Analysis of Image Generation by different Generator in GANs

Jiabei He\textsuperscript{1,a,}\textsuperscript{†}, Yuzheng Nie\textsuperscript{2,b,}\textsuperscript{†} and Ziwen Mao\textsuperscript{3,c,}\textsuperscript{†}

\textsuperscript{1} School of Arts and Science, Lehigh University, Bethlehem, Pennsylvania 18015, USA;
\textsuperscript{2} School of Physical Science, University of California—Irvine, Irvine, California, 95617, USA;
\textsuperscript{3} Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, 999077, China

\textsuperscript{a}jih224@lehigh.edu; \textsuperscript{b}yuzhengn@uci.edu; \textsuperscript{c}19079602d@connect.polyu.hk

\textsuperscript{†}These authors contributed equally.

Abstract. GAN is very useful in the field of image generation. Many related GANs have been proposed for generating images from the description. However, the research about analysis of image generation by different Generator in GANs is still insufficient. In this paper, different methods such as CNN and Resnet are used as the Generator in GANs for image generation. The subjective evaluation is used to analyze the performance of different generators of GAN. The result shows that the CNN-based generator performs better than the Resnet-based generator. And the increase of the number of parameters in the model will change the image quality. The analysis of different methods as the generator in GAN has a great referential significance in the field of image generation.

1. Introduction
Image generation has been growing at an unprecedented rate over the last several years, and its effects are being felt throughout our everyday life. Nowadays, scanners, digital cameras, displays, and printers are available at relatively inexpensive prices for commercial and consumer applications [1].

The holistic “gist” of images generated by traditional approaches is not natural enough. For instance, the foreground objects in these images tend to be deformed, blended into the background, and seem to be unnatural and irrecognizable [2]. Especially, Generative adversarial networks (GANs) proposed by Goodfellow et al. [3] have shown significant promise as generative models for natural images. A flurry of recent work has proposed improvements over the original GAN work for image generation [4-9].

However, the research about the design of generator in GANs is still insufficient. Therefore, the analysis of different generator designs has a great significance for further improving the performance of GANs in the field of image generation. Maybe the parameters or structure of the model for the design of generator in GANs would influence the effect of image generation. How can the above conditions influence the effect? Does the quality of the generated images have some relationship with the structure or parameters of the model? It is insteseting and useful to explore these problems. In this research, different deep learning models have been adopted to design the different generators of GANs, and evaluate the performance of GANs-based method for image generation by changing the size of dataset.
The rest of this paper is that Section 2 introduces the method and the experiment setup of this research; Section 3 introduces the results and the discussion; Section 4 gives the conclusion.

2. Method

In this paper, GANs are used to generate the target images. GAN includes two main parts: generator and discriminator. Details of this framework can be found in Figure 1.

![Image 1](image1.png)

**Figure 1. The Framework of GANs Generating Images**

In figure 1, the Feature encoder network $E$ contains multiple layers to generate the featured images. Then the images pass through the pooling layer to get labels and features. The generator $G$ translates featured images to advanced images, while the discriminator distinguishes the original images from the generated images.

2.1. Image Generation based on GAN

In the generative networks, the generative adversarial networks (GANs) proposed to implement an alternated training scheme motivated by adversarial characters—the generator and discriminator—attack each other to benefit themselves [11]. Hence, in this subsection, we are going to introduce the generators and discriminators used in this experiment.

2.1.1. Generators.

The Generators used in this paper includes CNN and Resnet, which are shown in Figure 2.

![Image 2](image2.png)

**Figure 2. The Framework of Generators Producing Images Through Multiple Layers**
1) ResNet

Recently proposed residual networks (ResNets) can deal with the training of extremely deep networks up to more than 1000 layers [10]. To generate realistic images, ResNets with deep layers proposed recently have been proved beyond the other shallow approaches. In this paper, ResNet has been used to know that if there are lots of layers in the network, the features can be extracted from different levels. However, the deep network can get more abstract features and semantic information. For the problem of "decreasing accuracy as the network deepens", ResNet provides two options: identity mapping and residual mapping. If the network has reached the optimal level and continues to deepen the network, the residual mapping will be pushed to 0. As result, only identity mapping is left. Theoretically, the network has been in the optimal state, the performance of the network will not be impaired as the depth increases.

2) CNN

Convolutional neural networks (CNNs) have been shown to succeed on many computer vision tasks, such as image classification, detection and segmentation[13]. In most cases, a task solved by CNNs contains image generation from raw sensor inputs to some kind of concise output representation, such as object identity, position or scale[14]. In this paper, the main research for accident image classification is to predict which category it belongs to for a given image. The image is a 3-dimensional array, and the array elements are integers ranging from 0 to 255. The size of the array is width x height x 3, where the 3 represents the three color channels of red, green and blue.

2.1.2. Discriminators

Two multi-layer perception neural networks are used as the Conv2D generative filters which include a tensor of outputs and tasks care of all convolutional layers. Dropout layers ignore some neurons while training to fight against the overfitting. Flatten layers are used since we want to make the output linear to pass to a Dense layer. Moreover, to compare the performance of GANs by changing the structure of the discriminator, we add more layers and use the sigmoid function as the last layer.

2.2. Experiment

In this subsection, we will illustrate design of the experiment and the method we use to generate the dataset.

