Image Inpainting using Block-wise Procedural Training with Annealed Adversarial Counterpart

Anonymous ECCV submission

Paper ID ***

Abstract. Recent advances in deep generative models have shown promising potential in predicting missing pixel values in an image using surrounding context. However, such models are either slow or fail to generate large hole contents. We present a new method for synthesizing high-quality photo-realistic inpaintings from incomplete images using conditional generative adversarial networks (conditional GANs). In particular, we introduce a block-wise training scheme, in conjunction with an annealed adversarial loss, to stabilize the training process of a very deep generative network. We also discuss the effectiveness of a novel perceptual similarity metric as an additional loss. Furthermore, we extend our framework to various inpainting scenarios, including object removal, image harmonization and guided inpainting. Extensive experiments and user-studies show that our method significantly outperforms existing methods in all these tasks.

Keywords: Image translation, image inpainting, image harmonization and image composition.

1 Introduction

Image inpainting is the task to fill in the missing part of an image with visually plausible contents. It is one of the most typical operations of image editing [1] and low-level computer visions [2, 3]. The goal of image inpainting is to create semantically plausible contents with rich texture details, which can either be consistent with the original contents or is coherent with the known context such that the output image appears realistic. Other than image restoring and fixing, inpainting can also be used to remove unwanted objects, or in the case of guided inpainting, it can be used to composite with the contents from another image. In the latter scenario, we often need harmonization to adjust the appearance of the guidance image to make it compatible with the known context. Meanwhile, inpainting is needed to fill in the gaps between the two images.

Traditional image inpainting methods mostly develop texture synthesis techniques to address the problem of hole-filling [4, 2, 5–8]. In [6], Barnes et al. proposes the Patch-Match algorithm which efficiently searches for the most similar patch to reconstruct the missing regions. Wilczkowiak et al. [8] takes further steps and detects desirable search regions to find better match patches. However,

these methods only exploit the low-level signal of the known contexts to hallucinate missing regions and fall short of understanding and predicting high-level semantics. Furthermore, it is often challenging to capture the global structure of images by simply extending texture from surrounding regions. Another line of work for inpainting aims to fill in holes with content from another guidance image, by using composition and harmonization [3, 9]. The guidance image is often retrieved from a large database of images before it is pasted blended with the original image. Although these methods are able to propagate high-frequency details from the guidance image, they often introduce inconsistent regions and gaps which are easily detectable with human eyes.

More recently, deep neural networks have exhibited excellent performance in various computer vision tasks, including texture synthesis and image completion. In particular, adversarial training becomes the de facto strategy to train an image inpainting model [10–14]. Pathak et al. [10] first proposes to train an encoder-decoder model to synthesize missing holes from surrounding pixels, using both the reconstruction loss and the adversarial loss. In [11], Yeh et al. addresses inpainting by using a pre-trained model to find the most similar encoding of the corrupted image. Yang et al. [13] proposes a multi-scale neural patch synthesis approach, which optimizes the hole contents such that its feature extracted from middle layers of a pre-trained CNN matches with the features of the surrounding context. The optimization greatly improves the inpainting quality and resolution at the cost of computational efficiency. Iizuka et al. [14] instead proposes a pure feed-forward model trained with global and local GANs, and generates excellent results for small holes. However, DNN based methods have several limitations. First, they are either too slow due to optimization [13] or cannot generate sufficient high-frequency details, especially for large holes [14]. Second, it is difficult to handle perceptual continuity, making it necessary to resort to post-processing (e.g. Poisson blending for [14]) to smooth out the coalescing regions.

