Conditional Deep Convolutional GANs for Image Completion

Zhi Ji[1], Ziyu Chen[2], and Han Zhang[3]
Department of Electrical Engineering[1,3] and Department of Statistics[2]
Columbia University
zj2242@columbia.edu[1] zc2393@columbia.edu[2] hz2482@columbia.edu[3]

ABSTRACT

Image completion has shown significant progress due to the improvement in generative adversarial networks(GANs). The aim image completion problem is to fill missing or unwanted parts in images in a visually satisfactory manner. In this project, we develop a conditional GAN architecture with global and local discriminators using pyramid structure. We test our model on human face dataset and demonstrate its high performance to naturally complete the images of human faces.

Index Terms— image completion, generative adversarial networks.

1. INTRODUCTION

Image completion, also known as image inpainting, is a widely researched topic in computer vision. It refers to the tasks of filling up the target regions of an image with alternative contests. This image editing technique can be also used to remove unwanted objects and reconstruct occluded regions. In recent years, due to the advances in deep learning methods, image completion has made rapid development. It shows that deep neural networks can synthesize realistic looking images in many applications.

In this project, we focus on face completion task, and proposed a pyramid structure conditional deep convolutional GANs(cDCGANs) based on residual network with two discriminators: a global context discriminator and a local context discriminator. We applied and evaluate our approach on Large-scale CelebFaces Attributes(CelebA) Datase. By using our model, we increase the quality of the completed face images and achieve a high performance in model convergence.

2. RELATED WORKS

Traditional approaches. There are a number of different approaches to solve the problem of image completion. Traditional methods are mostly non-learning approaches. One of them exploits a diffusion equation which largely involves iteratively propagating local image appearance information to the target region based on certain mechanisms [6][14]. Another traditional approach is patch-based methods [7][15]. However, these approaches are usually computationally costly and not fast enough for real-time applications.

Convolutional Neural Networks. In the recent decade, many new approaches have emerged. For example, CNNs have also been used to handle image completion tasks [5].

A key implementation of image completion using CNN was introduced by Pathak et al. [8] where they trained a CNN adversarially to achieve the effect of pixel-wise reconstructing the missing image regions given or conditioned on its surrounding pixels.

In 2016, Yu and Koltun introduced dilated convolution to enhance the performance of semantic segmentation models without loss of resolution or coverage [4]. In 2017, Lin et al. introduced a new feature pyramid networks for object detection and building high-level semantic feature maps at all scales [3]. We will borrow the ideas of these two structures into our own generator network to achieve a semantically aware image completion effect.

Generative Adversarial Networks. In more recent years GANs have attracted increasingly amount of attention from researchers. GANs are a framework to train generative models that are able to synthesize high quality images [9]. They can also be applied to image completion tasks as long as the model is constrained by the provided corrupted image. A novel architect is to add a local discriminator that only takes the patch and its immediately surroundings as input [1]. The local discriminator along with the global discriminator results in images that are both locally and globally consistent. We follow the idea of using two discriminators in our model.

3. METHOD AND MODEL

3.1. Conditional Deep Convolutional GAN

Generative adversarial networks(GANs) consists of two neural networks, a generator G and a discriminator D [9]. These two networks are trained in competition with each other, while G tries to generate outputs $G(z)$ that have similar distribution to the input data sample x, and D attempts to discriminate between real samples x and fake samples $G(z)$.

Conditional deep convolutional GAN (cDCGAN) model is mostly based on DCGAN but introduces additional inputs, some extra information y[5]. y could be class labels or other data. By feeding the inputs y into both the generator and the discriminator, the model achieves conditioning [10].

3.2. Generator

The input of the generator is an RBG image plus a mask indicating which pixels are missing. The mask is a binary channel with 1 for pixels to be completed and 0 for others. The output is also an RGB image, which has the same resolution as the original one.

The generator network structure is based on residual networks. Dilated convolutional layers are used in the middle layers. A 2D dilated convolution layer can be defined as:

$$y(u,v) = \sigma\left[b + \sum_{i=1}^{U}\sum_{j=1}^{V} x(u+ri, v+rj)w(i,j)\right] \quad (1)$$

where $x(u,v)$ and $y(u,v)$ are the pixel of the input and output of the layer, σ is a non-linear transfer function, w is the matrices of the filter, and b is the bias. The parameter r is the dilation rate. For the layers with dilated rate larger than 1, adjacent pixel in output are computed from completely separate pixels in a much larger area of the input. The dilated convolutions allow the model to learn a larger area of the image while saving on the number of parameters [4].

