General Deep Image Completion with Lightweight Conditional Generative Adversarial Networks

Ching-Wei Tseng
sky1001993@gmail.com
Hung Jin Lin
vtsh.jn@gmail.com
Shang-Hong Lai
lai@cs.nthu.edu.tw

Computer Vision Lab
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

Abstract

Recent image completion researches using deep neural networks approaches have shown remarkable progress by using generative adversarial networks (GANs). However, these approaches still have the problems of large model sizes and lack of generality for various types of corruptions. In addition, the conditional GANs often suffer from the mode collapse and unstable training problems. In this paper, we overcome these shortcomings in the previous models by proposing a lightweight model of conditional GANs with integrating a stable way in adversarial training. Moreover, we present a new training strategy to trigger the model to learn how to complete different types of corruptions or missing regions in images. Experimental results demonstrate qualitatively and quantitatively that the proposed model provides significant improvement over a number of representative image completion methods on public datasets. In addition, we show that our model requires much less model parameters to achieve superior results for different types of unseen corruption masks.

1 Introduction

Image inpainting and completion are classical problems in computer vision and graphics. The objective is to fill semantic and reasonable contents to the corruptions (missing regions) in an image. Humans can fill the missing regions by the empirical knowledge to the diverse object structures from the real world. Nevertheless, it is not easy for machines to learn a wide variety of structures in natural images and predict what to fill in unknown missing-data regions in images. Thus, it is crucial to know how to learn an image transformation from a corrupted image with missing data to a completed image.

In the past, vision-based methods [5, 9, 12, 13, 16, 17, 29, 32] mainly focus on utilizing existing patterns and structure information from non-corrupted regions to fill in the corrupted part in a copy-paste manner or optimizing a specific cost function. Such approaches perform well for the cases where surrounding contexts share similar patterns and colors. Nonetheless, previous approaches are quite limited to complete various kinds of corrupted images since image contents are quite complicated and usually come with distinct structures. Retrieving image contents for filling the missing regions from the remaining completed images or millions of images [11] cannot not serve as a general or semantically meaningful solution in most cases. Recently, researchers try to apply deep learning methodologies to reconstruct an semantic image by learning image statistics for the image completion problem.

Recent deep network researches have brought great improvements on various computer vision problems. Moreover, generative adversarial training technique[10] can produce stunning results on image generation[27], and other tasks that condition on images, such as super-resolutions[18], style transfers/texture synthesis[19], image completion[20, 26, 33], and other image translation problems[14]. The advantage is that these conditional GANs models can learn the latent representations of images, and can reconstruct the desired images in a min-max optimization of generator and discriminator. Nevertheless, these type of deep-network based models often suffer from requiring large training parameters, and unstable condition caused by adversarial training. Recently, there are several newly designed adversarial training strategies[4, 6, 23] that can improve such issues on image generation tasks, but how to combine with conditional GANs still remains undiscovered. In the image completion problems, most deep-network based methods[26, 33] can only be applied to image completion with a specific corrupted mask, and the model needs to be retrained for a different type of corruptions.

To address the aforementioned issue, we aim to transform corrupted images with different types of missing regions to completed images that look as natural as possible (as in Figure 1). We propose a new training strategy and a lightweight conditional generative adversarial model that can effectively resume various corrupted images to flawless and realistic images. In the experiments, we demonstrate that the proposed model provides the best performance among vision-based and deep-network based methods on multiple datasets. Furthermore, the proposed method can be applied to real-world images (high resolution) with unseen missing regions. After cautious validations, we summarize our contributions as:

- We build a lightweight conditional GANs that first adapt more stable adversarial learning from LSGAN[23]. Our experiments show the proposed model requires less training parameters and outperform other deep models on wide range datasets.

- We propose a new training strategy that generates four representative types of corruptions to enhance learning generalization that can complete various types of corrupted images. Different from other methods, the proposed method can deal with corrupted masks that are different in terms of shapes or locations in the images.