In practice, we found that directly training a very deep generative network to synthesize high-frequency details is difficult. Most often we fail to stabilize the training process and the results are either overly smooth or containing significant noise and artifacts. To overcome this limitation, we discuss a new approach that produces high-quality inpainting results for various inpainting tasks, refer to as **B**lock-wise **T**rained **G**enerative **M**odel for **I**npainting (BTGMI). More specifically, we decompose the generator into a ResNet head followed by multiple refinement residual blocks. For the first phase, we train the **ResNet head** for inpainting until it converges. Then we add residual blocks one at a time. Each time we train with an additional block, we use skip connections to initialize with the trained network and gradually increase the weight of the new block. This introduces the new block gradually, forcing it to refine from the previous results and learn to generate richer details. In addition, we observe that it is essential to steadily reduce the weight of the generator adversarial loss during block-wise refinement. We refer to this training scheme as **A**dversarial **L**oss **A**nnealing (ALA). Intuitively, AAL is helpful as the adversarial loss usually dominates in the end phase of training and the generator will falsely create noise patterns to foul

the discriminator. Finally, Zhang et al. [15] shows that the perceptual similarity, measured by internal activations of networks trained for high-level classification tasks, corresponds to human perceptual judgment far better than commonly used metrics such as the Euclidean distance. Inspired by this finding, we introduce a novel **P**atch **P**erceptual **L**oss (PPL), which penalize the perceptual difference between the inpainted patch and the original patch. Different from the perceptual losses ,used in style transfer [16, 17] and image synthesis [18, 19], we compute the feature disparity across all layers. In our experiment, we found that PPL works better than the reconstruction loss and the general perceptual loss.

To evaluate the proposed inpainting approach, we conduct extensive experiments on different datasets. We also show that our model, although being designed for inpainting, can be used for general image translation tasks including image harmonization and composition. This enables us to jointly train inpainting with those tasks and makes it suitable for a wide range of inpainting scenarios such as object removal and guided inpainting. As shown by visual results and user-study, our network already outperforms state-of-the-art inpainting and harmonization methods without using block-wise refinement. By leveraging block-wise training, it can further add high-frequency details and eliminate the perceptual discontinuity. Finally, we demonstrate our approach is both effective and simple to use in multiple real use cases.

In summary, in this paper we present:

1. The Block-wise Trained Generative Model for Inpainting (BTGMI) as a novel, end-to-end model that generates state-of-the-art image inpainting and image composition results.
2. The Adversarial Loss Annealing (ALA) and the Patch Perceptual Loss (PPL) as a novel training scheme and training loss that improve the quality of inpainting.
3. An easy-to-use user interface for real use cases of distractor removal and guided inpainting, where the users could conveniently select objects from either the original image (for removal) or from the guided image (for composition).

2 Related Work

Deep image generation and manipulation Generative Adversarial Network (GAN) [20] uses a mini-max two-player game to alternatively train a generator and a discriminator, and has shown impressive capacity to generate natural and high-quality images. However, for the vanilla GANs, the training instability makes it hard to scale to higher resolution images. Several techniques have been proposed to stabilize the training process, including Laplacian pyramid GAN [21], DCGAN [22], energy-based GAN [23], Wasserstein GAN (WGAN) [24], WGAN-GP [25] and the Progressive GAN [26]. Both BTGMI and Progressive GAN gradually increase the depth of the network during training. While Progressive GAN addresses the image synthesis problem, our architecture poses as an ideal candidate for image translation tasks. A major model difference

between BTGMI and Progressive GAN is that, rather than bringing in convolutional layers at the end of the network, we progressively insert residual blocks before the upsampling layers.

Adversarial training, as a general idea for DNN based methods, has been widely applied to various research fields, especially many image editing tasks such as image super-resolution [27–29], image-to-image translation [30, 31], image inpainting and image harmonization. Recently, [32] proposes the Pix2Pix HD model for high-resolution image synthesis using conditional GANs, which produces very high-quality simulated images from semantic maps, human sketches, etc. Our ResNet head simplifies the Pix2Pix HD model and also improves the synthesis quality by re-designing its losses. For the image inpainting task, many DNN based approaches achieve good performance by different network topology and training procedure [10, 13, 11, 14]. Image harmonization, on the other hand, aims to adjust the appearances of the foreground and background regions such that they are compatible and the composition is realistic. In [33], Zhu et al. trains a CNN model to measure how realistic of a composite image is, and uses this metric to adjust and optimize the appearance of the foreground region. Tsai et al. [9] also proposes a DNN based method by training a deep CNN to learn and predict the context and semantic information of composite images. However, the limitation of image harmonization alone is that low-level appearance or color adjustments are often inadequate to make the composition realistic. In contrast, our approach of guided inpainting not only adjusts the appearance of the guidance patch, but also synthesizes new contents to fill in the gaps and smoothes the transition between the foreground and the background.

