Exploring Image Generation with Deep Learning: Deep Convolutional Generative Adversarial Networks

ECE 176 – Final Project

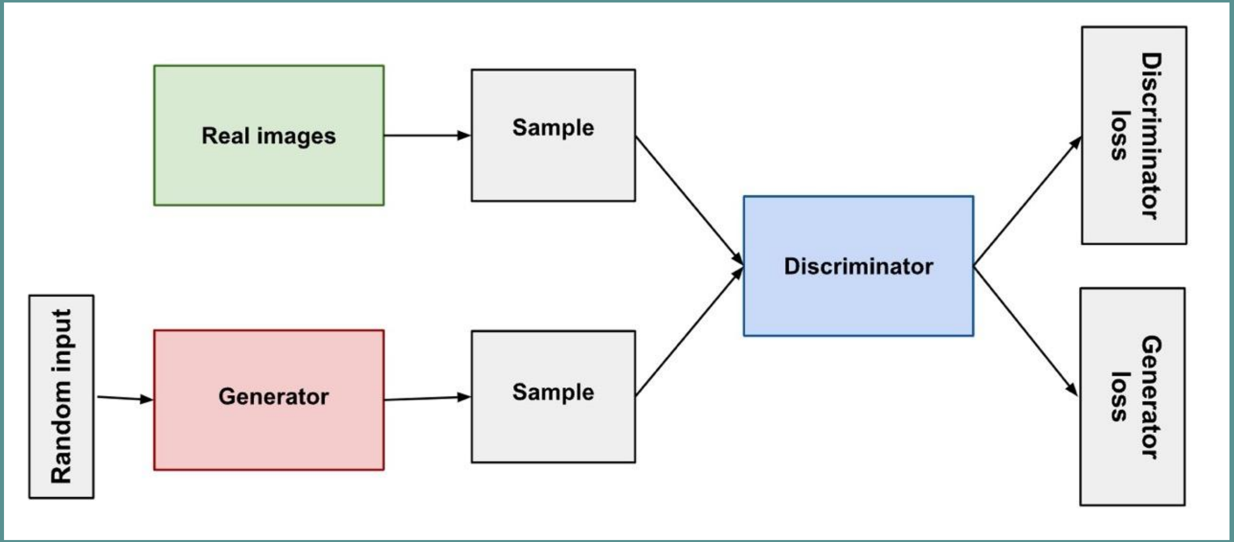
Kenneth Casimiro, Computer Engineering B.S.
Steven Sharp, Electrical & Computer Engineering M.S.

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Background

Generative Adversarial Networks (GAN)

- Generator
- Discriminator
- Adversarial Process
- Creating Realistic Images

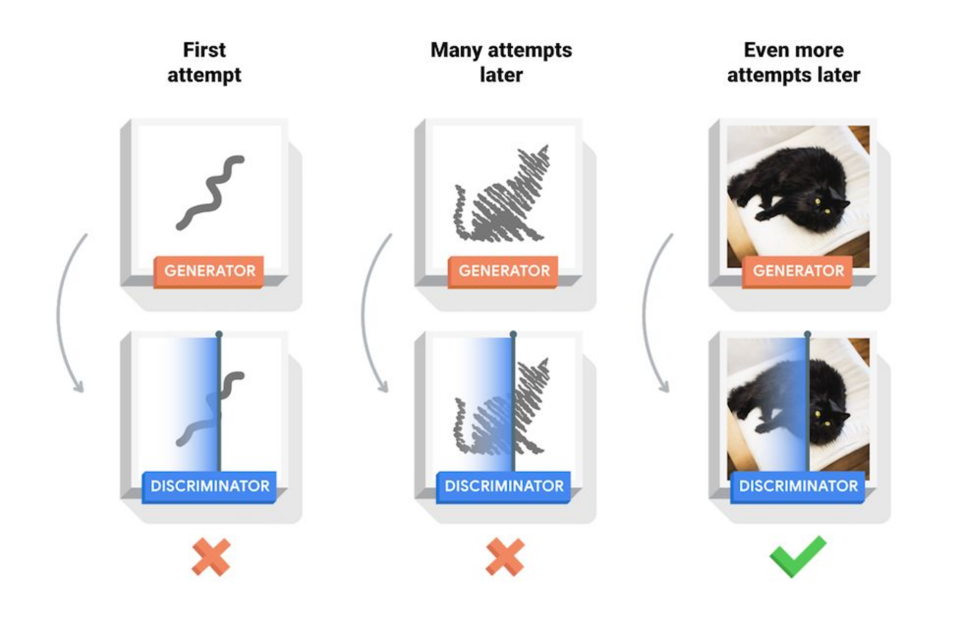


GAN Architecture [4]