2 Proposed Model

In this section, we will present how the proposed model accomplishes image completion tasks. Figure 2 gives an overview of our model. First, we show the details of our network architecture composed of an autoencoder $\mathcal{G}$ (containing two parts: semantic feature extractor and a simple generator), and a discriminator $\mathcal{D}$. Then, we introduce an objective function minimizing unrealistic content error, and training strategies that are applied to solve various types of corruptions in image completion problems. In the testing phase, the semantic feature extractor and simple generator (combined as $\mathcal{G}$) are only needed to recover corrupted images.

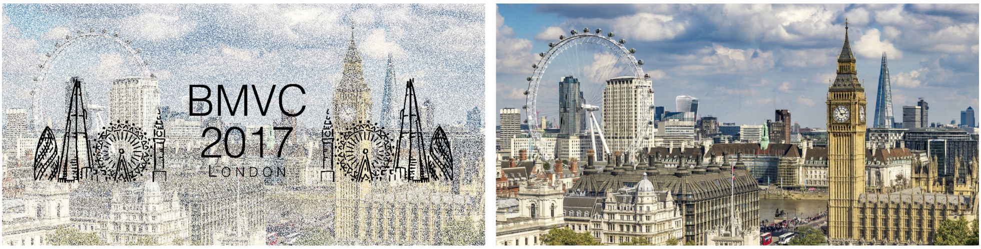

Figure 1: A teaser result from the proposed deep image completion model. We can apply our trained model on any resolution public images with user-defined corruption masks. The corrupted parts are visualized in white dots, black drawings and texts (in the left image). The right image depicts the completed images by using our model. The size of image is 640 × 1280. The images and corruption materials are from [1, 2]

2.1 Network Architecture

Semantic Feature Extractor: Given a corrupted image I with size $H \times W$ as input, we use a part of layers (before conv4-1) from VGG19 [30] architecture as our semantic feature extractor (encoder) to obtain a high-level and semantic feature patch. To preserve image details, we replace all the pooling layer with strided convolutions as the pooling layer has been proved[14, 26] that it tends to lose some image information in the reconstruction based networks. Also, we decrease the filter numbers of each layer to reduce the total model size. Each convolution is followed by ELU activation [7] that enhances the performance of the autoencoder.

Simple Generator: Common deep encoder-decoder networks use symmetric structure that extracts features and generates outcome through the same number of layers. However, if the layer goes deeper, it would be difficult to train on GPUs efficiently due to explosion of parameters and memory usage. In addition, the deeper structure in the decoder, the harder to propagate learned feature information from the encoder. Therefore, we construct a simple generator (or decoder) that takes the semantic feature patch as input and then reconstructs a complete image in a short path. Our simple generator only contains two modules, and each is formed of (convolution, fraction-strided convolution, ELU). Then, one convolution and fraction-strided convolution is added at the end of the modules to produce the desired image. Although our generator is short and simple, we can still obtain very good image reconstruction as long as we learn high-level and semantic feature patches through the encoder.

Feature Skip Connections: In image completion problems, corrupted images and output images share a certain amount of low-level features, like prominent information of non-corrupted regions, luminance and resolutions. However, deep network based methods with bottleneck layer often lose details of images when propagating feature maps in the training stage. Moreover, one may suffer from the vanishing gradient problem as the network layer goes deeper. To shuttle the image information through the networks and reduce the training burden, we apply the skip connections strategy that is similar to [14, 22, 28]. Particularly, we only add two skip connections that just simply concatenate the channels of the output from our semantic feature extractor and that from our simple generator.

Discriminator In GANs related works, it often requires several tricks to train a generator and discriminator jointly. Otherwise, we can easily confront mode collapse and obtain unstable results. So, we follow the same discriminator architecture from [26] that outputs a

