Image Inpainting using Deep Learning Models

Akanksha Gupta
University of Massachusetts
akankshagupt@cs.umass.edu

Abstract

Image inpainting task is a challenging task which involves filling missing pieces in a given image using visually similar images. Recent papers have shown promising results by using deep generative models and convolution neural networks (CNN). In this paper, I'm comparing the results of three recent papers: Semantic Image Inpainting with Deep Generative Models (SIIDGM) [3], Globally and Locally Consistent Image Completion (GLCIC) [1], Generative Image Inpainting with Contextual Attention (GIICA) [4]. I'm also presenting the results from my method by improving upon them. My network trains CNN and contextual attention layers in parallel, fuses the two embeddings and send to global and local discriminators. The entire network is learned end-to-end. The results obtained are visually appealing and decently closer to the ground truth.

1. Introduction

The images of our visual world are highly structured. If some piece is missing or damaged, human eye can easily imagine that part by extrapolating or taking reference from some other images. Recent success of deep generative models and convolutional neural networks have made it possible to learn these structures. Given an image with missing regions, the structures learned can be used to fill the pieces and reconstruct the image. This is an image inpainting task. I recreated the results from three recent papers SIIDGM, GLCIC and GIICA for this task. The images generated were either distorted, blurry or not matching the ground truth. None of the methods were able to generate visually appealing and realistic images. I improved upon the methods and my methods are able to remove the distortion and generating realistic, visually appealing images.

In this paper, I'm presenting two modified approaches. The first one is inspired from [1] and [4] papers. The results from the two papers were not satisfactory. The first one was generating distorted images without texture information and the second was capturing texture but generating deformed images. My architecture has three components: encoder-decoder part for completing the image; contextual attention part for capturing the texture information; global and local discriminator for global coherency and capturing fine grained details of the local region. The first two parts are parallel to each other. The outputs from them are fused in between and passed to the decoder which then outputs the completed image. The completed image is passed to the global discriminator and the missing region area is passed to the local discriminator. The outputs from the two discriminators are concatenated and passed to a fully connected layer which outputs the probability of the image being completed or real. The network is learned end-to-end. The results generated by my network are impressive. In the second approach, I've added a symmetry loss in SIIDGM paper to capture the inherent symmetry of faces. The details of the two approaches are discussed in section 3.

2. Related Work

The three networks I've done a comparative study on and have improved upon are discussed below. Before these papers, the best results were presented by Context Encoders [2] which is an unsupervised visual feature learning algorithm driven by context-based pixel prediction.

2.1. Semantic Image Inpainting with Deep Generative Models

This method does not require the masks for training and can be applied for arbitrarily structured missing regions during inference as compared to the CE. They consider semantic inpainting as a constrianed image generation problem and take advantage of recent development in generative modeling. The network uses an already trained generative model, in this case Deep Convolutional Generative Adversarial Network (DCGAN). The network searches for an encoding (random noise z) of the corrupted image that is closest to the image in the latent space. The encoding is then used to reconstruct the image using the generator. The closest is defined by a weighted context loss to condition on the corrupted image, and a prior loss to penalize unrealistic images.

2.2. Globally and Locally Consistent Image Completion

As compared to CE this approach can handle arbitrary inpainting masks and high resolution images. The 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 used for global and local consistency. During training, the discriminators are updated first to 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.

2.3. Generative Image Inpainting with Contextual Attention

It's a two stage process. First stage is a simple dilated convolutional network trained to output coarse results. The second stage contains contextual layer and convolutional layer in parallel to get refined results. Contextual attention layer is implemented with convolution for matching generated patches with known contextual patches, channel-wise softmax to weigh relevant patches and deconvolution to reconstruct the generated patches with contextual patches. It also has spatial propagation layer to encourage spatial coherency of attention. The parallel convolutional layer is used to hallucinate novel contents. The two results are aggregated and fed into single decoder to obtain the final output. After this global and local discriminator is used similar to GCIC paper.

