Image Completion

Manikanta B, Geetha Charan Y
Department of Computer Science and Automation
Indian Institute of Science, Bangalore
manikantab@iisc.ac.in, geethacharan@iisc.ac.in

Abstract

In this work we explore two approaches presented for image completion. The first approach is an unsupervised visual feature learning algorithm driven by context-based pixel prediction[1], in which Context Encoder is proposed – which is a convolutional neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings. The second approach[2] uses a fully-convolutional neural network to complete images of arbitrary resolutions by filling in missing regions of any shape and then trains this image completion network with global and local context discriminators that are trained to distinguish real images from completed ones resulting in images that are both locally and globally consistent.

Figure 1: Example of Image Completion used for object removal.

1 Introduction

Image completion(or image inpainting) is a technique that allows filling in target regions with alternative contents. Although many approaches have been proposed for image completion it remains a challenging problem because it often requires high-level recognition of scenes as it is important to understand the anatomy of the scene and objects being completed. One of the main motivations of image completion is being able to remove unwanted objects in images. Such an object removal can be observed in Figure 1.

1.1 Image Completion using Context Encoders

In this approach, the authors showed that it is possible to learn and predict the structure of missing parts in a image using convolutional neural networks. Given an image with a missing region it is trained on a convolutional neural network to regress to the missing pixel values. This model was called context encoder, as it consists of an encoder capturing the context of an image into a compact

latent feature representation and then a decoder which uses that representation to produce the missing image content.

The context encoders are similar to autoencoders, But differs from autoencoders as they take an input image and try to reconstruct it after it passes through a low-dimensional bottleneck layer, with the aim of obtaining a compact feature representation of the scene. Unfortunately, this feature representation is likely to just compresses the image content without learning a semantically meaningful representation. Denoising autoencoders address this issue by corrupting the input image and requiring the network to undo the damage. However, this corruption process is typically very localized and low-level, and does not require much semantic information to undo. Whereas the context encoder needs to fill in large missing areas of the image, where it can't get hints from nearby pixels. This requires a much deeper semantic understanding of the scene, and the ability to synthesize high-level features over large spatial extents.

This requires the model to understand the content of an image, as well as produce a plausible hypothesis for the missing parts. The context encoders are trained jointly to minimize both a reconstruction loss and an adversarial loss. The reconstruction (L2) loss captures the overall structure of the missing region in relation to the context, while the the adversarial loss has the effect of picking a particular mode from the distribution.

1.2 Globally and Locally Consistent Image Completion

This approach builds upon the Context Encoder approach, which employs a Convolutional Neural Network (CNN) that is trained with an adversarial loss. The Context Encoder approach was motivated by feature learning, and did not fully describe how to handle arbitrary inpainting masks nor how to apply the approach to high resolution images. Whereas this proposed approach addresses these two points and further improves the visual quality of results.

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

2 Context Encoders for Image Completion

2.1 Architecture

The overall architecture is a simple encoder-decoder pipeline. The encoder takes an input image with missing regions and produces a latent feature representation of that image. The decoder takes this feature representation and produces the missing image content. We found it important to connect the encoder and the decoder through a channel wise fully-connected layer, which allows each unit in the decoder to reason about the entire image content.

Encoder The proposed architecture of Encoder takes an input image of size 128×128, we use the five convolutional layers, wherein each layer contains a convolution operation, LeakyReLU and Batch Normalization in the same order. There are no pooling layers involved in the encoder architecture. Intuitively there is no specific reason to use pooling for reconstruction-based networks. In classification, pooling provides spatial invariance, which may be detrimental for reconstruction based training. The final output of encoder is a bottleneck of 4000 units.

Decoder The bottleneck layer is followed by a series of five up-convolutional layers with learned filters, each with a rectified linear unit (ReLU) activation function. An up-convolutional is simply a convolution that results in a higher-resolution image. It can be understood as upsampling followed by convolution, or convolution with fractional stride.