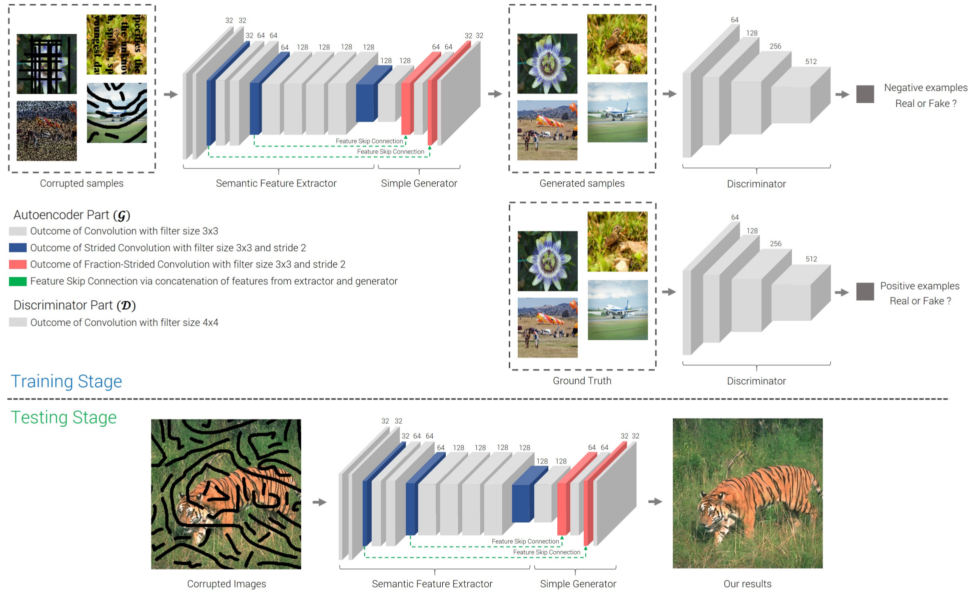

Figure 2: An overview of the proposed model. The training stage requires paired corruption masks and ground truth images from our training strategy for $\mathcal{G}$ to learn the image completion transformation, and the discriminator $\mathcal{D}$ plays a supervised role to distinguish generated samples and ground truth as real or fake. Only $\mathcal{G}$ is needed in the testing stage and real world applications.

value of realism for a given input image. In order to obtain more stable and reasonable inpainting results, we use the newly introduced adversarial loss that prevents the discriminator from crash. Details will be described in Section 2.2.

2.2 Objective Function

Our model aims to regress the corrupted image I to the ground truth image I_g. Normally, this can be achieved by using l_2 or l_1 norm (reconstruction loss) as the objective function in the deep autoencoder structures. Nowadays, many conditional GANs related researches showed that the essence of generating realistic results is to combine an adversarial loss and a reconstruction loss to the objective function. If one only minimizes the reconstruction loss (l_2 or l_1), it will encourage the networks to produce "averaged" and "smooth" results. Therefore, we apply the same way of combination of loss as the objective of our model .

For the reconstruction loss, we explored that minimizing l_1 norm can generate less blurry result than minimizing l_2 norm. The same finding is also mentioned in [14]. Hence, in our optimization of $\mathcal{G}$, we calculate the pixel-wise l_1 norm as the reconstruction loss (denoted as L_{rec}) between the generated image $\mathcal{G}(I)$ from our autoencoder model and its corresponding ground truth I_g. Note that H and W indicate the image size, and x, y represent the image coordinates:

$$\mathcal{L}_{rec} = \frac{1}{H \times W} \sum_{x \in H} \sum_{y \in W} \parallel I_g(x,y) - \mathcal{G}(I)(x,y) \parallel_1 \tag{1}$$

Similar to others, we adapt adversarial training to our deep autoencoder $\mathcal{G}$ and discriminator $\mathcal{D}$. Our $\mathcal{D}$ is dedicated to distinguish the generated images $\mathcal{G}(I)$ (fake) from ground truth image I_g (real) while our $\mathcal{G}$ is used to generate images that are real enough to deceive $\mathcal{D}$. However, conventional adversarial loss that calculates binary cross entropy (BCE) between real/fake label often makes network suffer from unstable training. To alleviate this issue, we apply the LSGAN [23] strategy that has proved substituting least square minimization for BCE can improve the stability of GAN models, and obtain more realistic results. In practice, our $\mathcal{D}$ minimizes the loss term $\mathcal{L}_{adv\_d}$ to classify ground truth I_g as the real label (given as value 1), and the generated image $\mathcal{G}(I)$ as the fake label (given as value 0):

$$\mathcal{L}_{adv\_d} = \frac{1}{2} \times \|\mathcal{D}(I_g) - 1\|_2 + \frac{1}{2} \times \|\mathcal{D}(\mathcal{G}(I)) - 0\|_2 \tag{2}$$