3. Description of my method and architecture

3.1. First Approach

My approach is inspired from the two papers [1] and [4]. After recreating the results from both the methods, I noticed the flaws and imperfection in the generations. The first one generates distorted images without texture information as shown in appendix 4 and the second one captures the texture but generates deformed images as shown in appendix 5. The network architecture of the second is complicated and takes time to train and infer. The architecture of the first is simple but doesn't involve contextual information. My approach is a combination of the two and is a simplified version. It involves deep convolutional neural network parallel with the contextual attention layer which learns where to borrow or copy feature information from known background patches. This is for completing the image and called completion network. The embeddings generated by the two networks are concatenated in the middle followed by rest of the deep convolutional network. The output is passed to the local and global discriminators and the entire network is trained end-to-end. The discriminators are trained to see if the image generated is realistic. The completion network is trained to fool them. The overview of the approach can be seen in Figure 1. I'm not showing global and local discriminator architecture here as it is similar to GLCIC paper except that I'm using one less layer in the starting. There are four parts in the network which are explained below.

3.1.1 Deep Convolutional Neural Network

This is a fully convolutional network modeled in encoder decoder fashion by first downsampling the size of the image and then upsampling it to the original size. The input is a 128X128 RGB image and a binary mask image with value 1 for the missing regions and the output is a 128X128 RGB image filling the missing region using the network. The non-missing region is restored from the input image. The network has 6 convolutional layers with 5X5 filter on the first and 3X3 on the rest. followed by 4 dilated layers with 3X3 filters. The dilated layers are used to provide a larger area of the low resolution input image while keeping the same number of parameters. This is followed by 2 convolutional layers with filter 3X3 then deconvolution with filter 4X4 and stride 1/2, convolution with filter 3X3, deconvolution with filter 4X4, and finally two more convolutions of 3X3 filter. Instead of using maxpoolinng for downsampling and upsamling the network uses strides in order to get non blurred images.

3.1.2 Contextual Attention Layer

The contextual attention layer I used is similar to as discussed in the related work section. The deep convolutional layers are not capable of borrowing texture information from distance pixels. This layer converts the background patches as convolution filters and try to find the patch which is similar to the foreground region. For parallel training with deep convolutional layers, same architecture is followed. The contextual information is obtained after 6 convolutional layers which is followed by two more convolutional layers. The output from this and deep convolutional layer is concatenated along the third axis before sending it to the seventh convolutional layer.

3.1.3 Discriminators

The discriminator networks are trained to determine if the image is real or completed by the completion network. They follow the same principles as any other discriminator in Generative Adversarial Networks (GAN). They compress the image into feature embeddings which are concatenated and passed to a fully connected layer. The final output is a real number denoting the probability if the image is real or completed.

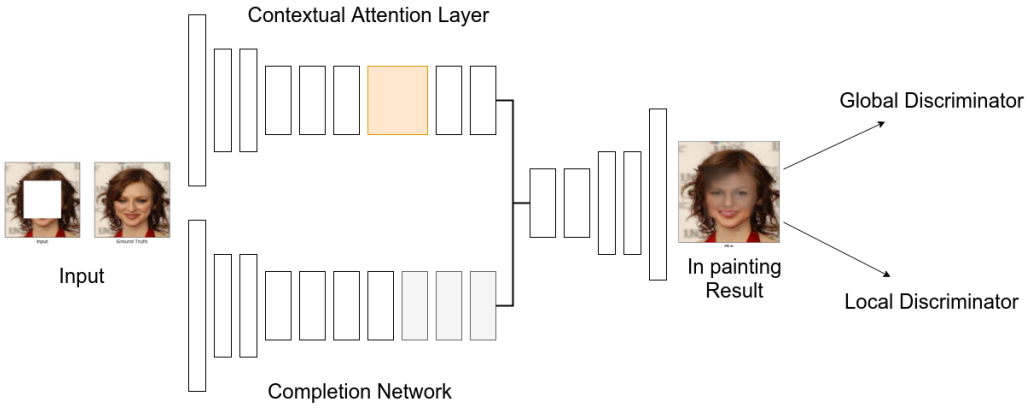

Figure 1. My architecture

Global context discriminator network The entire completed image 128X128 is passed to this network in order to provide the global information. In a scenic image, if a small portion of mountain is missing, it would be beneficial to provide the information of sky. It consists of five convolutional layers and a single fully-connected layer that outputs a single 1024 dimensional vector. I'm using a smaller size image as compared to the original paper [1] therefore just need five layers instead of six. All the convolutional layers employ a stride of 2X2 pixels to decrease the image resolution while increasing the number of output filters. All convolutions use 5X5 kernels.

