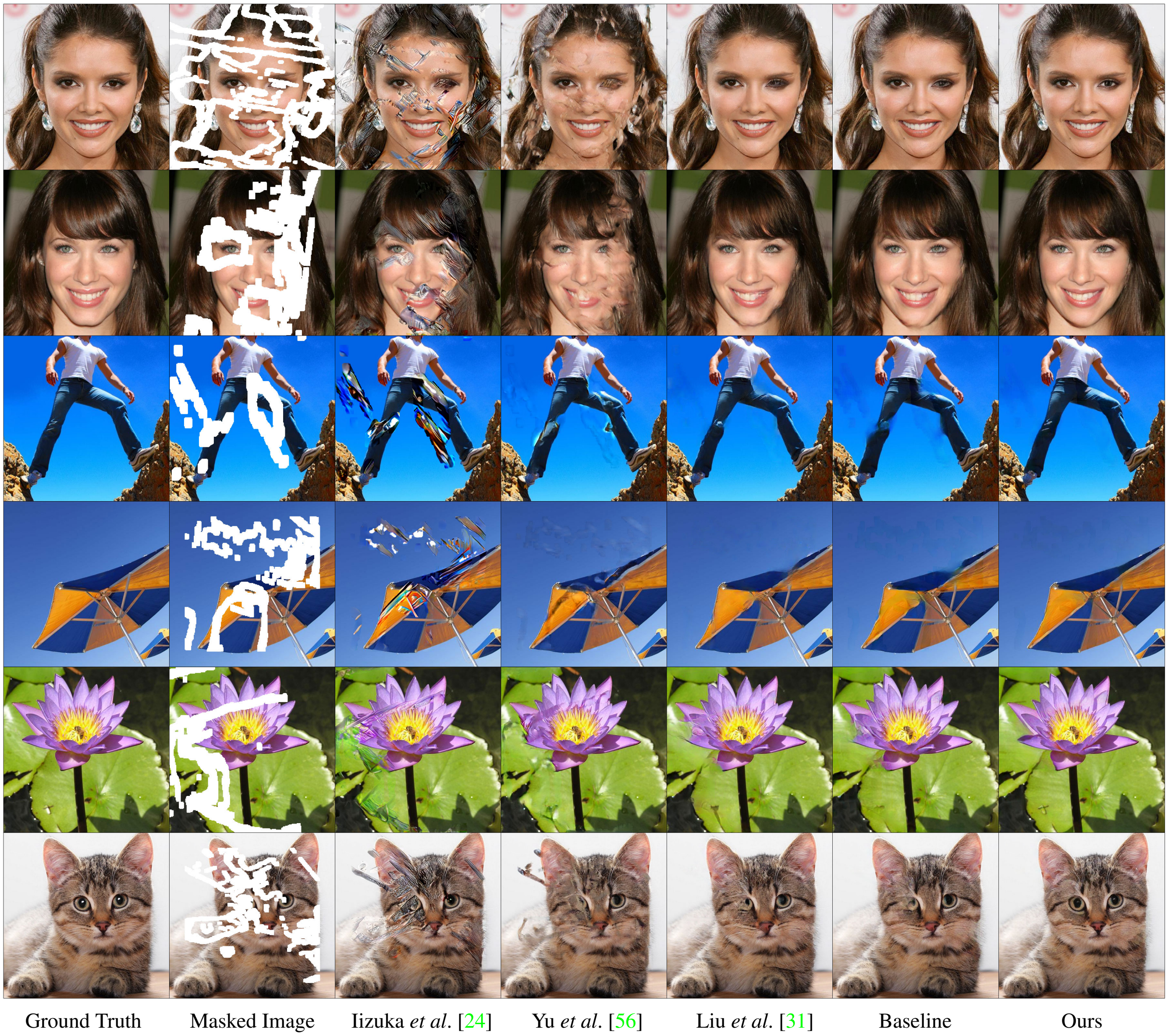

Figure 3: Comparison of qualitative results of 512×512 images with existing models. From left to right: Ground Truth, Masked Image, Iizuka *et al*. [24], Yu *et al*. [56], Liu *et al*. (Partial Convolution) [31], Baseline (no edge data, G_2 only), Ours (Full Model).

5.3. Ablation Study

Quantity of Edges versus Inpainting Quality We now turn our attention to the key assumption of this work: edge information helps with image inpainting. Table 3 shows inpainting results with and without edge information. Our model achieved better scores for every metric when edge information was incorporated into the inpainting model, even when a significant portion of the image is missing.