Meanwhile, in order to trigger $\mathcal{G}$ to generate realistic images, we also minimize the error between $\mathcal{D}(\mathcal{G}(I))$ and the real label, which provide the loss gradients when optimizing $\mathcal{G}$, which is given by

$$\mathcal{L}_{adv\_g} = \frac{1}{2} \times \|\mathcal{D}(\mathcal{G}(I)) - 1\|_2 \tag{3}$$

In summary, we define a joint loss term to optimize our proposed model. For $\mathcal{G}$ and $\mathcal{D}$, the hyper-parameters $\lambda_{\mathcal{G}}$ and $\lambda_{\mathcal{D}}$ are used to balance the loss effects to the joint optimization. The overall objective of our deep image completion model $\mathcal{G}$ is:

$$\mathcal{G}^* = \arg\min_{\mathcal{G},\mathcal{D}} \ \lambda_{\mathcal{G}}\mathcal{L}_{rec} + \lambda_{\mathcal{D}}\mathcal{L}_{adv\_g} + \lambda_{\mathcal{D}}\mathcal{L}_{adv\_d} \tag{4}$$

2.3 Training

In the training stage, there shall be a way to prompt the deep networks to capture the essential features in diverse corrupted masks in order to increase generality of deep image completion model for recovering images from various types of missing regions. In our proposed method, we first create a mask operator $\mathcal{M}$ that can produce four common types of corruptions, including text(t), line(l), scribble(s), random(r). The details about how $\mathcal{M}$ produces each type of corruptions are given as follows:

- Text(t): We create three huge masks (size 512×512) that contains massive texts and sentences, and each comes with three different font sizes: $12, 18, 24$. Then. we randomly pick a huge mask and crop a size of 128×128 region as the corruption mask.

- Line(l): We randomly determine the line width ranging from 2 to 4 pixels, and produce 4 to 8 line horizontally and vertically in a mask of size 128×128.

- Scribble(s): Similar to text, we manually draw two huge masks with large and small size of scribble. The corruption mask is generated from picking a huge mask and then cropping a size of 128×128 region.

- Random(r): We perform uniform dropout pixels to generate random corruption mask and the dropout ratio is randomly selected from 0.1 to 0.5 with 0.1 interval.

Then, we can produce the corrupted images I for training data, which can be seen as a set of processed images by using our $\mathcal{M}$ operator given ground truth images I_g and corruption indicator (t, l, s, r) as input (in **Eq.** 5). Also, we randomly apply rotation 0°, ±90°, 180° before output of $\mathcal{M}$ to increase variations of corruption masks:

$$I = \{\mathcal{M}(I_g,t),\mathcal{M}(I_g,l),\mathcal{M}(I_g,s),\mathcal{M}(I_g,r)\} \tag{5}$$

In the experiments, we will show that our model can learn the generalization to complete various kinds of corrupted images on different datasets based on this training strategy. For the testing stage, we also use the same strategy to produce corruptions masks, which means the masks are not be exactly the same as those in the training data. Please refer to our supplements for further implementation details.

3 Experimental Evaluation

We carry out experiments to evaluate the generality of different image completion methods toward diverse datasets quantitative and qualitatively. First, we describe the datasets that are used for training, validation and testing. Second, we prove that the proposed deep model takes the least training parameters among all the GANs models used for image completions, and ablation study on the results of optimizing different loss functions. Then, experimental comparisons among the proposed method and several image completion methods are shown and discussed. Finally, we demonstrate how well our model can do when dealing with high-resolution images and user-given masks that are unseen in training and validation sets. For the details of experiment settings, please refer to our supplementary materials.

3.1 Datasets

General image datasets include diverse contents and complex structures of objects from different domain. To verify the image completion performance of different methods, we evaluate on three different types of image datasets. Two of them, denoted as **(Particular)**, are featured with one particular type of objects, and the other one, denoted as **(Various)**, contains various types of objects and scenes.

- **Particular:** Caltech-UCSD Birds-200-2011[31]. It includes $11,788$ bird images with 200 categories. For the model learning, we randomly select $10,982$ images for the training ground truth, 320 for validation and 486 for testing. The appearance of categories in each set are roughly equal.