We also employ a feature pyramid structure in the generator. This structure can generate multi-scale feature maps and extract higher quality feature information. It involves a bottom-up and a top-down pathway. The bottom-up pathway extract features in several sales. The resolution of the feature decreases as we compute to a higher level, while the semantic value increases. The top-down pathway reconstructs higher resolution features by upsampling the semantically stronger feature layer provided by the higher pyramid levels. By implementing the pyramid structure in the generator, the network enhances the efficiency of detecting accurate feature information [3].

3.3. Discriminators

The discriminator network in our model consists of aglobal discriminator and a local discriminator. Both of them are used to distinguish between the original images and the completed images. The output of this network is the fusion of the two discriminators which represents the probability of the image to be real. The global discriminator takes the whole image as its input in order to learn the global consistency. While, the local discriminator only inputs a small area around the completed region so it can measure the completion quality of the details [1][11].

Table 1. The architecture of our generator G. Each convolution is followed by a Leaky Rectied Linear Unit (ReLu) activation.

Block	Type	Kernel	Dilation (η)	Stride	Outputs
Conv1	conv.	3 × 3	1	1 × 1	64
Residual block 1	conv.	3 × 3	1	2 × 2	64
	conv.	3 × 3	1	2 × 2	64
	conv.	1 × 1	1	2 × 2	64
Residual block 2	conv.	3 × 3	1	2 × 2	128
	conv.	3 × 3	1	2 × 2	128
	conv.	1 × 1	1	2 × 2	128
Residual block 3	conv.	3 × 3	1	2 × 2	256
	conv.	3 × 3	1	2 × 2	256
	conv.	1 × 1	1	2 × 2	256
Dilated conv.	dilated.	3 × 3	2	1 × 1	1024
	dilated.	3 × 3	4	1 × 1	1024
	dilated.	3 × 3	8	1 × 1	1024
	dilated.	3 × 3	16	1 × 1	1024
Residual block 4	conv.	3 × 3	1	2 × 2	256
	conv.	3 × 3	1	2 × 2	256
	conv.	1 × 1	1	2 × 2	256
Residual block 5	conv.	3 × 3	1	2 × 2	128
	conv.	3 × 3	1	2 × 2	128
	conv.	1 × 1	1	2 × 2	128
Residual block 6	conv.	3 × 3	1	2 × 2	64
	conv.	3 × 3	1	2 × 2	64
	conv.	1 × 1	1	2 × 2	64
Deconv	deconv.	4 × 4	1	1/2 × 1/2	32
	conv	3 × 3	1	1 × 1	32
N/A	output	3 × 3	1	1 × 1	3

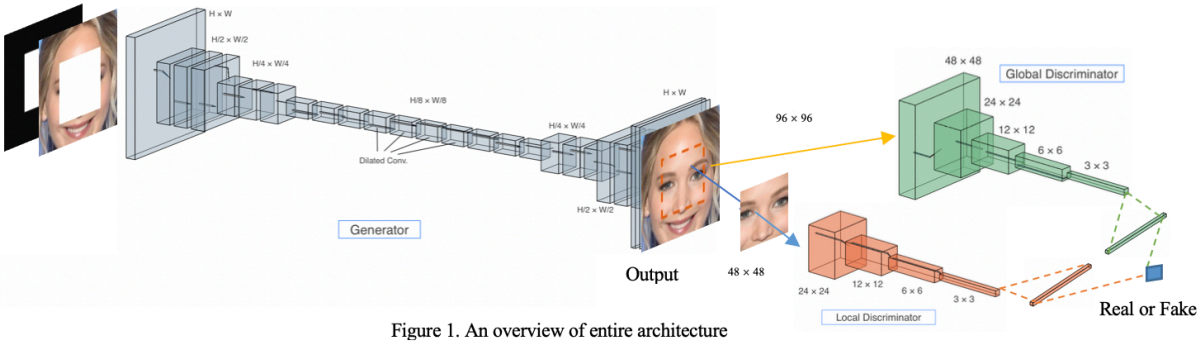


Figure 1. An overview of entire architecture

3.4. Classifier Network

Our model uses a general multiclass classifier. By adding this auxiliary classifier, the generator is forced to complete the image belonging to a specific class, thus enhance the completion quality [1].

3.5. Loss Function

Let x denote the input image and M_g denote the completion region mask. Then $G(x, M_g)$ denotes the generator network. And similarly $D(x, M_d)$ denotes the combined discriminator networks.

For the generator and discriminator networks, the GAN loss is used to determine the optimal point of the minimax game of the GAN:

$$\mathcal{L}_g = \log D\big(G\big(x, M_g\big), M_g\big) \tag{2}$$

$$\mathcal{L}_d = \mathbb{E}\Big[\log D(x, M_d) + \log\Big(1 - D\big(G\big(x, M_g\big), M_g\big)\Big)\Big] \tag{3}$$

at which point, the generator will generate features similar to the real ones, and the discriminators will distinguish the real and fake features. These loss functions measure how well each network is doing compared to the other.

