ISAGA 2022 53rd Anniversary Conference
Northeastern University
College of Arts, Media and Design

DC-Art-GAN: Stable Procedural Content Generation using DC-GANs for Digital Art

Rohit Gandikota
gandikota.ro@northeastern.edu

Nik Bear Brown
ni.brown@northeastern.edu

Northeastern University
Boston, MA

Northeastern University
Khoury College of Computer Sciences

AI Skunkworks
Research and Develop Artificial Intelligence

Northeastern University
College of Engineering

Abstract

- The work mainly focuses on using the Deep Convolutional Generative Adversarial Network (DC-GAN) and explore the techniques to address the common pitfalls in GAN training.
- We compare various architectures and designs of DC-GANs to arrive at a recommendable design choice for a stable and realistic generation.
- The main focus of the work is to generate realistic images that do not exist in reality but are synthesised from random noise by the proposed model.
- We also show how training image preprocessing plays a massive role in GAN training.

Introduction

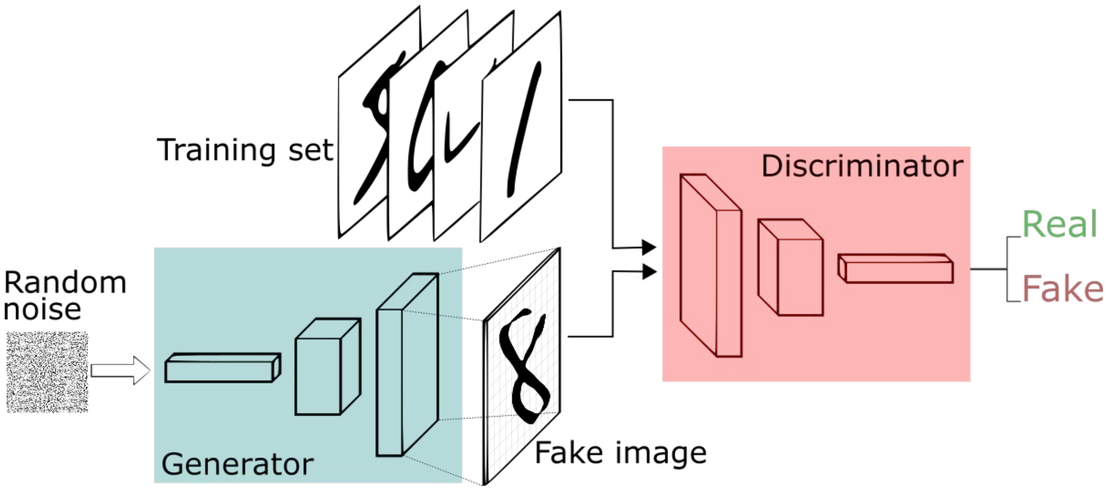


Image Credits: [Medium Article by Thales Silva]

1. No direct training of generator
2. Non-conditional generation (random latent inputs)
3. Deep convolutional generator and discriminator
4. Min-max game between generator and discriminator

Dataset

[Choi, Yunjey, et al. "Stargan v2: Diverse image synthesis for multiple domains." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.]