- **Various:** MSCOCO[21]. We use $82,783$ general images for the training set. Then, we randomly pick 200 images from the $40,504$ images from MSCOCO validation set, and distribute 100 images for each validation and testing set.

By applying our data augmentation strategy, the actual training images are **four times** the size of the original training data. Furthermore, in our experiments on one dataset, deep-model based methods are trained on the same data that produced from its original training set, and then pick their best model through the corresponding validation set for final testing. Note that no labels or other information related to images are used in our evaluation and all images are resized to 128×128.

	Deep-based methods			
Comparison	RED-Net[22]	CE[26]	pix2pix[14]	Ours
Adversarial training	No	Yes	Yes	Yes
Number of layers	$10(\mathcal{G})$	$12(\mathcal{G})+4(\mathcal{D})$	$16(\mathcal{G})+4(\mathcal{D})$	$17(\mathcal{G})+4(\mathcal{D})$
Required training parameters	0.3M	71M	57M	1.1M

Table 1: Model comparison on different deep-based methods. The numbers inside the brackets stand for the numbers of layers in its generator $\mathcal{G}$ or discriminator $\mathcal{D}$.

3.2 Comparisons and Results

We demonstrate the quantitative and qualitative results of previous and ours image completion methods on different datasets with four different testing corruptions (**text**, **line**, **scribble**, **random**). Previous methods can be roughly categorized into the vision-based[8, 12, 17, 29] and deep-based[14, 22, 26] approaches. Due to the limited space, please refer to our supplemental materials for details and full comparison of the methods on different datasets.

Model Comparison: We compare our deep models with others in Table 1. Even though adapting parts of VGG[30] can lead to increasing layers, we only require nearly 1.5% training parameters compare to two deep GANs (CE[26] and pix2pix[14]). Although RED-Net[22] has the smallest model size, the model without adversarial training gives unpleasant inpainting results (shown in the later section).

Ablation Study of Loss Functions: We give analysis of loss functions to check the contribution of different loss terms and the improvements using least square (LS) adversarial loss term. We compare model trained on L1 Loss ($\mathcal{L}_{rec}$), L1 + BCE ($\mathcal{L}_{rec}+\mathcal{L}_{adv(bce)}$), L1 + LS ($\mathcal{L}_{rec}+\mathcal{L}_{adv(ls)}$). Table 2 shows the evaluation of PSNR/SSIM on four different types of corruptions from CUB testing dataset. With the adversarial training like L1+BCE and L1+LS, the completed results can be more realistic and yield higher PSNR/SSIM. Moreover, we can find that optimizing the proposed loss function (L1+LS) is better than the convention loss (L1+BCE) in conditional GANs. Although these two optimizations are trying to generate results that not only resemble to the ground truth but also are as real as possible, our proposed model (L1+LS) improves the unstable results in (L1+BCE) and our results are closer to the ground truth pixel-wise and structurally.

Quantitative Results: Table 3 shows testing quantitative results of all models on CUB[31] and MSCOCO[21] with different corruptions types. We use common metrics PSNR and SSIM to evaluate image quality. No matter which datasets are used, our model is able to generate competitive results over other methods, especially outperforming the previous

	Optimizing different loss functions in our model		
Testing Types	Ours ($\mathcal{L}_{rec}$ only)	Ours ($\mathcal{L}_{rec}+\mathcal{L}_{adv(bce)}$)	Ours ($\mathcal{L}_{rec}+\mathcal{L}_{adv(ls)}$)
Text	31.84/0.945	32.17/0.951	**32.34/0.952**
Line	30.32/0.924	30.38/0.926	**30.62/0.928**
Scribble	28.83/0.913	28.59/0.913	**29.01/0.918**
Random	33.75/0.952	34.98/0.964	**35.15/0.965**
Average	31.19/0.933	31.53/0.938	**31.78/0.941**

Table 2: Quantitative Results under different loss functions of our model on CUB [31] testing dataset and various corruptions.