Local context discriminator network The architecture of this layer is similar to global discriminator layer. The input is a 56X56 image patch which was completed by the completion layer. During training time only one such image patch will be present while during inference the network can handle any number of missing pieces. As the size is half of what is provided to the global discriminator, the first layer wouldn't be present here. The output here is again a 1024 dimensional vector providing local information around the missing patch region which was completed.

3.2. Second Approach

This approach is inspired from [3] paper. The paper uses generative model to complete the image and then iteratively try to find the match closest to the original image. The input to the netowrk is random noise z and the network try to find the z which will generate the best image. Since we know that the face images are symmetric in nature, I put a constraint of generating symmetric images by adding one more loss to the already existing contextual and discriminator losses.

3.3. Losses

Loss for first approach One is contextual loss which is weighted difference between the completed image and the original image. The second and the third ones are discriminator losses similar to GAN models. Let $C(x, M_c)$ be the completion network in a functional form, with x the input image and M_c the completion region mask. Similarly, $D(x, M_d)$ be the combined context discriminators in a functional form. Therefore, the overall loss will look:

$$L(x, M_c) = ||M_c \odot (C(x, M_c) - x)||^2 \tag{1}$$

$$L_p = \min_C \max_D E[log(D(x, M_d)) + log(D(C(x, M_c), M_c))] \tag{2}$$

The two losses will be minimized during backpropogation

Loss for second approach Apart from the two losses, contextual and discriminator, one more loss will be added for symmetry constraint. The loss will look like:

$$L_c(z|y, M) = ||W \odot (G(z) - y)||^2 + ||W \odot (G(z) - mirror(y))||^2 \tag{3}$$

where W_i denotes the importance weight at pixel location i. G(z) is the generated image and y is the missing region image.

$$L_p(z) = \lambda log(1 - D(G(z))). \tag{4}$$

The two losses will be minimized during back propogation.

4. Experimental Evaluation

I recreated the results of three papers [3] as shown in appendix 6, [1] in appendix 4 and [4] in appendix 5. The first paper can only produce 64X64 images as it's using DCGAN generative model. The results are a little distorted, doesn't capture texture information but closer to the ground truth. I used this code `https://github.com/`

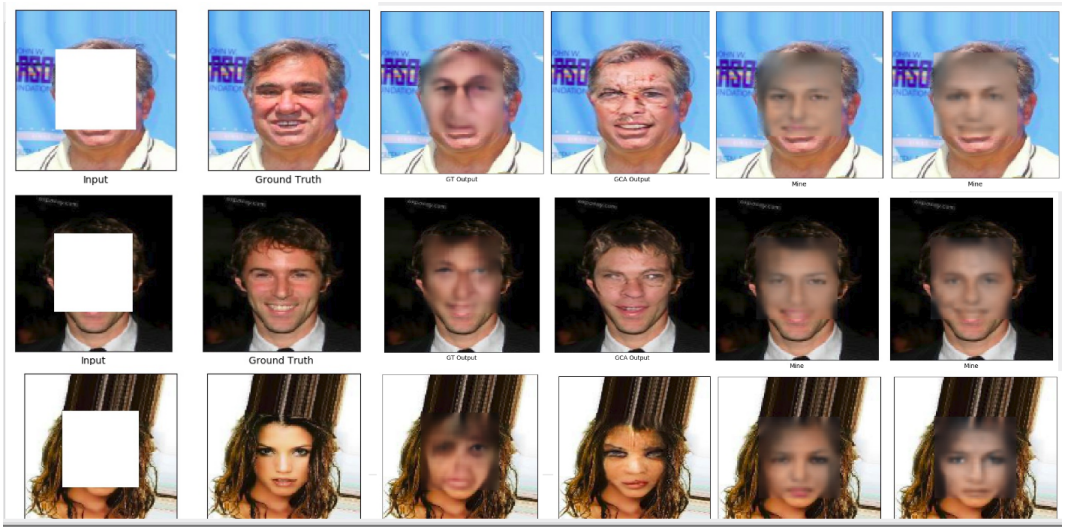

Figure 2. From left to right: First column contains the input image with missing region, second contains the ground truth, third contains results from Globally and Locally Consistent Image Completion paper, fourth contains results from Generative Image Inpainting with Contextual Attention paper, fifth contains results from my architecture during initial stage of epochs, sixth contains results after 60 epochs