For the classifier, we have a loss function for real features as:

$$\mathcal{L}_c^r = \mathbb{E}[\log P(Y = y|x)] \tag{4}$$

and one for the fake features:

$$\mathcal{L}_c^f = \mathbb{E}\Big[\log P\Big(Y = y\Big|G\big(x, M_g\big)\Big)\Big] \tag{5}$$

3.6. Training

Algorithm 1 summarize the training procedure. After preprocessing, we train the generator and generate fake images. Then, the discriminator and the classifier are trained to fix the generator. Finally, update all the network parameters.

4. EXPERIMENTS

4.1. Dataset

Different from other kind of completion, the face image completion problem is a more challenging task, as it often requires to generate novel objects or missing key components. The human face dataset we use in this project is the CelebFaces Attributes (CelebA) Dataset [12]. This large-scale face attributes dataset consists of 202,599 number of face images of 10,177 unique identities, with 5 landmark locations and 40 binary attributes annotations per image. The rich attributes annotations of each image including wearing hat, eyeglasses, wavy hair, mustache, smiling, etc.

4.2. Environment

In the experiment, the training was carried out on the Google Cloud Platform. We used one NVIDIA Tesla P100 GPU which contains 16GB RAM. Besides we used the TensorFlow-gpu, CUDA, and cuDNN SDK. The model takes about 5 hours for a 5000-iteration training.

Algorithm 1 Training algorithm

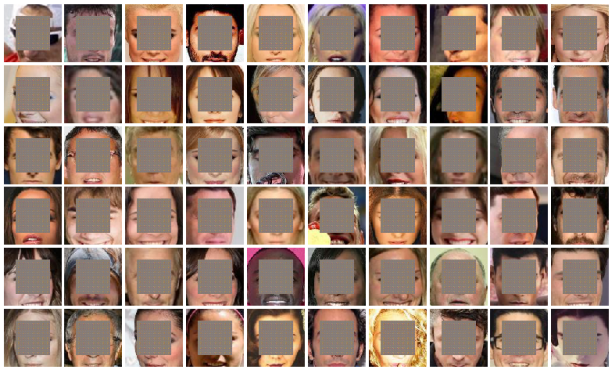
Input: images x, labels y
while iterations $t < T_{train}$ **do**
  Sample minibatch of images x from training data.
  Generate masks M_g for each image x in the minibatch.
  if $t < T_g$ **then**
    Update the generator G with loss $\mathcal{L}_g$.
  else
    Generate mask M_d for each image x in the minibatch.
    Update the discriminator D with loss $\mathcal{L}_d$.
    Update the classifier C with loss $\mathcal{L}_c^r$ and $\mathcal{L}_c^f$.
    if $t > T_g + T_d$ **then**
      Update the generator G with the joint loss gradient, and update discriminator D.
    end if
  end if
end while

4.3. Result

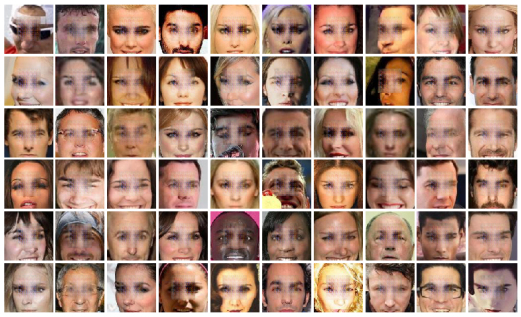
4.3.1. Face Completion

Given the face images with masks, the model will generate completed faces. The completion results are shown in figure 2. We can see that our model learns to generate missing components successfully and realistically. The generated areas are of high quality and both locally and globally coherent.

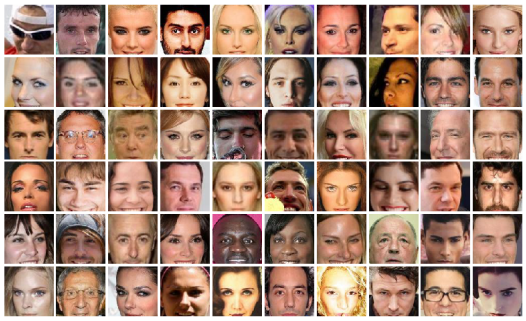
We also study the influence of various configurations by training the image completion model using different structures. Figure 3 presents the results for 3 settings and the full approach. As we can see, the simple GAN with only one discriminator has a poor-quality output and the patches are not consistent with the entire faces. The pyramid structure can synthesize some components, but still lack consistency with surrounding regions. By using dilated convolutional layers in the generator, the output has a much higher quality while the synthesized area is completion by some blur due to not using the local discriminator. Only the result trained by the full approach with two discriminators is consistent both globally and locally and have fine details.