	Algorithms							
Datasets, types	CSH[17]	TNNR[12]	FoE[29]	nans[8]	RED-Net[22]	CE[26]	pix2pix[14]	Ours
CUB[31], Text	23.82/0.842	25.90/0.850	31.50/0.953	31.70/0.954	24.38/0.776	26.57/0.836	25.40/0.853	**32.25/0.951**
CUB[31], Line	22.43/0.806	15.56/0.604	27.96/0.906	29.74/0.927	23.21/0.747	26.37/0.827	26.52/0.825	**30.53/0.928**
CUB[31], Scribble	20.81/0.761	21.51/0.791	26.92/0.899	27.98/0.914	23.11/0.775	25.72/0.830	24.63/0.820	**29.11/0.919**
CUB[31], Random	23.08/0.684	32.05/0.905	35.27/0.974	**36.24/0.975**	21.04/0.546	26.46/0.777	23.40/0.752	35.18/0.964
Average	22.54/0.774	23.76/0.787	30.41/0.933	31.42/**0.942**	22.94/0.711	26.28/0.818	24.99/0.812	**31.78**/0.940
MSCOCO[21], Text	22.79/0.843	24.85/0.842	29.09/0.940	29.27/0.937	22.19/0.734	25.50/0.830	23.49/0.763	**30.75/0.944**
MSCOCO[21], Line	20.66/0.788	15.04/0.610	24.95/0.874	27.18/0.898	21.13/0.703	24.81/0.811	22.19/0.723	**28.34/0.909**
MSCOCO[21], Scribble	20.67/0.765	20.78/0.788	25.12/0.886	26.28/0.896	20.65/0.727	24.48/0.825	22.31/0.745	**27.77/0.911**
MSCOCO[21], Random	22.22/0.666	30.22/0.885	32.32/**0.962**	33.60/0.961	21.88/0.612	25.96/0.776	22.23/0.691	**34.09**/0.959
Average	21.43/0.765	22.72/0.781	27.87/0.916	29.08/0.923	21.46/0.694	25.19/0.811	22.56/0.730	**30.24/0.931**

Table 3: Quantitative results from vision-based and deep-based methods on different types of datasets and corruptions. The higher the (PSNR/SSIM) are, the closer the completed images are compared to the ground truth images. Bold and under-line indicate the best and the second best performance, respectively.

deep-based methods. For the experiments on CUB[31], our model may not consistently provide better PSNR or SSIM compared to FoE[29] and inpaint_nans[8]. This is because a general image featuring one object contains less complex structures, it can be easier for those two vision-based methods to find a proper and coherent content distribution from the whole image or neighboring patches to describe the corrupted regions. This circumstance happens especially in random corruption type where ones can simply take surrounding non-corrupted pixels into account to fill the corruption part. For more complex images from MSCOCO[21], our models can provide the best results among all methods. In summary, this quantitative results demonstrate that the proposed model achieves the best performance in average, and our training strategy improve the generalization of the trained deep model that can perform image completion on various types of corruptions for different complex scenes.

Qualitative Results: According to the quantitative evaluations, we compare the **top two** best-performing methods in vision-based and deep-based image completion approaches on MSCOCO[21] dataset with four different types of corruptions in Figure 3. Our model can recover these types of missing regions more semantically while others contain blurry regions and seams in the filled part. When the corrupted regions are bigger and longer, the flaws in the inpainting results become more obvious, such as in the types of scribble and lines. As mentioned before, despite the fact that one can produce very nice results dealing with random corruptions, and also provide very high PSNR/SSIM such that people can not easily perceive the differences between them, our method can generate more clear results than other deep learning methods.

3.3 Performances Beyond Model

In this section, we challenge the performances and extensions of image completion methods and models that are trained on MSCOCO[21] datasets using our training strategy. We first show that our method can be applied on higher resolution testing sets. Moreover, to test the capability and generalization of our model, we show some results of recovering user-defined corrupted images.