Non-neural image inpainting and harmonization Traditional image completion algorithms can be either diffusion-based [4, 34] or patch-based [7, 6]. Diffusion-based methods usually cannot synthesize plausible contents for large holes or textures, due to the fact that it only propagates low-level features. Patch-based methods, however, largely rely on the assumption that the desired patches exist in the database. For harmonization, traditional methods usually apply color and tone matching, by matching global statistics [35] or multi-scale statistics [36], extracting gradient domain information [37, 38], or utilizing semantic clues [39]. [40] further develops a data-driven method, which searches and retrieves multiple real images with similar structural layouts and use them to transfer the appearances. A complete comparison with non-neural inpainting and harmonization algorithms is beyond the scope of our paper.

3 Our Method

In this section, we describe our model and several training schemes. First, we illustrate the details of our basic components: the Generator head and the training losses in Sec. 3.1. Then, we characterize our block-wise procedural training scheme together with the adversarial loss weight annealing in Sec. 3.3. Finally, we summarize our implementation and training details.

3.1 The Generator Head

Our generator head is a conditional GAN network [41], which takes an incomplete image as the input, and outputs a complete image. Conditional GANs for image inpainting usually consist of a generator G and a discriminator D. The generator G learns to predict the hole contents and restore the complete image, while the discriminator D learns to distinguish real images from the generated ones. The model is trained in a self-supervised manner via the following minimax game:

$$\min_G \max_D E_{(s,x)}[\log D(s,x)] + E_s[\log(1 - D(s, G(s)))], \quad (1)$$

where s and x are the incomplete image and the original image respectively, and $G(s)$ is the generator prediction given the input s. Note that if $G(s)$ predicts an entire image as output, we only keep the hole contents and concatenate with the known context of s.

Previous research experimented with different architecture of G, most notably the U-Net style generator of [10] and the FCN style generator of [14]. [14] shows that FCN style inpainting network produces less blurred results than U-Net, mainly because instead of using the fully connected layer as a bottleneck, it only uses fully convolutional layers which avoids significant resolution reduction or information loss.

Similar to [14], our generator head is based on FCN and leverage the properties of convolutional neural networks, including translation invariance and parameter sharing. Nevertheless, a major limitation of FCN is the constraint of the receptive field size, since the convolution layers are locally connected, making pixels far away from the hole carry no influence on the predicted hole content. We rely on several strategies to alleviate such drawback. First, like [14], we use a down-sampling front end to reduce the feature size, followed by multiple ResNet blocks, as well as an up-sampling back end to restore the full dimension. By downsampling, we increase the receptive field of the ResNet blocks. Second, we stack multiple ResNet blocks to further enlarge the receptive field. Finally, we adopt the dilated convolutional layers [42] in all ResNet blocks, with the dilation factor set to 2. Dilated convolutions use spaced kernels, making it compute each output value with a larger input coverage, without increasing the number of parameters and computational power. Overall, as context is critical for realism, we observed that the receptive size poses as an important role for image inpainting, which also differentiates it from other image translation tasks.

More specifically, the down-sampling front-end consists of three convolutional layers each with stride 2, and the intermediate residual blocks contain 9 blocks stacked together, and the up-sampling back-end consists of three transposed convolution of stride 2. Each convolutional layer is followed by batch normalization (BN) and ReLu as the activation layer, except for the last layer which outputs the image. For down-sampling and up-sampling, an alternative would be to use interpolated convolution to reduce the checkerboard effect, as suggested by [43]. Interpolated convolution uses a dimension-preserving convolution layer of stride 1, followed by max pooling or bilinear up-sampling. However, we observed that

using interpolated convolution creates overly smooth effects. A detailed ablation study is presented in Sec. 5.

The detailed architecture of our generator head is illustrated in Fig..