`moodoki/semantic_image_inpainting` to generate the results. The second paper can work on any image size and any scale of missing regions. The images are distorted, not close to the ground truth, not visually appealing and don't capture texture information. I used this code `https://github.com/tadax/glcic` to generate the results. This is not the official code provided by the authors but I checked the architecture and the code and it seemed to be similar to the architecture described in the paper. The third paper is doing a good job in capturing texture information but the results are distorted and not visually appealing. I used this code `https://github.com/JiahuiYu/generative_inpainting` to generate the results. It's difficult to evaluate the methods for tasks like image synthesis, image editing, image-to-image translations, image inpainting, etc quantitatively. All the methods discussed above are using different size images and working on different scale of missing regions. There can be a huge information loss while resizing the images. The loss can be large because of small distortions even when the image generated is visually appealing and closer to the ground truth. Therefore, for the purpose of my evaluation I'm relying only on qualitative results by looking at the same set of images for all the methods.

4.1. First approach

I used 12000 images of Celeb A dataset which contains faces of celebrities. I trained the network on one GPU Tesla M40 for 60 epochs. The training took more than five-six days as the network is learned from scratch without any transfer learning. As can be seen from Figure 2, my results are much better than GLCIC and GIICA papers. The images are not distorted and visually appealing. As can be seen from fifth (inital stage of learning) and sixth (after 60 epochs) columns, the results are moving towards the ground truth. Currently, the images generated are not capturing the texture information but if all of the CelebA data set is used containing 202,599 images as used in the other papers, the results will be much better. So, from my method I'm removing the distortion, the results are better looking and visually appealing and the texture information will be captured given enough time and resources. Comparison results for side pose can be found in appendix 7 and more results can be found in appendix 8

4.2. Second approach

I added the symmetry loss and then iteratively updated the z in order to minimize the combined loss. Even though this method just works with 64X64 images, the training time is much less than all the other methods if we don't consider the training time of GANs. Now that the GAN models are available for higher resolution images, this method can generate better results than other methods discussed in this paper. The results can be seen in Figure 3. The results have improved by adding the symmetry loss but at the same time results didn't improve for side poses.

5. Conclusion

My network is able to generate promising results. Currently, it's able to generate non distorted and realistic images. Given more resources, it'll capture texture information also.

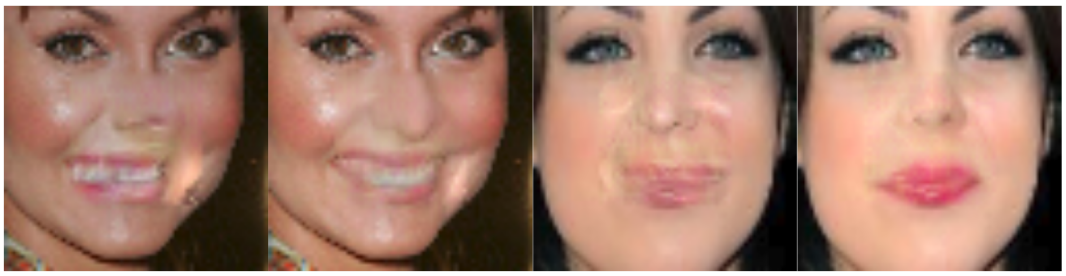

Figure 3. The left image is generated by Semantic Image Inpainting with Deep Generative Models paper and right is generated by mine

References

[1] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, July 2017.

[2] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. 2016.

[3] R. A. Yeh, C. Chen, T. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *CoRR*, abs/1607.07539, 2016.