Next, we turn to a more interesting question: How much edge information is needed to see improvements in the generated images? We again use Canny edge detector to construct edge information. We use the parameter σ to control the amount of edge information available to the image completion network. Specifically, we train our image completion network using edge maps generated for $\sigma = 0, 0.5, \ldots, 5.5$, and we found that the best image inpainting results are obtained with edges corresponding to $\sigma \in [1.5, 2.5]$, across all datasets shown in Figure 4. For large values of σ, too few edges are available to make a difference in the quality of generated images. On the other hand, when σ is too small, too many edges are produced,

	Mask	CA	GLCIC	PConv	Ours
ℓ_1 (%)†	10-20%	2.41	2.66	1.55	**1.50**
	20-30%	4.23	4.70	2.71	**2.59**
	30-40%	6.15	6.78	3.94	**3.77**
	40-50%	8.03	8.85	5.35	**5.14**
	Fixed	4.37	4.12	3.95	**3.86**
SSIM*	10-20%	0.893	0.862	0.916	**0.920**
	20-30%	0.815	0.771	0.854	**0.861**
	30-40%	0.739	0.686	0.789	**0.799**
	40-50%	0.662	0.603	0.720	**0.731**
	Fixed	0.818	0.814	0.818	**0.823**
PSNR*	10-20%	24.36	23.49	27.54	**27.95**
	20-30%	21.19	20.45	24.47	**24.92**
	30-40%	19.13	18.50	22.42	**22.84**
	40-50%	17.75	17.17	20.77	**21.16**
	Fixed	20.65	21.34	21.54	**21.75**
FID†	10-20%	6.16	11.84	**2.26**	2.32
	20-30%	14.17	25.11	**4.88**	4.91
	30-40%	24.16	39.88	**8.84**	8.91
	40-50%	35.78	54.30	15.18	**14.98**
	Fixed	8.31	8.42	10.53	**8.16**

Table 1: Quantitative results over Places2 (256×256) with models: Contextual Attention (CA) [56], Globally and Locally Consistent Image Completion (GLCIC) [24], Partial Convolution (PConv) [31], Ours. The best result of each row is boldfaced except for Canny. †Lower is better. *Higher is better.

	Mask	CA	GLCIC	PConv	Ours
JND (%)	10-20%	20.98	16.91	36.04	**39.69**
	20-30%	15.45	14.27	30.09	**36.99**
	30-40%	12.86	12.29	20.60	**27.53**
	40-50%	12.74	10.91	18.31	**25.44**
Y-N (%)	10-20%	38.71	22.46	79.72	**88.66**
	20-30%	23.44	12.09	64.11	**77.59**
	30-40%	13.49	4.32	52.50	**66.44**
	40-50%	9.89	2.77	37.73	**58.02**

Table 2: Y-N and JND scores for various mask sizes on Places2. Y-N score for ground truth images is 94.6%.

which adversely affect the quality of the generated images. We used this study to set $\sigma = 2$ when creating ground truth edge maps for the training of the edge generator network.

Figure 5 shows how different values of σ affects the inpainting task. Note that in a region where edge data is sparse, the quality of the inpainted region degrades. For instance, in the generated image for $\sigma = 5$, the left eye was reconstructed much sharper than the right eye.

	CelebA		Places2	
Edges	No	Yes	No	Yes
ℓ_1 (%)	4.11	3.03	6.69	5.14
SSIM	0.802	0.846	0.682	0.731
PSNR	23.33	25.28	19.59	21.16
FID	6.16	2.82	32.18	14.98

Table 3: Comparison of inpainting results without edge information (G_2 only, baseline) versus results with edge information (full model). Statistics are based on $10,000$ random masks with size 40-50% of the entire image.

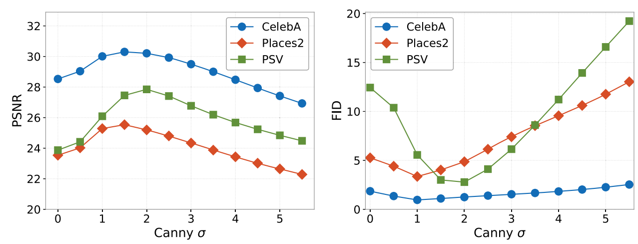

Figure 4: Effect of σ in Canny detector on PSNR and FID.