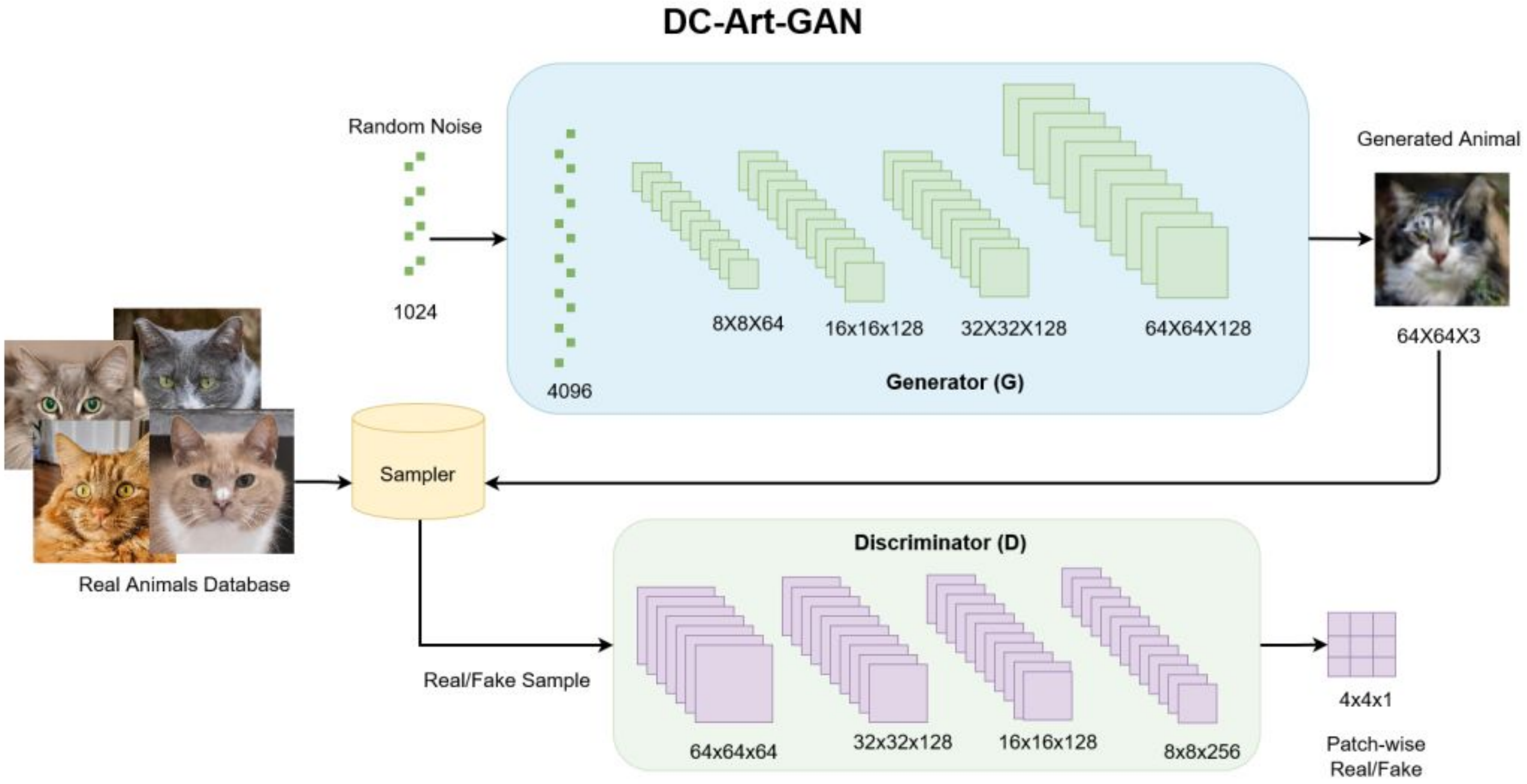
Dataset Preview

About This Data

This dataset, also known as Animal Faces-HQ (AFHQ), consists of 16,130 high-quality images at 512×512 resolution.
There are three domains of classes, each providing about 5000 images. By having multiple (three) domains and diverse images of various breeds per each domain, AFHQ sets a challenging image-to-image translation problem. The classes are:

- Cat;
- Dog;
- Wildlife.

DC-Art-GAN


DC-Art-GAN
Random Noise
1024
4096
8X8X64
16x16x128
32X32X128
64X64X128
Generator (G)
Generated Animal
64X64X3
Real Animals Database
Sampler
Real/Fake Sample
Discriminator (D)
64x64x64
32x32x128
16x16x128
8x8x256
4x4x1
Patch-wise
Real/Fake

Training Details

1. Leaky-ReLU for both Discriminator and Generator (More variance at output)
2. 3 epochs of Generator learning for 1 epoch of Discriminator learning (Improved min-max game, else generator had exploding training loss while discriminator got too good)
3. Learning rates at 0.0002 for generator and 0.00002 for discriminator (again to boost generator's learning so as to maintain a fair fight)
4. Used simple binary cross entropy (Should try MSE and MAE losses)

```
def define_gan(g_model, d_model):
    d_model.trainable = False
    model = Sequential()
    model.add(g_model)
    model.add(d_model)
    opt = Adam(lr=0.00002, beta_1=0.5)
    model.compile(loss='binary_crossentropy', optimizer=opt)
    return model
```