3.2 The Training Losses

Different losses have been used to train an inpainting network. These losses can be cast into two categories. The first category, which we refer to as *similarity loss*, is used to measure the similarity between the output and the original image. The second category, which we refer to as the *realism loss*, is used to measure how realistic-looking the output image is. We summarize the losses used in different approaches in Table 1.

Method	Similarity Loss	Realism Loss
Context Encoder [10]	ℓ_2	Global Adversarial Loss
Global Local GAN [14]	ℓ_2	Global and Local Adversarial Loss
Our Approach	Patch Perceptual Loss (PPL)	Improved Multi-Scale Adversarial Loss

Table 1. Comparison of training losses in different methods.

Patch Perceptual Loss As shown in Table 1, using ℓ_2 loss for reconstruction and measure the disparity between the output and the original image has been a default choice of previous inpainting methods. However, it is known that ℓ_2 loss does not correspond well to human perception of visual similarity [15]. This is because ℓ_2 losses wrongly assumes each output pixel is conditionally independent of all others. A well-known issue, for example, is that blurring an image leads to small changes in terms of Euclidean distance but causes significant perceptual difference. Recent research suggests that a better metric for perceptual similarity is the internal activations of deep convolutional networks, usually trained on a high-level image classification task. Such loss is called "perceptual loss", and is used in various tasks such as neural style transfer [17], image super-resolution [16], and conditional image synthesis [18, 19].

Based on this observation, we propose a new "patch perceptual loss" as the substitute of the ℓ_2 losses. Traditional perceptual loss typically uses VGG-Net, and computes the ℓ_2 distance of the activations on a few feature layers. Recently, [44] specifically trained a patch perceptual network to measure the perceptual differences between two image patches based on AlexNet, making it an ideal candidate for our task. The patch perceptual network computes the activations across all feature layers and sums up the ℓ_2 distances scaled by learned weights at each layer. Furthermore, to take into account both the local view and the global view of perceptual similarity, we compute PPL at two scales. Local PPL considers the local hole patch, while the global PPL slightly zooms out to cover a larger contextual area. More formally, our PPL is defined as:

$$\sum_{k=1,2} PPL_k(G(s)_p, x_p) = \sum_{k=1,2} \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \| w_l^T \odot (\hat{F}(x_p)^l_{hw} - \hat{F}(G(s))^k_{hw}) \|_2^2 \quad (2)$$

Here k refers to the patch scale. p is the patch covering the hole. $\hat{F}$ is the AlexNet and l is the feature layer. Sec. 5 shows that PPL gives better inpainting quality than both ℓ_2 and VGG-based perceptual losses.

Multi-Scale Patch-wise Adversarial Loss Adversarial losses are given by trained discriminators to discern whether an image is real or fake. The global adversarial loss of [10] takes the entire image as input and outputs a single real/fake prediction, which does not consider the realism of local patches, especially the holes. The additional local adversarial loss of [14] adds another discriminator specifically for the hole, but it requires the hole to be fixed shape and size during training to fit the discriminator. To consider both the global view and the local view, we propose to use discriminators at three scales of image resolutions. The discriminator at each scale is identical, only the input is a scaled version of the *entire image*. Furthermore, we use PatchGAN discriminator at each scale, which uses convolutional discriminator to output a vector of predictions, each value corresponds to an image patch. In this way, the discriminators are trained to classify global and local image patches across the image, which enables us to use random holes and shapes during training. However, directly using PatchGAN is problematic in our case, as for the output restored image, only the patches overlapping with the hole area should be considered fake patches. Therefore when computing the discriminator loss of the restored image, instead of forcing the entire output to be fake, only the patches overlapping with the holes are labeled as fake. More formally, our Patch-wise Adversarial Loss is defined as:

$$\min_{G} \max_{D_1,D_2,D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k) = \sum_{k=1,2,3} E_{(s,x)}[\log(s, x)] + E_s[\log(Q - D_k(s, G(s))] \quad (3)$$

Here Q is a patch-wise real/fake vector, based on whether the patch overlaps with the holes. Using multiple GAN discriminators at the same or different image scale has been proposed in unconditional GANs [45] and conditional GANs [32]. Here we extend the design to take into account inpainting hole locations, which is critical in obtaining semantically and locally coherent image completion results.