Introduction & Motivation

Deep Convolutional Generative Adversarial Network (DCGAN)

- **Revisit the original DCGAN study (*Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,* [2])**
- **Apply its methods to diverse datasets to explore the technique's effectiveness**
- **Goal is to understand its inner workings and identify its strengths and weaknesses.**



DCGAN Example [5]

Approach & Methods

1. **Data Loading & Preprocessing**
2. **Generator and Discriminator Architecture**
3. **Training & Loss Analysis**
4. **Image Generation Analysis**

```
[9/10][2800/3166]       Loss_D: 0.1270  Loss_G: 6.9989  D(x): 0.9127    D(G(z)): 0.0117 / 0.0036
[9/10][2850/3166]       Loss_D: 0.1473  Loss_G: 5.7211  D(x): 0.9611    D(G(z)): 0.0866 / 0.0079
[9/10][2900/3166]       Loss_D: 0.0800  Loss_G: 4.4613  D(x): 0.9724    D(G(z)): 0.0468 / 0.0201
[9/10][2950/3166]       Loss_D: 0.0919  Loss_G: 4.9013  D(x): 0.9352    D(G(z)): 0.0168 / 0.0219
[9/10][3000/3166]       Loss_D: 0.0559  Loss_G: 5.3102  D(x): 0.9578    D(G(z)): 0.0088 / 0.0115
[9/10][3050/3166]       Loss_D: 0.0426  Loss_G: 5.3165  D(x): 0.9767    D(G(z)): 0.0173 / 0.0108
[9/10][3100/3166]       Loss_D: 0.0479  Loss_G: 5.7214  D(x): 0.9764    D(G(z)): 0.0218 / 0.0081
[9/10][3150/3166]       Loss_D: 0.0713  Loss_G: 5.1483  D(x): 0.9479    D(G(z)): 0.0121 / 0.0186
```

DCGAN - Loss Probability

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

DCGAN - Convolutional Layers Architecture

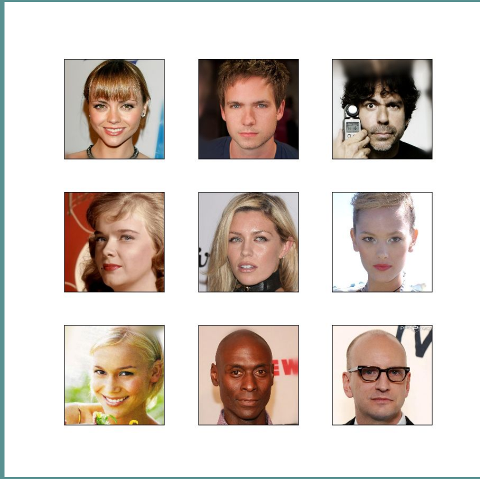
Experiments

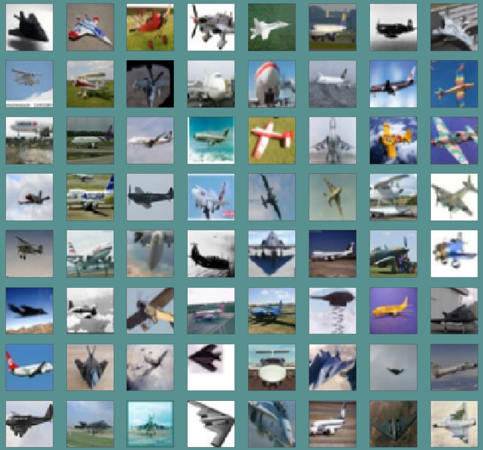
Apply the approach to diverse datasets.