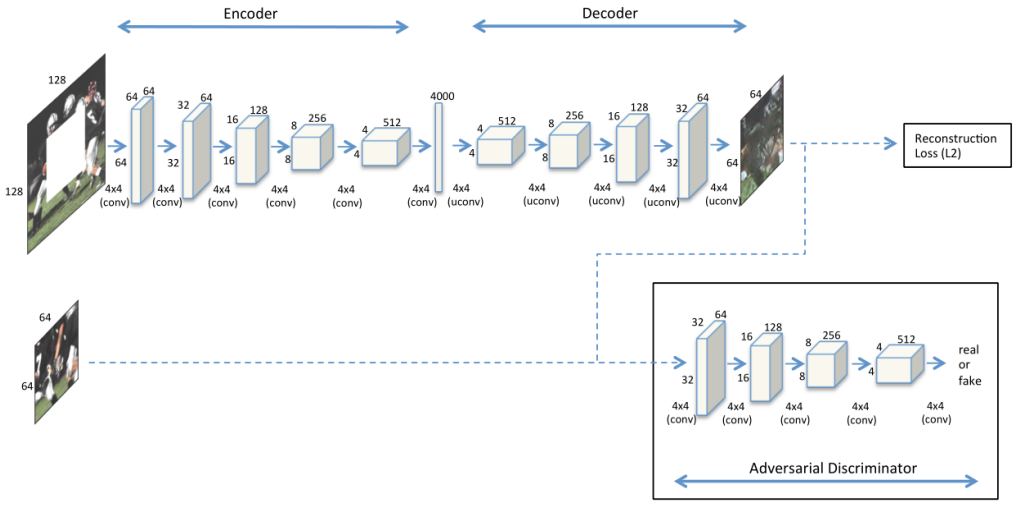

Figure 2: Overview of the Architecture. [Context Encoder for filling central missing region]

2.2 Loss Functions

We have already established that the reconstruction (L2) loss is responsible for capturing the overall structure of the missing region and coherence with regards to its context, but tends to average together the multiple modes in predictions. Whereas the adversarial loss tries to make prediction look real, and has the effect of picking a particular mode from the distribution.

For each ground truth image x, our context encoder F produces an output $F(x)$. Let $\hat{M}$ be a binary mask corresponding to the dropped image region with a value of 1 where a pixel was dropped and 0 for input pixels. We now describe different components of our loss function.

Reconstruction Loss We use a normalized masked $L2$ distance as our reconstruction loss function, $\mathcal{L}_{rec}$,

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

where $\odot$ is the element-wise product operation. The authors of the paper experimented with both L1 and L2 losses and found no significant difference between them. While this simple loss encourages the decoder to produce a rough outline of the predicted object, it often fails to capture any high frequency detail. This stems from the fact that the L2 (or L1) loss often prefer a blurry solution, over highly accurate textures. It is believed that this happens because it is much safer for the L2 loss to predict the mean of the distribution, because this minimizes the mean pixel-wise error, but results in a blurry averaged image. This is alleviated by adding an adversarial loss.

Adversarial Loss The adversarial loss is based on Generative Adversarial Networks (GAN). To learn a generative model G of a data distribution, GAN proposes to jointly learn an adversarial discriminative model D to provide loss gradients to the generative model. G and D are parametric functions (e.g., deep networks) where $G : \mathcal{Z} \rightarrow \mathcal{X}$ maps samples from noise distribution $\mathcal{Z}$ to data distribution $\mathcal{X}$. The learning procedure is a two-player game where an adversarial discriminator D takes in both the prediction of G and ground truth samples, and tries to distinguish them, while G tries to confuse D by producing samples that appear as real as possible. The objective for discriminator is logistic likelihood indicating whether the input is real sample or predicted one:

$$\min_G \max_D \mathbb{E}_{x \in \mathcal{X}}[\log(D(x))] + \mathbb{E}_{z \in \mathcal{Z}}[\log(1 - D(G(z)))]$$

This method has recently shown encouraging results in generative modeling of images. Thus the authors adapted this framework for context prediction by modeling generator by context encoder. To customize GANs for this task, one could condition on the given context information; i.e., the mask $\hat{M} \odot x$. However, conditional GANs don't train easily for context prediction task as the adversarial

discriminator D easily exploits the perceptual discontinuity in generated regions and the original context to easily classify predicted versus real samples. Thus an alternate formulation is used, by conditioning only the generator (not the discriminator) on context. The authors also found results improved when the generator was not conditioned on a noise vector.

Hence the adversarial loss for context encoders is

$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{x \in \mathcal{X}}[\log(D(x)) + \log(1 - D(F((1 - \hat{M}) \odot x)))]$$

where, in practice, both F and D are optimized jointly using alternating SGD. Note that this objective encourages the entire output of the context encoder to look realistic, not just the missing regions.