Our full objective combines both losses is therefore defined as:

$$\min_{G}((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k)) + \lambda_{PPL} \sum_{k=1,2} PPL_k(G(s)_p, x_p)) \quad (4)$$

where λ controls the importance of the two terms. We set $\lambda_{PPL} = 10$ in our experiments.

3.3 Procedural Training with Adversarial Loss Annealing

Our experiments show that using the described generator head and training losses already give inpainting results better than state-of-the-art. However, the results can still lack fine details, especially when synthesizing textures. There could also be noise patterns when the image is complicated. A straight-forward effort to improve the results would be to stack more intermediate residual blocks to further expand the receptive view, and also increase the expressiveness of the

model. However, we found that directly stacking more residual blocks makes it more difficult to stabilize the training. As the search space becomes much larger, it is also more challenging to find local optimum. In the end, the inpainting quality deteriorates as the model depth increases.

Recently, Progressive GAN [26] was proposed as a new training methodology for generative adversarial networks. The key idea is to grow both the generator and discriminator progressively, by adding new layers that model increasingly fine details as training progresses. This strategy makes the training faster and more stable and enables it to synthesize mega-pixel images with unprecedented visual quality.

We adapt this idea for image inpanting and propose to use procedural block-wise training to gradually increase the depth of the inpainting network. More specifically, we begin by training the generator head until it converges. Then, we add a new residual block after the already trained residual blocks, right before the back-end upsampling layers. In order to smoothly introduce the new residual block without suddenly breaking the trained model, we add another skip path from the trained residual blocks to the upsampling layers. Initially, the weight of the skip path is set to 1, while the weight of the path containing the new block is set to 0. This essentially makes the initial network identical to the already trained network. We then slowly decrease the weight of the skip path and increase the weight of the new residual block as training progresses. In this way, the newly introduced residual block is trained to be a fine-tuning component, which adds another layer of fine details to the original results. This step are repeated multiple times, where each time we expand the model by adding a new residual block. In our experiment, we found that the results improve significantly after fine-tuning with the first additional residual block. The output becomes stabilized after three residual blocks, and no discernible changes can be detected if more residual blocks are introduced. The procedural training process is illustrated in Fig..

We observe that the block-wise procedural training has several benefits. First, it guides the training process of a very deep generator. Starting with the generator head and gradually fine-tuning with more residual blocks makes it easier to discover the mapping between the incomplete image and the complete image, even though the search space is large given the diversity of natural images and the random holes. Another benefit is reduced training time, as we found decoupling the training of the generator head and the fine-tuning of additional residual blocks requires significantly less training time comparing with training the network all at once.

Adversarial Loss Annealing During training, the generator adversarial loss updates the generator weight if the discriminator successfully detects the generated image as fake:

$$\sum_{k=1,2,3} E_s[\log(\bar{Q} - D_k(s, G(s)))].$$

Note that here $\bar{Q}$ reverses Q of 3.2. As shown in Figure., we observe that the generator adversarial loss becomes dominant over PPL as training progresses. This

is because the discriminator becomes increasingly good at detecting fake images during training. This is less of a problem for image synthesis tasks. However, for our inpainting task, the outcome is that the generator deliberately adds noise patterns to confuse the discriminators, bringing more artifact to output or even synthesizing wrong textures. Based on this observation, we propose to use adversarial loss annealing, which decreases the weight of the generator adversarial loss when adding new residual block. More formally, let the initial weight of the generator adversarial loss be $\lambda^0_{G_{adv}}$, and the weight of the generator adversarial loss be $\lambda^i_{G_{adv}}$ after adding the i_{th} residual block. We found that simply decay the weight linearly by setting $\lambda^i_{G_{adv}} = 0.1^i \lambda^0_{G_{adv}}$ reduces the noise level. Detailed results are shown in Sec. 5.

We summarize the discussed training schemes in Alg. 1.