Three Datasets:

1. EMNIST
2. Celeb-A
3. CIFAR-10

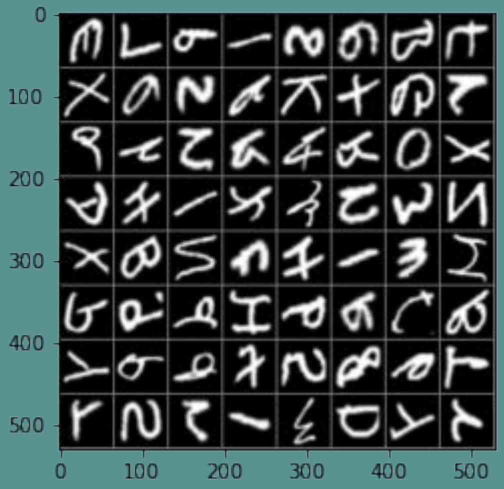
Analyze results through metrics and image comparison.



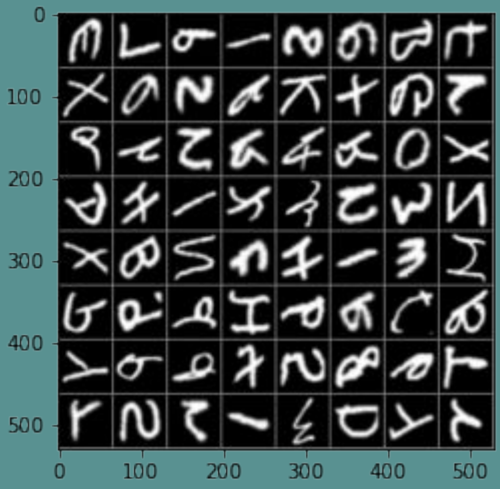


Experiments - EMNIST

- **Real and generated batch of images are identical**
- **Discriminator successfully recognizes real images**
- **Generator struggles to create realistic images**



Real Images



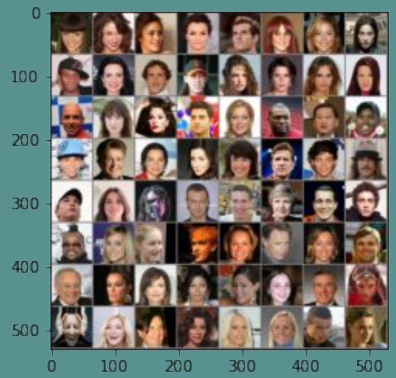
Generated Images

```
[10/10] [700/882] Loss_D: 0.0001      Loss_G: 9.8182  D(x): 1.0000    D(G(z)): 0.0001 / 0.0001
[10/10] [750/882] Loss_D: 0.0000      Loss_G: 12.1525 D(x): 1.0000    D(G(z)): 0.0000 / 0.0000
[10/10] [800/882] Loss_D: 0.0001      Loss_G: 9.9660  D(x): 0.9999    D(G(z)): 0.0001 / 0.0000
[10/10] [850/882] Loss_D: 0.0003      Loss_G: 9.1757  D(x): 0.9998    D(G(z)): 0.0001 / 0.0001
```

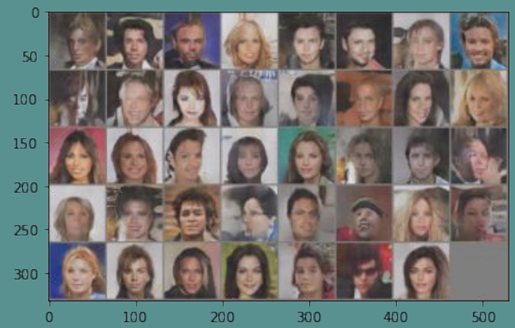
Loss Metrics

Experiments - Celeb-A

- Generated images appear disfigured
- Generator loss is relatively high
- Generator is not producing realistic images to deceive the Discriminator
- Discriminator performs well