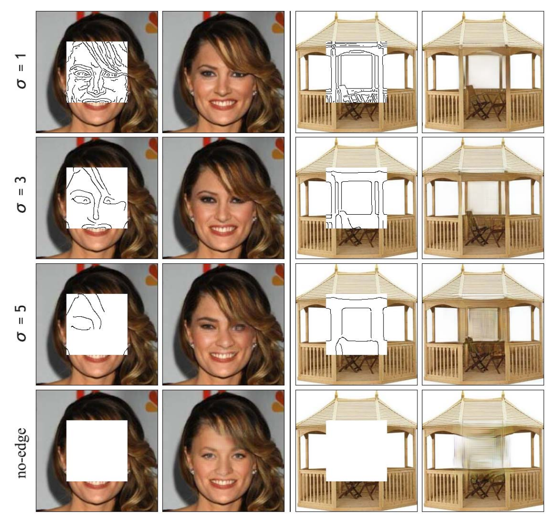

Figure 5: Effect of σ in Canny edge detector on inpainting results. Top to bottom: $\sigma = 1, 3, 5$, no edge data.

Alternative Edge Detection Systems We use Canny edge detector to produce training labels for the edge generator network due to its speed, robustness, and ease of use. Canny edges are one-pixel wide, and are represented as binary masks (1 for edge, 0 for background). Edges produced with HED [51], however, are of varying thickness, and pixels can have intensities ranging between 0 and 1. We noticed that it is possible to create edge maps that look eerily similar to human sketches by performing element-wise multiplication on Canny and HED edge maps (Figure 6). We trained our image completion network using the combined edge map. However, we did not notice any improvements in the inpainting results.[6]

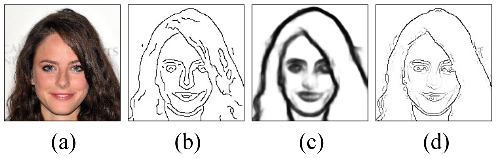

Figure 6: (a) Image. (b) Canny. (c) HED. (d) Canny⊙HED.

6. Discussions and Future Work

We proposed EdgeConnect, a new deep learning model for image inpainting tasks. EdgeConnect comprises of an edge generator and an image completion network, both following an adversarial model. We demonstrate that edge information plays an important role in the task of image inpainting. Our method achieves state-of-the-art results on standard benchmarks, and is able to deal with images with multiple, irregularly shaped missing regions.

While effectively delineating the edges is more useful than hundreds of detailed lines, our edge generating model sometimes struggles to accurately depict the edges in highly textured areas, or when a large portion of the image is missing especially for higher resolution images. We plan to address this with a multi-scale approach, by first predicting a low-resolution variant of edge data. The edge data will be up-sampled, then refined. This process is repeated until the desired resolution is reached. This allows image structure to be scaled up with minor degradation using common interpolation techniques. Since our image completion network was able to produce photo-realistic results provided that the edge data is accurate, we believe our fully convolutional model can be extended to very high-resolution inpainting applications by following a pyramid model for edge prediction.

The trained model can be used as an interactive image editing tool. We can, for example, manipulate objects in the edge domain and transform the edge maps back to generate a new image. We provide some examples in Figure 7. Here we have removed the right-half of a given image to be used as input. The edge maps, however, are provided by a different image. The generated image seems to share characteristics of the two images. Figure 8 further shows examples where we attempt to remove unwanted objects from existing images.

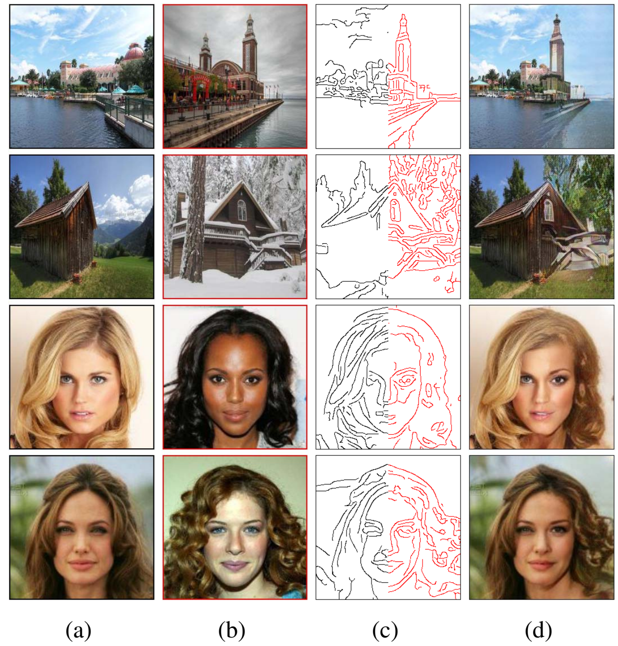

Figure 7: Edge-map (c) generated using the left-half of (a) (shown in black) and right-half of (b) (shown in red). Input is (a) with the right-half removed, producing the output (d).