Algorithm 1 Training the Inpainting Network

1: Set $\lambda_{PPL} \leftarrow 10$ and $\lambda^0_{G_{adv}}$
2: Train the Generator Head G_0 until converges.
3: **for** i=1 to 3 **do**
4: Set $\lambda^i_{G_{adv}} = 0.1^i \lambda^0_{G_{adv}}$
5: Add the skip path and the residual block r_i
6: Train the new generator G_i until it converges.
7: **return** G_3

4 Results

In this section, we first describe our dataset and experiment setting (Sec. 4.1). We then provide quantitative and qualitative comparisons with several methods, and also report a subjective human perceptual test with user-study (Sec. 4.2). In Sec. 4.3, we conduct several ablation study about the design choice of the models, losses, and training scheme. Finally, we show how our method can be applied to real use cases of object removal and guided inpanting. In particular, the inpainting model can be adapted to train an image harmonization network, which generates state-of-the-art harmonization results. We then demonstrate that by jointly training a model for inpainting and harmonization we can easily achieve guided inpainting (Sec. 4.4).

4.1 Experiment Setup

We evaluate our inpainting method on several representative datasets. COCO [46] is a large dataset containing both object and scene images; Place2 [47] includes images of a diversity of scenes and was originally meant for scene classification; finally, we train and test on CelebA [48], which consists of 202,599 face images with various viewpoints and expressions. For a fair comparison, we follow the

standard split with 162,770 images for training, 19,867 for validation and 19,962 for testing.

In order to compare with existing methods, we train on images of size 256x256. We also train another network at a larger scale of 512x512 to demonstrate its ability to handle higher resolutions. As the pre-processing step, we first resize the image and then conduct random cropping. We then apply data augmentation with random flipping. For each image, we create a mask containing one or two rectangle holes. The size of the hole ranges from 1/4 to 1/2 of the image's dimension, and are positioned at random locations. Note that during inference, our network is able to handle masks with arbitrary number of holes of any shape. Finally, we shift and rescale the pixel value from [0,255] to [-1,1] and fill in the masked regions with zeros. We then concatenate the corrupted image and the mask as input.

For all our training, we set the learning rate with polynomial decay starting from 0.0002, and adopt Adam for optimization. We set the batch size to 8, and regardless of the actual dataset size, we train 150,000 iterations for the generator head and another 1,500 iterations for each additional residual block. For each dataset, the training takes around 2 days to finish on a single Titan X GPU.

4.2 Comparison with Existing Methods

We compare our results with Content-Aware Fill (CAF) [6], Context Encoder (CE) [10] and Global-Local Inpainting (GLI) [14]. For CE and GLI, we use off-the-shelf pre-trained model from the web. Given CE's model is trained with square holes located at the center, we evaluate on both settings of arbitrary holes (Fig.) and center hole (Fig.). For center hole completion, we compare with CAF, CE and GLI on ImageNet test images [49], where our results are directly generated by models trained on COCO. For arbitrary shape completion, we compare with CAF and GLI using images from COCO, Place2 and CelebA. As GLI applies Poisson Blending for post-processing, we show both their results before and after post-processing. For our approach, we show the results generated by the Generator Head alone as well as the final results using procedural training as refinement. The comparison results shown are randomly sampled from the entire test set.

Based on the visual results we can see that, for CAF as a non-learning and patch-based approach, its main issue is the inability to generate novel objects not available in the known context. This is especially an issue for highly specific and complex structure such as face inpainting. Furthermore, while CAF is able to generate realistic-looking details, they do not always capture the global structure and the inpainting is often inconsistent when the contexts are complex. CE's result is blurrier, and the border between the hole and the context is easily detectable. Comparing our approach with GLI, even without fine-tuning, our results exhibit better textures and less noise, and are visually more coherent with the surrounding context. This is especially true for large holes.

Quantitative Evaluation Table. shows quantitative comparison between CAF, CE, GLI and our approach. The values are computed based a random subset

of 200 images selected from the test set. We can see that our method performs better than other methods. In addition, our procedural training further reduces the losses from the generator head.