(a) Input



(b) Output



(c) Ground truth

Fig.2. Example result of our face completion model.

(a) (b) (c) (d)

Fig.3. Comparison of training with different configurations. Here show the results of model trained with different structure configurations: (a) using simple GAN with only global discriminator, (b) using pyramid structure generator and only global discriminator, (c) using pyramid structure generator with dilated convolutional layers and only one discriminator, (d) using pyramid structure generator with dilated convolutional layers and both global and local discriminators.

4.3.2. Training Loss

Figure 4 illustrates the training loss curves for 3 model structures. We can observe that all three models result in similar loss for generator, discriminator, and classifier. But the for the real loss and fake loss of the classifier, the model using pyramid structure generator with dilated convolutional layers and the full approach model shows the lower loss. All the three models converge fast.

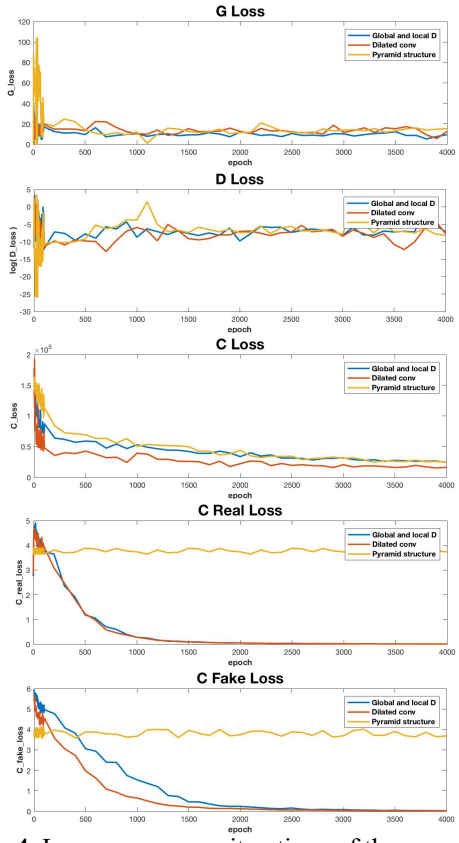


Fig.4. Loss curves over iterations of three model configurations.

5. CONCLUSION/DISCUSSION

We proposed and realized a pyramid structure conditional deep convolutional GANs with global and local discriminators for human face completion based on residual network. We demonstrate that our model can synthesize high quality realistic face components. We also compare the influence of different partial structure with the full model and show superiority of our model.

In future work, we are going to adapt our image completion model to other scenarios, such as building and animal images. Moreover, we aim to apply and further extend the model to tackle face recognition problems.

ACKNOWLEDGMENT

WE WOULD LIKE TO THANK THE INSTRUCTORS AND ALL THE TAS IN COURSE EECS6894 FOR THE PATIENT GUIDANCE AND ADVICE.

APPENDIX

All codes will be publicly available at https://github.com/zailchen/Conditional-Deep-Convolutional-GANs-for-Image-Completion

REFERENCES

[1] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa, “Globally and Locally Consistent Image Completion,” ACM Transaction on Graphics (Proc. of SIGGRAPH), 2017.

[2] Z. Ding, Y. Guo, L. Zhang and Y. Fu, “One-Shot Face Recognition via Generative Learning,” 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, 2018, pp. 1-7.

[3] Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan and Serge J. Belongie, “Feature Pyramid Networks for Object Detection,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017), pp. 936-944.

[4] Yu, Fisher and Vladlen Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions,” CoRR abs/1511.07122 , 2015.

[5] Köhler Rolf, Schuler Christian, Schölkopf Bernhard and Harmeling Stefan, “Mask-Specific Inpainting with Deep Neural Networks,” German Conference on Pattern Recognition, 2014.

[6] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," in IEEE Transactions on Image Processing, vol. 10, no. 8, pp. 1200-1211, Aug. 2001.

[7] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1033-1038 vol.2.

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

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

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

[11] Y. Li, S. Liu, J. Yang, and M.-H. Yang, “Generative face completion,” arXiv preprint arXiv:1704.05838, 2017.

[12] Z. Liu, P. Luo, X. Wang and X. Tang, “Deep Learning Face Attributes in the Wild,” 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 3730-3738.

[13] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, “Semantic Image Inpainting with Perceptual and Contextual Losses”, 2016.

[14] Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. "Image inpainting." In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417-424. ACM Press/Addison-Wesley Publishing Co., 2000.

[15] Efros, Alexei A., and Thomas K. Leung. "Texture synthesis by non-parametric sampling." In *iccv*, p. 1033. IEEE, 1999.