Joint Loss Finally the overall loss function is defined as

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}.$$

3 Globally and Locally Consistent Image Completion

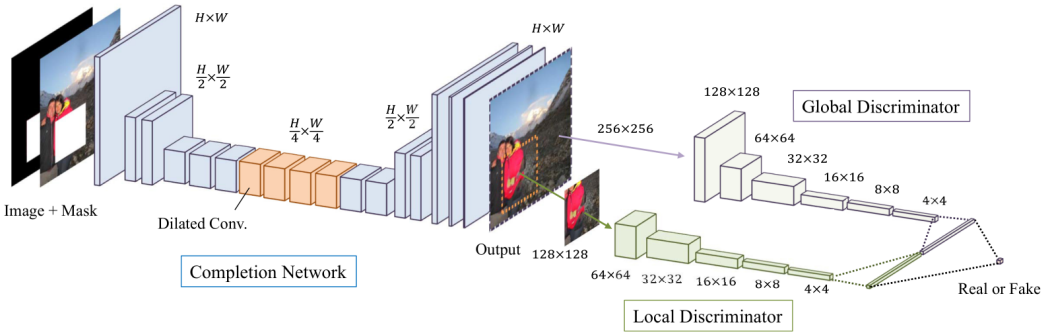

Figure 3: Overview of the Architecture [Approach 2]

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

3.1 Completion Network

The completion network is based on a fully convolutional network. The input of the completion network is an RGB image with a binary channel that indicates the image completion mask (1 for pixel to be completed), and the output is an RGB image. As we do not wish any change in areas other than the completion regions, the output pixels outside of the completion regions are restored to the input RGB values.

The general architecture follows an encoder-decoder structure, which allows reducing the memory usage and computational time by initially decreasing the resolution before further processing the image. Afterwards, the output is restored to the original resolution using deconvolution layers, which consist of convolutional layers with fractional strides. Unlike other architectures that use many pooling layers to decrease the resolution, this network model only decreases the resolution twice, using strided convolutions to $1/4$ of the original size, which is important to generate non-blurred texture in the missing regions.

Dilated convolutional layers are also used in the mid-layers. Dilated convolutions use kernels that are spread out, allowing to compute each output pixel with a much larger input area, while still using the same amount of parameters and computational power. This is important for the image completion

task, as the context is critical for realism. By using dilated convolutions at lower resolutions, the model can effectively see a larger area of the input image when computing each output pixel than with standard convolutional layers. The resulting network model computes each output pixel under the influence of a 307×307 pixel region of the input image. Without using dilated convolutions, it would only use a 99×99 pixel region, not allowing the completion of holes larger than 99×99 pixels.

3.2 Global and Local Discriminators

A global context discriminator network and a local context discriminator network are based on convolutional neural networks that compress the images into small feature vectors. Outputs of the networks are fused together by a concatenation layer that predicts a continuous value corresponding to the probability of the image being real.

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

The local context discriminator follows the same pattern, except that the input is a 128×128 pixel image patch centered around the completed region. In the case the image is not a completed image, a random patch of the image is selected, as there is no completed region to center it on. As the initial input resolution is half of the global discriminator, the first layer used in the global discriminator is not necessary. The output is a 1024 dimensional vector representing the local context around the completed region.

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

3.3 Training

Let $C(x, M_c)$ denote the completion network in a functional form, with x the input image and M_c the completion region mask that is the same size as the input image. The binary mask M_c takes the value 1 inside regions to be filled-in and 0 elsewhere. As a preprocessing, C overwrites the completion region of the training input image x by a constant color, which is the mean pixel value of the training dataset, before puting it into the network. Similarly, $D(x, M_d)$ denotes the combined context discriminators in a functional form. In order to train the network to complete the input image realistically, two loss functions are jointly used: a weighted Mean Squared Error (MSE) loss for training stability, and a Generative Adversarial Network (GAN) loss to improve the realism of the results. Using the mixture of the two loss functions allows the stable training of the high performance network model, and has been used for image completion [1], and concurrently with this work, for various image-to-image translation problems . Training is done with backpropagation. In order to stabilize the training, a weighted MSE loss considering the completion region mask is used. The MSE loss is defined by:

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

where $\odot$ is the pixelwise multiplication and $\|.\|$ is the Euclidean norm. The context discriminator networks also work as a kind of loss, sometimes called the GAN loss [3]. This is the crucial part of training in this approach, and involves turning the standard optimization of a neural network into a min-max optimization problem in which at each iteration the discriminator networks are jointly updated with the completion network. For this completion and context discriminator networks, the optimization becomes:

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

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

By combining the two loss functions, the optimization becomes:

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

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

$$\mathbb{E}[\nabla_{\theta c} L(x, M_c) + \alpha \log(1 - D(C(x, M_c), M_c)]$$

In practice, we take a more fine-grained control, such as initially keeping the norm of the MSE loss gradient roughly the same order of magnitude as the norm of the discriminator gradient. This helps stabilize the learning. We also update the discriminator networks D similarly, except we take update in the opposite direction so that the loss increases. Note that here D consists of the local and the global context discriminators. So the fow of the gradient in backpropagation initially splits into the two networks and then merge into the completion network. In optimization, we use the ADADELTA algorithm [4], which sets a learning rate for each weight in the network automatically.