User study

4.3 Ablation Study

Failure Cases

4.4 A user interface of guided inpainting

For guided inpainting, the data acquistion is a bit two different. We randomly crop a bounding box containing the object from the original image, and we then keep the object only and discard the background. We then randomly select another image from dataset, and do color transfer using []'s algorithm. After that we paste the object onto the randomly selected image and then crops a bounding box, and paste it back to the original image. Assuming we know the segmentation, we have two masks as input, one is the background inside the bounding box for inpainting, and the other image the object mask for harmonization. We use the same network and losses as unguided inpaiting, only this time we output two images, one for the inpainting result, and the other for harmonization result. The final result is a composition of the inpainting result and harmonization result.

We use a paste the object (without the hole) onto another randomly select it object object it into another image. We then use color transfer to transfer the color of the object.

COCO (?) and Place2 and CelebA dataset (?). COCO contains ? training images and ? test images. Place2 contains ? and ? test images. The place We train and test on four datasets, the COCO, ADE20K, CelebA, and Place.

For guided inpainting, we trained on COCO dataset. Our data acquisation is as following. The input is an image a.

Figure a.

5 Conclusions

References

1. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Advances in Neural Information Processing Systems. (2015) 262–270
2. Komodakis, N.: Image completion using global optimization. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 442–452
3. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: ACM Transactions on Graphics (TOG). Volume 26., ACM (2007) 4
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (2000) 417–424
5. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Volume 1., IEEE (2004) I–I
6. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics-TOG **28**(3) (2009) 24
7. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE transactions on image processing **12**(8) (2003) 882–889
8. Wilczkowiak, M., Brostow, G.J., Tordoff, B., Cipolla, R.: Hole filling through photomontage. In: BMVC 2005-Proceedings of the British Machine Vision Conference 2005. (2005)
9. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Lu, X., Yang, M.H.: Deep image harmonization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
10. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2536–2544
11. Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint arXiv:1607.07539 (2016)
12. Li, Y., Liu, S., Yang, J., Yang, M.H.: Generative face completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2017) 6
13. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2017) 3
14. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**(4) (2017) 107
15. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. arXiv preprint arXiv:1801.03924 (2018)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, Springer (2016) 694–711
17. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, IEEE (2016) 2414–2423

18. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems. (2016) 658–666
19. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: The IEEE International Conference on Computer Vision (ICCV). Volume 1. (2017)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
21. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in neural information processing systems. (2015) 1486–1494
22. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
23. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)
24. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
25. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028 (2017)
26. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
27. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1646–1654
28. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision, Springer (2014) 184–199
29. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802 (2016)
30. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016)
31. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)
32. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. arXiv preprint arXiv:1711.11585 (2017)
33. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Learning a discriminative model for the perception of realism in composite images. In: Computer Vision (ICCV), 2015 IEEE International Conference on. (2015)
34. Elad, M., Starck, J.L., Querre, P., Donoho, D.L.: Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). Applied and Computational Harmonic Analysis **19**(3) (2005) 340–358
35. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer graphics and applications **21**(5) (2001) 34–41
36. Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale image harmonization. In: ACM Transactions on Graphics (TOG). Volume 29., ACM (2010) 125

37. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on graphics (TOG) **22**(3) (2003) 313–318
38. Tao, M.W., Johnson, M.K., Paris, S.: Error-tolerant image compositing. In: European Conference on Computer Vision, Springer (2010) 31–44
39. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Yang, M.H.: Sky is not the limit: semantic-aware sky replacement. ACM Trans. Graph. **35**(4) (2016) 149–1
40. Johnson, M.K., Dale, K., Avidan, S., Pfister, H., Freeman, W.T., Matusik, W.: Cg2real: Improving the realism of computer generated images using a large collection of photographs. IEEE Transactions on Visualization and Computer Graphics **17**(9) (2011) 1273–1285
41. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
42. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
43. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016)
44. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., Metaxas, D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv preprint arXiv:1612.03242 (2016)
45. Durugkar, I., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673 (2016)
46. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
47. Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055 (2016)
48. Ziwei Liu, Ping Luo, X.W., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV). (2015)
49. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3) (2015) 211–252