COMP551 Assignment 4

Daniel Dai Kendall Wei Lucy Mao

Reproducibility Summary

Scope of Reproducibility

The claims made by the paper "Unsupervised Representation Learning with Deep Convolutional Generative Adverserial Networks" are as follows. It proposes that the model architecture outlined in the paper provides constraints to building a DCGAN model that is stable to train in most settings. Additionally it performs supervised classification tasks to validate the efficacy of the discriminator as a feature extractor. Finally through a number of experiments, it provides visualizations that further explore the properties and capabilities of both the discriminator and generator [10].

Methodology

The code for the DCGAN model was written following a tutorial by PyTorch [7]. Most members of the team trained on a far smaller dataset than the ones presented in the paper due to the constraints of hardware and resources. Training on most of the datasets took around 15-25 minutes on Google Colab's T4 GPU.

Results

Many visualization tasks were subject to the viewer's judgments; overall, the visualization experiments produced similar results, but with the smaller datasets that the models were trained on there were discrepancies in quality. The experiments run on CIFAR-10 showed similar, albeit, slightly worse results as well (just around 20% lower). In total, most experiments showed the viability of the architecture and parameters proposed in the paper but only within the capabilities of the machine upon which the experiments were reproduced.

1 Introduction

The paper proposes a new way to use Convolutional Neural Networks (CNNs) for unsupervised tasks, using convolutional layers both in the discriminator and generator of a Generative Adversarial Network (GAN). The paper puts forth an architectural topology along with a list of parameters that provide stability to training this new model [10].

Due to the nature of GANs and their inherent instability much of the paper focuses on how the model they put forth shows robustness in manipulation and generalization along with how well the model can used in classification against other benchmarks.

2 Scope of reproducibility

The following are the claims that will be tested in this report

- The architecture and parameters described allowed for stable training in most settings
- The trained discriminator showed competitive performance against other unsupervised algorithms
- Filters from the discriminator have learned to draw specific, meaningful shapes
- Generators show arithmetic qualities allowing for manipulation of objects generated in images [10]

3 Methodology

The authors did not make the code for the model available but many libraries such as PyTorch offers tutorials on setting up the architecture described in the paper [7]. Similarly, the code for the experiments performed in the paper was also

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

not published by the authors so the experiments were created from research or purely from scratch. Ablation studies – purely for the assignment and not mentioned in the paper were also either researched elsewhere or done from scratch.

3.1 Model descriptions

The guidelines for the model as outlined in the paper follows: The pooling layers that would downsample the data in a regular discriminator of a GAN are replaced with strided convolutional layers. Fractional-strided convolutions are used to upsample in the generator. BatchNorm was added in both the discriminator and generator for further stabilization to avoid vanishing or exploding gradients. The architecture also contains no fully connected layers, decreasing the number of parameters that need to be trained. ReLU activation was used in all layers excluding the output in the generator, which used Tanh, hence why all images that are inputted are normalized between [1,1]. In the discriminator, leaky ReLU activation was used, the small negative gradient allowed a stable flow of gradients back through the whole network [10].

The models were trained on several different datasets than those proposed in the paper, due to computational constraints. Notably, Imagenet-1k, was far too large to properly train on so a member of the team trained on a subset of the data [9].[2] As mentioned before the dataset of faces that were used in the paper were also not available publicly so the model trained on CelebA in our implementation [10]. For brevity, further deviations from the datasets seen in the paper were also due to computational constraints.

On a high level, a vanilla GAN and the DCGAN work very similarly, where the generator outputs a fake image that is passed to the discriminator. The discriminator then gives a probability that the image it received was real, using binary cross entropy as the loss function. The discriminator, in one step, will estimate whether the image inputted (x_i or $G(z_i)$) is either real ($y_i = 1$) or fake ($y_i = 0$)[5].

$$L_D = -\frac{1}{m}\sum_{i=1}^{m}[y_i \times log(D(x_i)) + (1 - y_i) \times log(1 - D(G(z_i)))]$$

While the generator is trying to create an image $G(z_i)$ such that it can "trick" the discriminator into thinking the image is real, such that

$$L_G = -\frac{1}{m}\sum_{i=1}^{m} log(D(G(z_i))$$

is minimized [5]. This essentially means that inputting an image generated by the generator will result in the discriminator outputting a value closer to 1.

Now, why convolutional layers are used instead of fully connected layers—a point about the number of parameters were already brought up but CNNs also possess the ability to learn spatial relations in images that will be further supported in the experiments.

3.2 Datasets

The DCGANs were trained on three datasets: LSUN (Large-scale Scene Understanding), a newly assembled Faces dataset, and ImageNet-1k. Additional training was done on the MNIST dataset to further explore DCGANs' capability.

LSUN Bedroom

The LSUN Bedroom dataset consists of 126227 images, each with a resolution of 64 x 64 pixels.[8] This is a 20% sample of the bedrooms category in the LSUN dataset. The Large-scale Scene Understanding (LSUN) dataset is designed to set a new standard in large-scale scene classification and understanding. It contains 10 scene categories including dining room, bedroom, church, etc. Each category contains around 120,000 to 3,000,000 images for training, 300 images for validation and 1000 images for testing.