Real Images



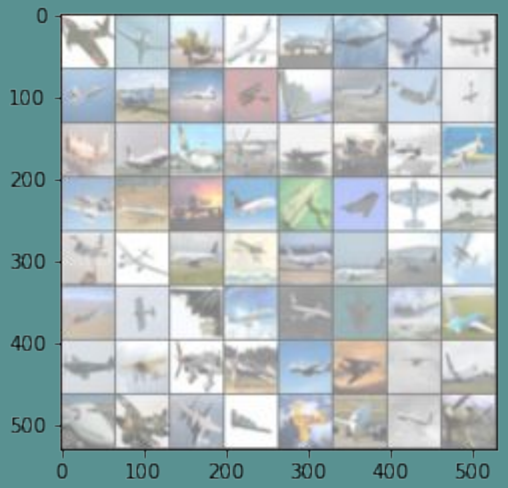
Generated Images

```
[9/10][2800/3166]       Loss_D: 0.1270  Loss_G: 6.9989  D(x): 0.9127   D(G(z)): 0.0117 / 0.0036
[9/10][2850/3166]       Loss_D: 0.1473  Loss_G: 5.7211  D(x): 0.9611   D(G(z)): 0.0866 / 0.0079
[9/10][2900/3166]       Loss_D: 0.0800  Loss_G: 4.4613  D(x): 0.9724   D(G(z)): 0.0468 / 0.0201
[9/10][2950/3166]       Loss_D: 0.0919  Loss_G: 4.9013  D(x): 0.9352   D(G(z)): 0.0168 / 0.0219
[9/10][3000/3166]       Loss_D: 0.0559  Loss_G: 5.3102  D(x): 0.9578   D(G(z)): 0.0088 / 0.0115
[9/10][3050/3166]       Loss_D: 0.0426  Loss_G: 5.3165  D(x): 0.9767   D(G(z)): 0.0173 / 0.0108
[9/10][3100/3166]       Loss_D: 0.0479  Loss_G: 5.7214  D(x): 0.9764   D(G(z)): 0.0218 / 0.0081
[9/10][3150/3166]       Loss_D: 0.0713  Loss_G: 5.1483  D(x): 0.9479   D(G(z)): 0.0121 / 0.0186
```

Loss Metrics

Experiments - CIFAR-10

- Generated images lack a discernable object and appears smudgy
- Discriminator successfully recognizes real images
- Generator struggles to create realistic images



Real Images

```
[5/10][50/79]   Loss_D: 0.4552  Loss_G: 1.8231  D(x): 0.7636    D(G(z)): 0.1608 / 0.1669
[6/10][0/79]    Loss_D: 0.4143  Loss_G: 1.9662  D(x): 0.8073    D(G(z)): 0.1712 / 0.1778
[6/10][50/79]   Loss_D: 0.3774  Loss_G: 1.8868  D(x): 0.8286    D(G(z)): 0.1701 / 0.1548
[7/10][0/79]    Loss_D: 0.3880  Loss_G: 2.0402  D(x): 0.8468    D(G(z)): 0.1869 / 0.1593
```

Loss Metrics

Generated Images

Conclusion

- Successfully re-implemented the DCGAN from the research paper.

- Did not have the best results compared to the research paper.

- Identified what requires further investigation with our experiments to improve our network.

Acknowledgements

This presentation was completed for the final project of ECE 176:Introduction to Deep Learning & Applications.

Project Report and codebase can be found in our [GitHub repository.]

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Kenneth Casimiro
A17132441
Computer Engineering B.S.

Steven Sharp
A13232191
Electrical & Computer Engineering M.S.
(Intelligent Systems, Robotics & Control)

References - Part 1

[1] Goodfellow, Ian, "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.
https://dl.acm.org/doi/pdf/10.1145/3422622}{https://dl.acm.org/doi/pdf/10.1145/3422622

[2] Radford Alec, Metz Luke, and Chintala Soumith. 2015. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." arXiv:1511.06434.
https://arxiv.org/abs/1511.06434}{https://arxiv.org/abs/1511.06434

[3] (n.d.). GAN Loss. Google - Machine Learning. Retrieved March 17, 2023,
https://developers.google.com/machine-learning/gan/loss

[4] Overview of GAN Structure; Google - Machine Learning. Retrieved March 17, 2023,
https://developers.google.com/machine-learning/gan/gan_structure

[5] Deep convolutional generative Adversarial Network - Tensorflow. Retrieved March 17, 2023.
https://www.tensorflow.org/tutorials/generative/dcgan

References - Part 2

[6] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters.
https://www.nist.gov/itl/products-and-services/emnist-dataset

[7] Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou. 2015. "Deep Learning Face Attributes in the Wild". Proceedings of International Conference on Computer Vision (ICCV)
https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[8] Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
https://www.cs.toronto.edu/~kriz/cifar.html