[4] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *CoRR*, abs/1801.07892, 2018.

A. Results

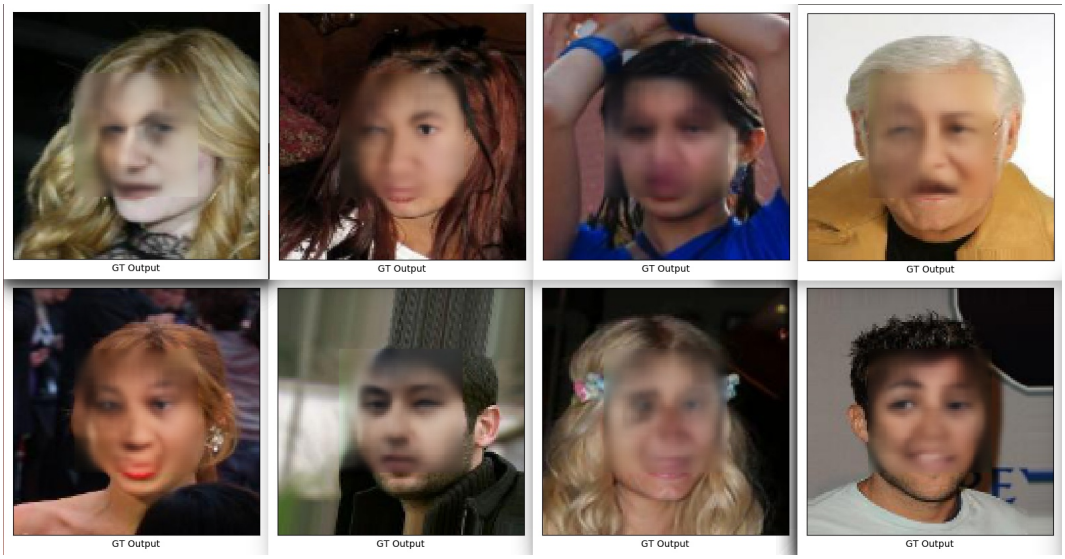

Figure 4. Globally and Locally Consistent Image Completion Results

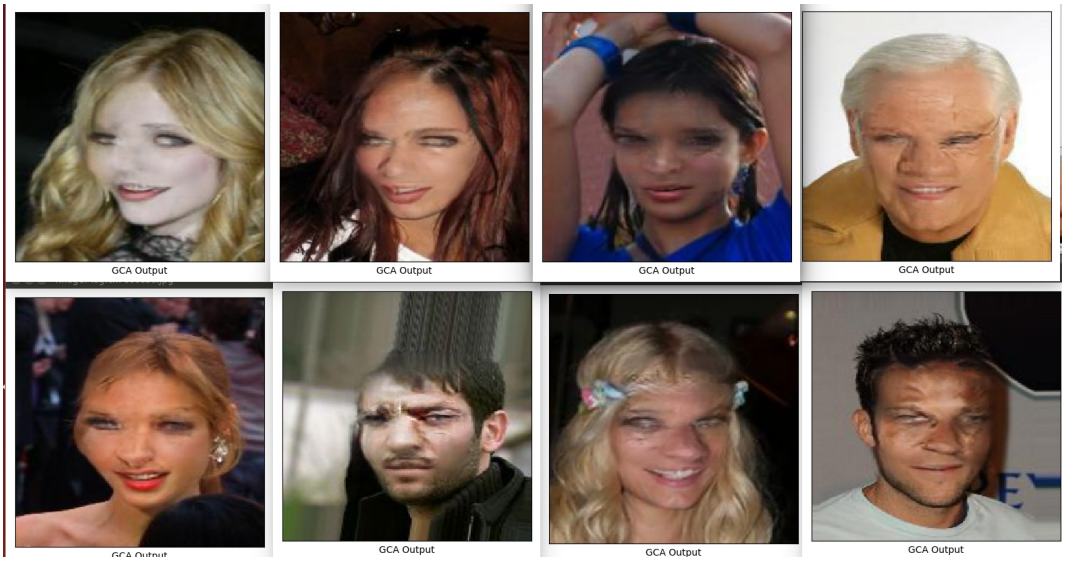