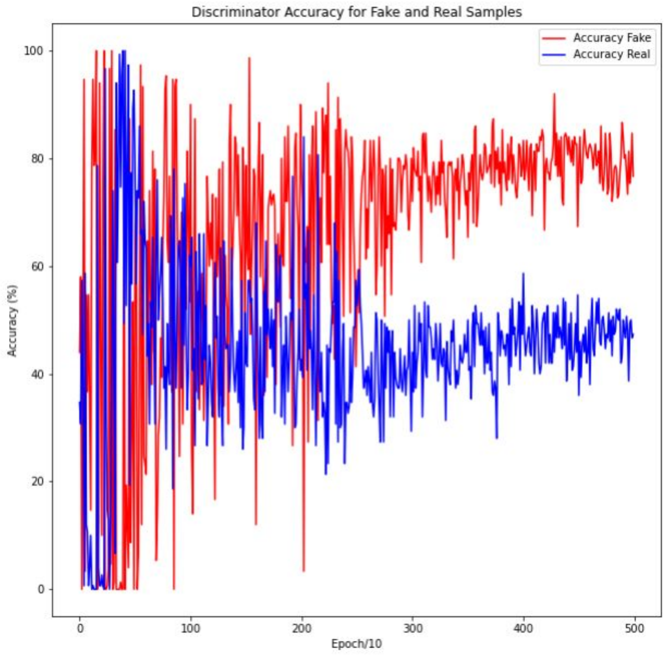
Pitfalls: Mode Collapse

Model Settings

1. ReLu loss
2. Same learning rate for discriminator and GAN
3. 1 epoch generator learning for 1 epoch of discriminator in each training session.
4. GAN generating same output for various random latent inputs (mode collapse ?)

- Adding Leaky ReLU activation at the discriminator will help boost the classification training as negative activations can be perceived as punishments bringing a reward-punishment based training system.
- Reducing the learning rate of the discriminator to avoid local minima convergence.
- Increasing the batch size of training with batch training to bring more diversity to the training process, pushing the discriminator out of converged local minima.

Pitfalls: Poor Generator Performance



Model Settings

1. Leaky ReLu at both generator and discriminator
2. 3 epochs of generator training for 1 epoch in discriminator
3. Faster learning rate for generator
4. Same facial expression, with varied background and fur
5. Face alignment and direction is same

This can be addressed with a few possible measures,

– Increase learning rate of generator for faster learning (not preferable as this may lead to an overshoot of global minima).
– Reduce the complexity of generator architecture for relatively faster convergence. (may lead to depreciation in output quality as we are reducing the model complexity to map higher-dimensional features)
– Removing any complex training layers like dropouts and batch norm layers (preferable as this does not compromise on convergence or quality)
– Ensuring the discriminator is set as non-trainable when training the generator.

Pitfalls: Over Saturated Samples

Model Settings

1. Global distribution normalization.

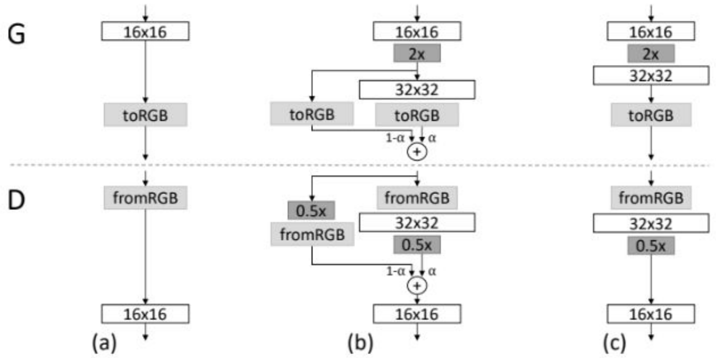
Even though the training curves show a fair min-max game, the generator output saturates due to the input space pre-processing during training, as shown in Fig. 4. It is essential to understand the pre-processing in the information flow domain. When global pre-processing is done, each image is normalized based on the statistical parameters of the entire dataset. This means the information of the dataset is biasing that information of a sample. This plays a huge role, especially in GAN training, as the generator learns information a transformation from random noise to the distribution of the dataset rather than a pixel to pixel mapping.