Higher-resolution evaluation: We test different methods on 200 images from BSDS500 test set[24], which is composed of general images containing complicated objects, and they are unrelated to any training and validation set in MSCOCO[21]. The images are resized to resolutions 320 × 320, which is larger than those in our training data and previous experi-

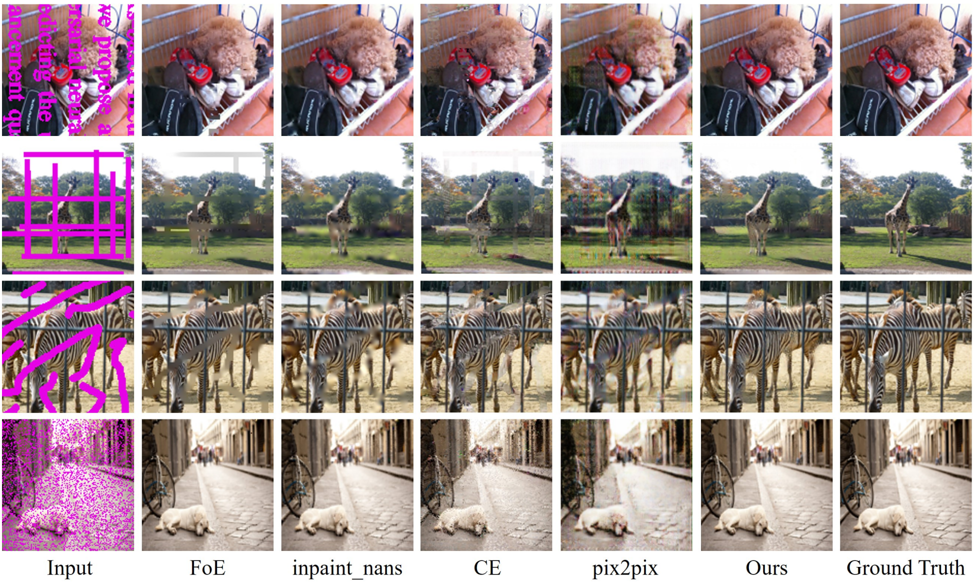

Figure 3: Qualitative results on MSCOCO[21]. Images from top to the bottom represent inpainting results for four corruption types (**text**, **line**, **scribble**, **random**). For visualization, purple regions indicate corrupted parts.

ments. Table 4 reveals that we also outperform other vision-based and deep-based methods in PSNR and SSIM for the four types of corruptions. There are qualitative results shown in Figure 4 comparing our model and the second best method in vision-based and deep-based approaches. We clearly generate more reasonable results even on higher resolution images, and the details can be semantically reconstructed back from corruptions.

Results from user specified masks: In order to further examine the general performance for our model, we invite subjects to manually create interesting corrupted mask on public high-resolution images, which means the model was not trained on these materials. One teaser qualitative result is shown in Figure 1. Even though the input images contain complex corrupted regions, our method can still well recover such images, which indicates that the proposed method has the potential for real-world applications. For more results, please check our supplemental materials.

	Algorithms							
Datasets, types	CSH[17]	TNNR[12]	FoE[29]	nans[8]	RED-Net[22]	CE[26]	pix2pix[14]	Ours
BSDS500[24], Text	24.62/0.869	27.67/0.863	30.56/0.941	30.73/0.939	22.84/0.721	25.74/0.699	20.98/0.618	**32.12/0.946**
BSDS500[24], Line	23.91/0.835	16.58/0.663	26.61/0.898	28.83/0.915	22.02/0.726	25.74/0.683	20.21/0.608	**30.03/0.924**
BSDS500[24], Scribble	22.34/0.795	24.59/0.820	26.78/0.894	27.62/0.901	20.89/0.705	23.89/0.678	21.14/0.630	**29.18/0.913**
BSDS500[24], Random	22.40/0.569	31.38/0.885	33.32/0.959	34.79/**0.961**	21.08/0.532	26.48/0.773	19.13/0.477	**35.47**/0.960
Average	23.32/0.767	25.05/0.808	29.32/0.923	30.49/0.929	21.71/0.670	24.68/0.708	20.37/0.583	**31.70/0.935**

Table 4: Quantitative results (PSNR/SSIM) from the vision-based and deep-based methods on different types of corruptions for BSDS500 datasets[24].