Figure 5. Generative Image Inpainting with Contextual Attention Results

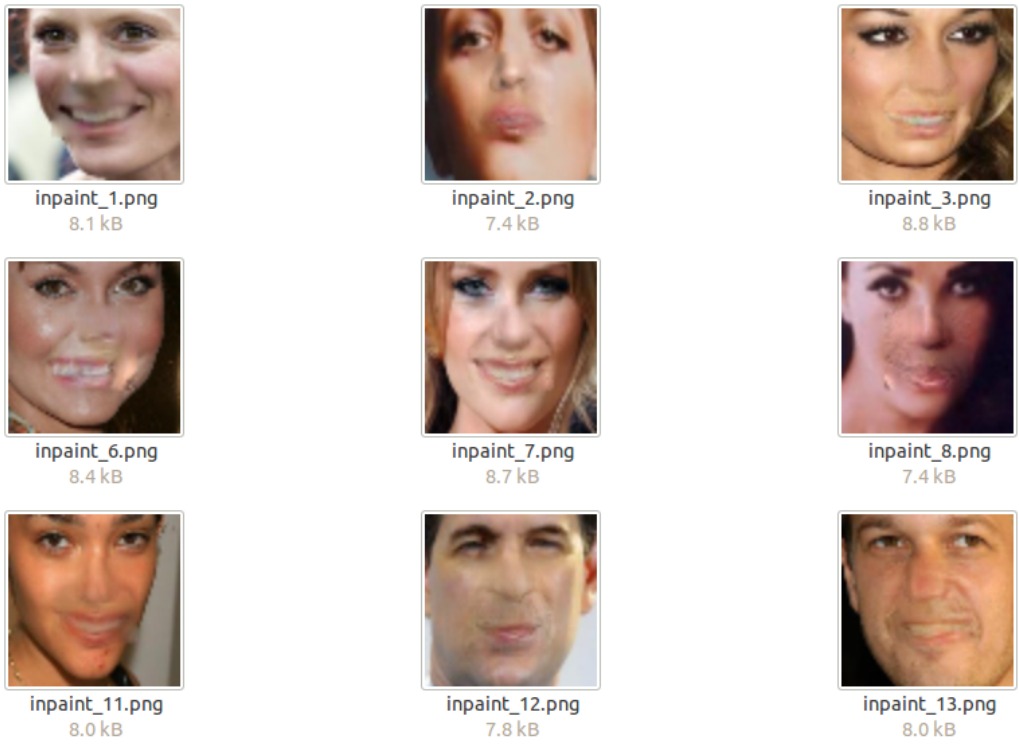

Figure 6. Semantic Image Inpainting with Deep Generative Models

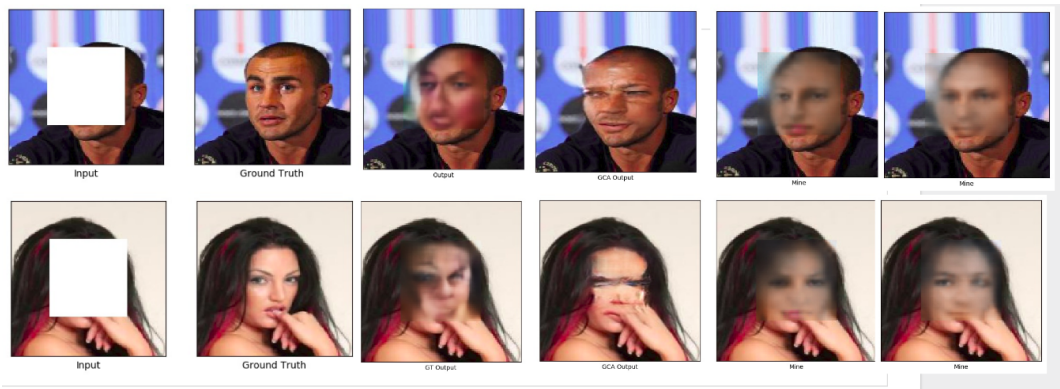

Figure 7. Comparision for side poses

Figure 8. Results from my method