4 Evaluation of Context Encoders

Now evaluating the Context Encoder for semantic image completion. Note that it can be extended to other image understanding tasks. The experiment is carried out using the MiniImageNet dataset without using any of the accompanying labels.

4.1 Semantic Inpainting

We train context encoders with the joint loss function defined in equation above for the task of inpainting the missing region. The encoder and discriminator architecture is similar to that of discriminator in [5], and decoder is similar to generator in that. However, the bottleneck is of 4000 units (in contrast to 100 in [5]). The authors used the default solver hyper-parameters suggested in [5] which are also used here. and the values of $\lambda_{rec} = 0.999$ and $\lambda_{adv} = 0.001$ are considered. However, a few things were crucial for training the model. We did not condition the adversarial loss nor did we add noise to the encoder. We use a higher learning rate for context encoder (10 times) to that of adversarial discriminator. To further emphasize the consistency of prediction with the context, we predict a slightly larger patch that overlaps with the context (by 7px). During training, we use higher weight ($10\times$) for the reconstruction loss in this overlapping region.

The qualitative results are shown in Figure 4 and 5. The model performs generally well in inpainting semantic regions of an image. However, if a region can be filled with low-level textures, texture synthesis methods, such as, can often perform better. It also shows that joint loss significantly improves the inpainting over both reconstruction and adversarial loss alone.

4.2 Results

The proposed model is implemented using the following github repo : code

Figure 4: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image generated from trained Context Encoder Model

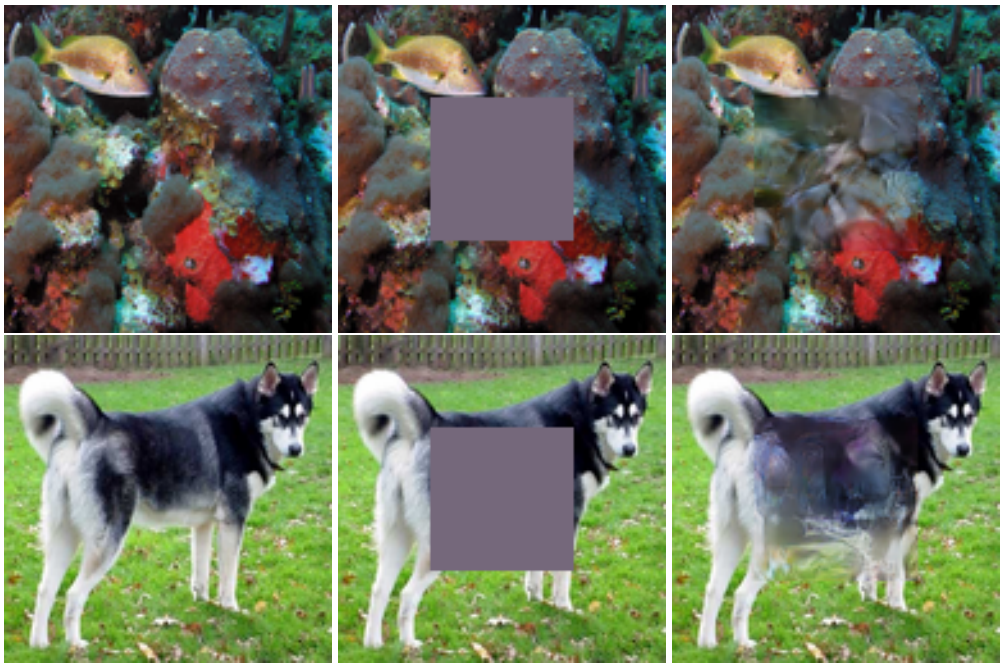

Figure 5: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image generated from trained Context Encoder Model

5 Evaluation of Globally and Locally Consistent Image Completion Model

This model is also trained using the MiniImagenet dataset. This dataset consists of 50000 training images and 10000 testing images, evenly distributed across 100 classes. We set the weighting hyper-parameter to $\alpha = 0.0004$, and train using a batch size of 96 images. The completion network is trained for $T_C = 90,000$ iterations; then the discriminator is trained for $T_D = 10,000$ iterations; and finally both are jointly trained to reach the total of $T_{train} = 150,000$ iterations.