2.2.1. Dataset

The dataset is used from the mnist dataset. MNIST is a handwritten digit database 11 that contains plenty of training instances and test instances. Mnist is written by different persons to avoid correlations. There are 10 classes, and the images in the dataset are distributed into those classes. There is one digit in each class from 0 to 9. Each class has approximately the same number of instances [15]. In this experiment, we separate the dataset as train and test. We also separate the labels and images. In order to visualize the numbers in the dataset, the matplotlib has been adopted for assistance.

2.2.2. Experiment Setup

The details of the structure for GANs are shown in Table 1.

| Table 1. The Main parameters of Generators |
|-------------------------------------------|
| Original Parameter Configuration | Value | Adjusted Parameter Configuration | Value |
|-------------------------------------|--------|----------------------------------|--------|
| Parameters Numer of Dense           | 1254400| Parameters Numer of Dense        | 1266944|
| Parameters Number of Conv2DT        | 204800 | Parameters Number of Conv2DT     | 911873 |
| Batch Normalization                 | 256    | Batch Normalization              | 128    |
| Total Parameters                    | 2330944| Total Parameters                  | 2229889|
In this experiment, not only the parameters of structure have been adjusted, but also the model of Generator has been changed to analyze the performance of GANs for image generation. Firstly, we add a Conv2DTranspose layer with 32 output filters in the convolution and the kernel size of (3,3). Then, we modify the strides from (1,1) to (5,5) in 128 Conv layer and modify them from (2,2) to (3,3) in 64, 32 and 1 Conv layer. We also use the bias in modified version. Besides, instead of using LeakyReLU functions, activation functions have been adopted to improve the results. We also add a UpSampling2D layer to the convolutional neural network.

3. Results and Discussion

3.1. Results of CNN-based GAN for image generation

The loss of GANs could show the performance of the model. After changing the structure of CNN designed in Generator of GANs, the loss comparisons are shown in Figure 3.

![Figure 3. Loss Comparison of different Generators for Image Generation. (a) the loss of adjusted CNN. (b) the loss of the original CNN model.](image)

In Figure 3, the red line represents the discriminator loss. The blue line represents the generator loss.

From the Figure 3, the loss trend arrives stably after adjusting operation for CNN as the generator in GANs. Obviously, the adjusting operations could improve the performance of the generator, which could arrives stably faster than the original model.

![Figure 4. The Generated Image by CNN and adjusted CNN. Subfigure; (a) the result of the original CNN; (b) the result of the adjusted CNN](image)
In Figure 4, obviously, the original CNN performs better than the adjusted CNN. From Table 1, the total parameters of the original CNN are more than that of adjusted CNN. This may be the reason why the original the CNN performs better than the adjusted as the Generator of GAN.

3.2. Results of Resnet and CNN for Generator

![Figure 5. The Comparison of ResNet training loss and CNN training loss. (a) the training loss of ResNet; (b) the training loss of original CNN model.](image)

The blue line represents the discriminator loss and the red line represents the generator loss. The results show that the loss of original CNN model seems more stable than the adjusted. This may be because the complicated structure and less number of parameters make the resnet-based Generator have lower performance than CNN-based.

![Figure 6. The Generated Image by GANs and ResNet. (a) the image generated by ResNet; (b) the image generated by the original CNN.](image)

In Figure 6, the result shows that the ResNet-based generated image is worse than the CNN-based generated image. This is because the CNN-based method has a better performance of the ResNet-based method.
The results in this paper suggest that the training loss of a dataset tends to be more stable by adjusting CNN models. Specifically, we change the parameters of the generator and the discriminator and also the model of generator. The change in training loss shows the evident improvement of the performance of the generator. However, the generated images of the original CNN perform better than the adjusted CNN which is likely because of the difference in the parameters. We also compare the performance of ResNet and CNN for the generator and the result shows that the training loss of CNN is more stable than the modified. Furthermore, the images generated by GANs and ResNet exhibit that ResNet performs worse than CNN.

In summary, the structure and parameter number can influence the results of image generation a lot. The more parameters there are, the better-the Generator in GAN would perform.

4. Conclusion
In this paper, we analyze how different generators perform in image generation by GANs and how they improve it. We use different deep learning models to test the performance of different generators and change the size of dataset to examine the results of image generation.

The results illustrate that different models and parameter numbers can affect the performance of image generation in GANs. These results may be used to decide what models, parameters or generators could be used in different scenarios. This research has a great referential significance in the field of generator design in GANs for image generation.

References
[1] Vrhel M-, Saber E-, Trussell H J-. Color image generation and display technologies. IEEE Signal Processing Magazine, 2005, 22(1), pp. 23-33.
[2] Yang J-, Kannan A-, Batra D-, et al. LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation. ICLzR 2017
[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, 2014. pp. 2672–2680
[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015
[5] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in neural information processing systems, pp. 1486–1494, 2015.
[6] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. arXiv preprint arXiv:1606.03498, 2016.
[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. arXiv preprint arXiv:1606.03657, 2016.
[8] I Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In European Conference on Computer Vision, pp. 597–613. Springer, 2016.
[9] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126, 2016.
[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, Advances in neural information processing systems, pp. 2672–2680, 2014.
[12] Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.
8798-8807. 2018.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[14] Dosovitskiy, Alexey, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1538-1546. 2015.

[15] Tabik S, Peralta D, Herrera-Poyatos A, Herrera F. A snapshot of image pre-processing for convolutional neural networks: case study of MNIST. International Journal of Computational Intelligence Systems. 2017,10(1), pp. 555-68.