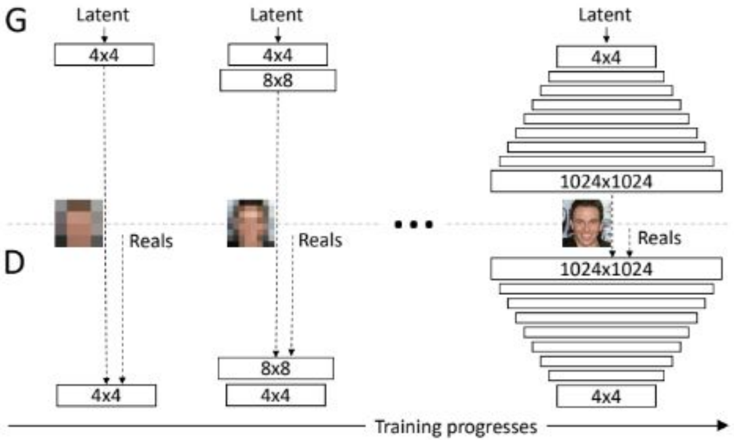
Due to the reasons mentioned above, sample-based normalization independent of dataset parameters will hugely improve the quality of output and more artistic variations, as shown in Fig. 5.

Code Repo: *https://github.com/aiskunks/AI_Research/tree/main/dc-gan-best-practices*

Challenges: Longer Training Times and Lower Resolutions

Fading in New Layers

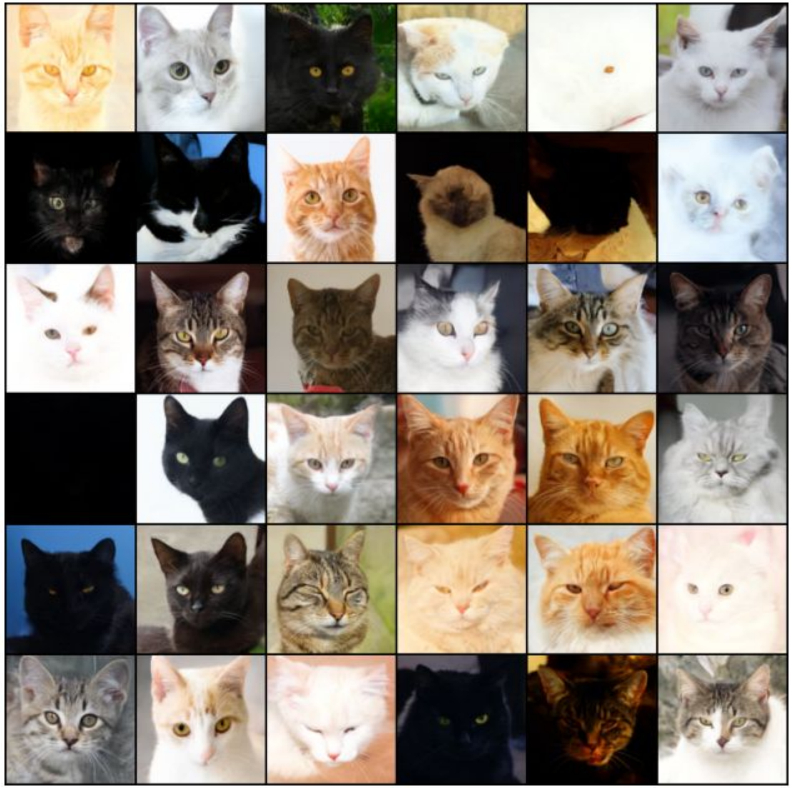




***Code Repo:** https://github.com/aiskunks/AI_Research/tree/main/growing-gan-best-practices*

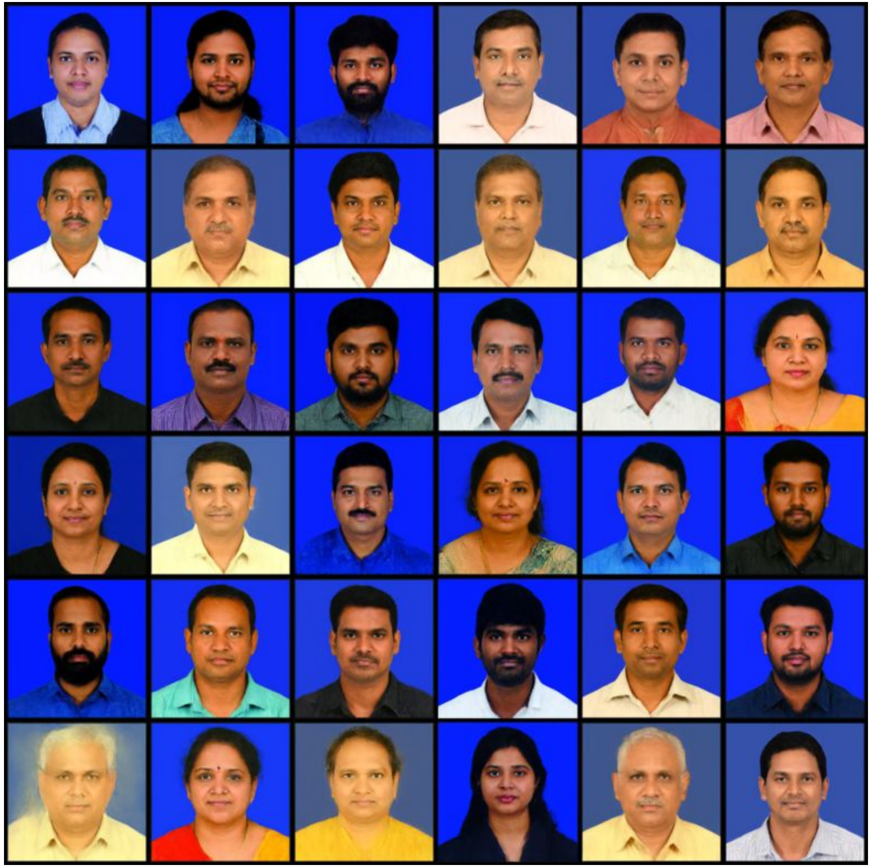
Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improvedquality, stability, and variation. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=Hk99zCeAb