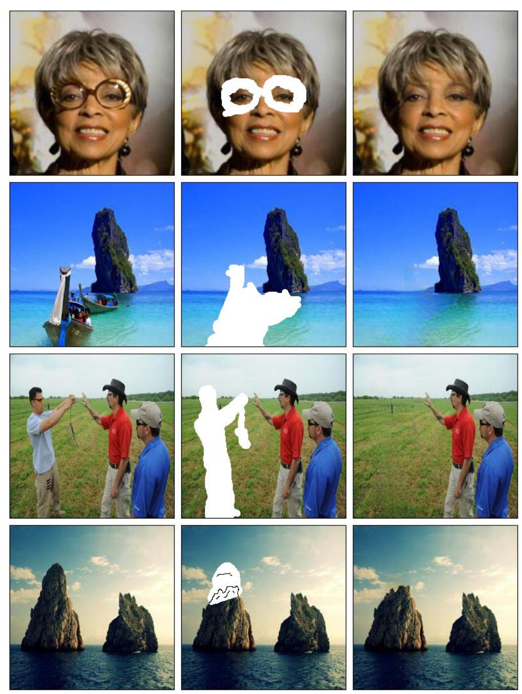

Figure 8: Examples of object removal and image editing using our EdgeConnect model. (Left) Original image. (Center) Unwanted object removed with optional edge information to guide inpainting. (Right) Generated image.

[6]Further analysis with HED are available in supplementary material.

Acknowledgments: This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Konstantinos G. Derpanis for helpful discussions and feedback. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 4

[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. 1, 2

[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)*, 28(3):24, 2009. 2

[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, 2000. 1, 2

[5] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003. 2

[6] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 3

[8] 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)*, 31(4):82–1, 2012. 1, 2

[9] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes paris look like paris? *ACM Transactions on graphics (TOG)*, 31(4), 2012. 2, 5

[10] B. Dolhansky and C. C. Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7902–7911, 2018. 2, 5

[11] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1964–1971. IEEE, 2006. 3

[12] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1558–1570, 2015. 3

[13] B. Edwards. *Drawing on the Right Side of the Brain: The Definitive, 4th Edition*. Penguin Publishing Group, 2012. 2

[14] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31(4):44–1, 2012. 2

[15] S. Esedoglu and J. Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(4):353–370, 2002. 1, 2

[16] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015. 4

[17] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 4

[18] M. W. Gondal, B. Schölkopf, and M. Hirsch. The unreasonable effectiveness of texture transfer for single image super-resolution. In *Workshop and Challenge on Perceptual Image Restoration and Manipulation (PIRM) at the 15th European Conference on Computer Vision (ECCV)*, 2018. 3

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2, 3

[20] 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 (CVPR)*, pages 770–778, 2016. 3

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 5

[22] H. Huang, K. Yin, M. Gong, D. Lischinski, D. Cohen-Or, U. M. Ascher, and B. Chen. " mind the gap": tele-registration for structure-driven image completion. *ACM Transactions on Graphics (TOG)*, 32(6):174–1, 2013. 2

[23] 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. 1, 2

[24] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 2, 6, 7

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[26] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 3, 4

[27] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 2, 5

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[29] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1619–1627, 2016. 3

[30] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang. Image compression with edge-based inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(10):1273–1287, 2007. 1

[31] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision (ECCV)*, September 2018. 2, 5, 6, 7

[32] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5872–5881. IEEE, 2017. 3

[33] 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)*, 2015. 2, 5

[34] M. Lučić, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, 2018. 5

[35] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[36] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 3, 4

[37] S. Nowozin, C. H. Lampert, et al. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011. 3

[38] A. Odena, J. Buckman, C. Olsson, T. B. Brown, C. Olah, C. Raffel, and I. Goodfellow. Is generator conditioning causally related to gan performance? In *Proceedings of the 35th International Conference on Machine Learning*, 2018. 4

[39] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 4

[40] 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 (CVPR)*, pages 2536–2544, 2016. 2

[41] 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. 4

[42] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4501–4510. IEEE, 2017. 3, 4

[43] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2

[45] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C. Jay. Contextual-based image inpainting: Infer, match, and translate. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[46] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C. J. Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 97, 2018. 3

[47] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 861–868. ACM, 2005. 2

[48] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[49] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5, 2018. 3

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[51] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1395–1403, 2015. 3, 7

[52] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[53] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[54] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[55] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 3

[56] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7

[57] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 4, 5

[58] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[59] 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. 2, 5

[60] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 3