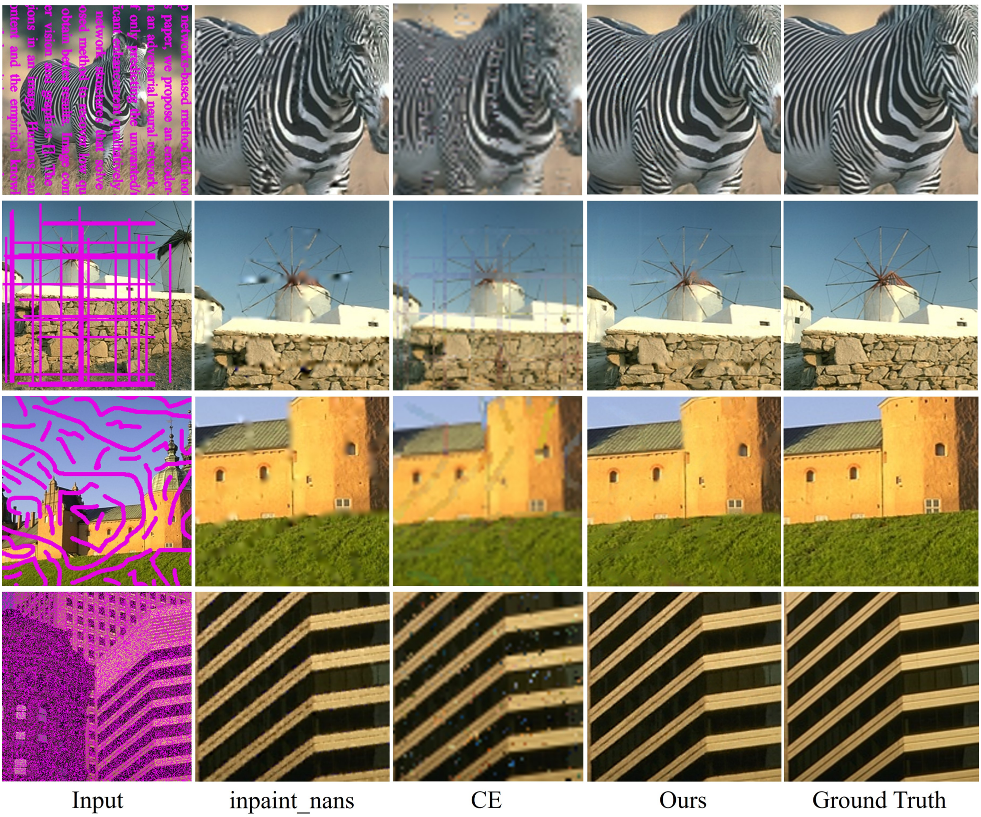

Figure 4: Qualitative results on BSDS500[24]. Images from top to the bottom represent inpainting results on four corruption types (**text**, **line**, **scribble**, **random**). Corrupted part is presented in purple color for visualization.

4 Conclusion

In this work, we show that our deep completion model and the proposed training strategy can provide superior image completion performance quantitatively and qualitatively on different datasets. Inspired by the existing conditional GANs and steady adversarial training techniques, the proposed lightweight deep networks can successfully generate stable and semantic image completion results and outperform previous methods. Besides, we also reveal the potential of using the proposed model on high-resolution images for real world applications. In the future, we will further enhance the generalization of our deep model to recover a wide-variety of image corruptions in practice.

References

[1] BMVC 2017. Retrieved from https://bmvc2017.london/, April 29, 2017.

[2] ARCHITECTURURAL DIGEST, London Travel Guide. Retrieved from http://www.architecturaldigest.com/london-travel-guide, April 29, 2017.

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.

[6] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[8] John D'Errico. inpaint_nans, MATLAB Central File Exchange, Retrieved March 1, 2017.

[9] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM Transactions on graphics (TOG)*, volume 22, pages 303–312. ACM, 2003.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] James Hays and Alexei A Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007.

[12] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2117–2130, 2013.

[13] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. *ACM Transactions on Graphics (TOG)*, 33(4):129, 2014.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[16] Nikos Komodakis and Georgios Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007.

[17] Simon Korman and Shai Avidan. Coherency sensitive hashing. In *2011 International Conference on Computer Vision*, pages 1607–1614. IEEE, 2011.

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[20] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[22] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Proc. Advances in Neural Inf. Process. Syst.*, 2016.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[25] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[29] Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

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

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[32] Marta Wilczkowiak, Gabriel J Brostow, Ben Tordoff, and Roberto Cipolla. Hole filling through photomontage. In *BMVC*, volume 5, pages 492–501, 2005.

[33] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.