5.1 Results

The proposed model is implemented using the following github repo : code

Figure 6: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image generated from trained Globally and Locally Consistent Image Completion Model

Figure 7: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image generated from trained Globally and Locally Consistent Image Completion Model

6 Comparison

Comparing the Context Encoder (CE) [1] approach and Global and Local discriminator approach (GL) on the 128×128 pixel test images, taken from MiniImageNet considered in CE approach, with the fixed 64×64-pixel inpainting masks in the center of the image. For a fair comparison, the models are trained similarly. And also no post-processing is performed for the results of the GL model. Results are shown below in Figure 6. For the center region completion task, the results of CE are signifcantly better than in the general arbitrary region completion case. It should be noted that, while the CE approach is specialized to inpaint images of this size and fixed holes, the global and local discriminator model is capable of arbitrary region completion at any resolution. It also signifcantly improves the visual quality by employing both a global and local discriminator.

Figure 8: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image (CE) (d) Reconstructed Image (GL)

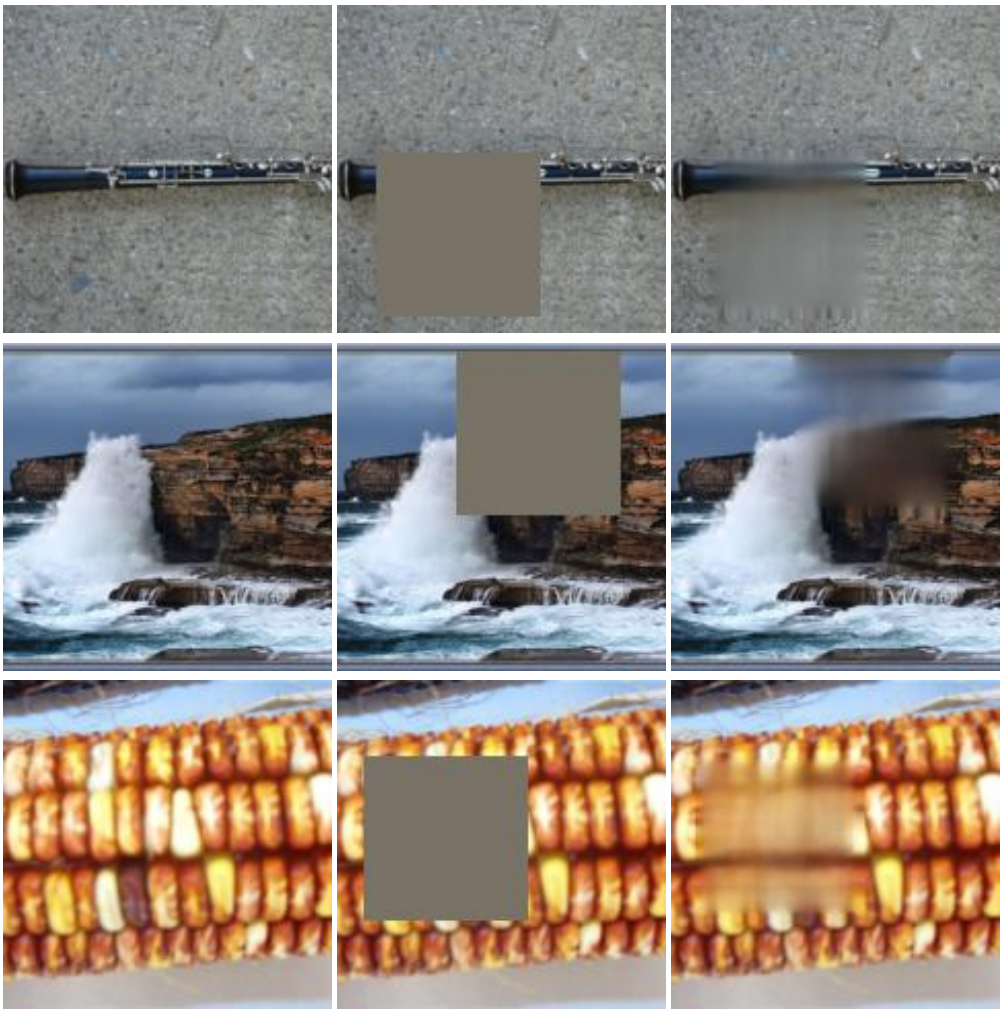

Figure 9: (a) Ground Truth Image (b) Masked Image (c) Reconstructed Image(GL)

7 Conclusion

In this work we qualitatively compared the Context Encoder approach and Globally and Locally Consistent Image Completion approach. The Context Encoder was successfully enable to inpaint the central missing region of an image. The global and local discriminator model is capable of arbitrary region completion at any resolution. It also significantly improves the visual quality by employing both a global and local discriminator.

References

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

[2] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. ACM Trans. Graph. 36, 4, Article 107 (July 2017), 14 pages.

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

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

[5] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR, 2016.