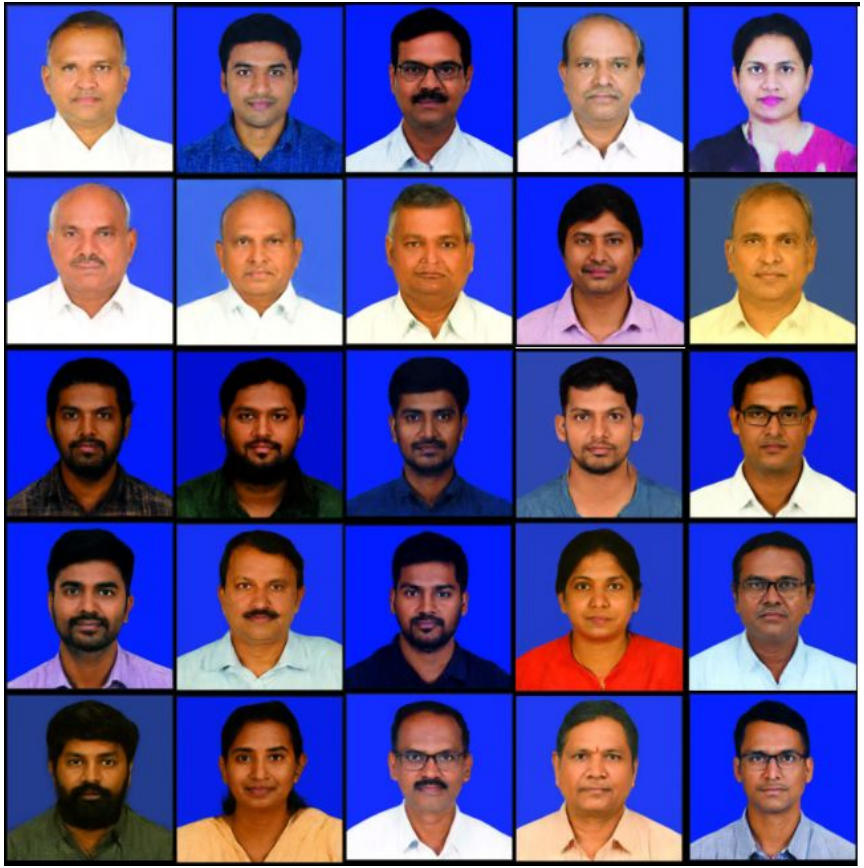
Final Samples



Final Samples (Extended)

Final Samples (Extended Work)





Thank You

Any questions?

(gandikota.ro@northeastern.edu)

References

1. Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
3. Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
4. Gandikota, R., Sharma, A., ManjuSarma, M., Bothale, V. M. (2020). RTC-GAN: Real-time classification of satellite imagery using deep generative adversarial networks with infused spectral information. In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium (pp. 6993-6996). IEEE.
5. Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
6. Brock, A., Donahue, J., Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. International Conference on Learning Representations.
7. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.
8. Gandikota, R., Mishra, D. (2019, December). Hiding Audio in Images: A Deep Learning Approach. In International Conference on Pattern Recognition and Machine Intelligence (pp. 389-399). Springer, Cham.