CelebA

The CelebFaces Attributes Dataset (CelebA) is a large-scale face attribute dataset commonly used for computer vision tasks. It contains over 200,000 celebrity images, each annotated with 40 distinct attributes.[1] Each image is annotated by 5 landmark locations for facial features and 40 binary attributes indicating different physical features. This makes it a valuable resource for tasks in facial recognition and analysis, such as face attribute recognition, face detection, and landmark localization. The dataset was originally collected from the internet by the MMLAB, The Chinese University of Hong Kong.

ImageNet-1k

The ImageNet dataset serves as a critical benchmark for deep learning and training advanced computer vision models. It's organized according to the WordNet hierarchy where each concept is described by word phrases called "synset".[2]

The dataset is sourced from the internet and designed to illustrate each synset with approximately 1,000 images. It contains 1000 object classes and includes 1,281,167 training images, 50,000 validation images, and 100,000 test images. This variety in images and scenarios makes ImageNet a crucial resource for image classification tasks. For instance, it is employed as the official dataset for the annual ImageNet Large Scale Visual Recognition Challenge, a competition that has profoundly impacted the field of computer vision, especially in object category classification and detection.[9]

MNIST

The MNIST(Modified National Institute of Standards and Technology) dataset is a subset of the NIST dataset containing 70,000 28 * 28 black-and-white handwritten digits[6]. It consists of 60,000 training images and 10,000 testing images, evenly separated into 10 classes. The dataset serves as a valuable testbed for experimenting with pattern recognition methods or machine learning algorithms, offering the advantage of requiring minimal effort in preprocessing and formatting.

3.3 Hyperparameters

The hyperparameters used in the assignment were taken from the paper itself, since one of the main findings was the parameters of the model, so further search was required during the reproduction. The parameters are as follows [10]:

- Images were normalized between [-1,1]
- Training was done with MBSGD with batch size = 128
- The initial weights were initialized to $\mathcal{N}(0, 0.02^2)$
- The slope for Leaky ReLU was 0.2
- Adam optimizer
- Learning rate of 2e-4
- The momentum coefficient was 0,5
- Image sizes generated were 64 by 64
- Dimension of the latent space: 100
- Number of convolutional layers: 5

Many of the parameters chosen were broadly backed up by the experiments that were run in the paper although they explicitly mentioned that the default momentum coefficient of 0.9 resulted in "training oscillations" [10]. Further ablation studies also showcase how the parameters listed in the paper and reiterated above were the best.

3.4 Experimental setup and code

Most of the ablation studies used loss as a metric for performance. Many of the ablation studies sought to challenge the proposals put forth in the paper and dissect how far performance would fall if any modification was made directly against what was listed. The setup for many of the experiments involved training a different model with slight modifications to its architecture or the hyperparameters that were listed.

The experiment with classification on Cifar-10 used accuracy. The set up of the experiment followed what was listed in the paper. For every convolution in the discriminator, a maxpooling layer was added to produce a 4*4 spatial grid. At the end, "the features were flattened and concatenated to form a 28672 dimensional vector and a regularized linear L2-SVM was trained... [was] trained on top of them" [10].

The visualization experiments were done qualitatively as the paper did not give strict numerical results to compare against. The exploration of latent space was set up using two unique points in latent and splitting the distance between the points in 9 equal steps. The experiment was performed 5 times with 5 different pairs of points to get a broader view of the space. The visualization of the features filtered by the discriminator was set up to present the first 6 features of the last convolutional layer, this once again was repeated for 10 different images. The experiment for vector arithmetic performed the calculation on the same set of input images 10 times.

4 Results

The results from the experiments all supported the claims and proposals made by the paper. All further tuning of parameters and modifications to the architecture of the model produced worse results than the guidelines provided.

Additionally, the classification experiment provided promising, although lower accuracy. This is explained by the fact that the discriminator was only able to be trained on a very small subset of the data compared to the model measured in the paper.

The visualization studies showed very similar results to what was discussed in the paper. Once again due to the small training size the results were not as clearly presented as those in the paper.

4.0.1 Result 1: Unsupervised classification

As seen in table 1 the training accuracy was very high and test accuracy was far above average. This meant that although there might have been some over fitting the discriminator was able to be used as a feature extractor for unsupervised classification tasks.

Train set accuracy	Test set accuracy
99.406%	65.270%

Table 1: Accuracy of DCGAN used as an unsupervised classifier on Cifar-10. Shows that it is viable as a classifier although the test accuracy was around 20% lower than the results in the paper, which could be due to training set size.

4.0.2 Result 2: Discriminator Visualization

The result shown in figures 1 show once again show that the discriminator was able to learn meaningful shapes and contours of images that make up the "real" images. Juxtaposed against a discriminator that has its weights randomly initialized further proves its capabilties.

Figure 1: Visualization of filters

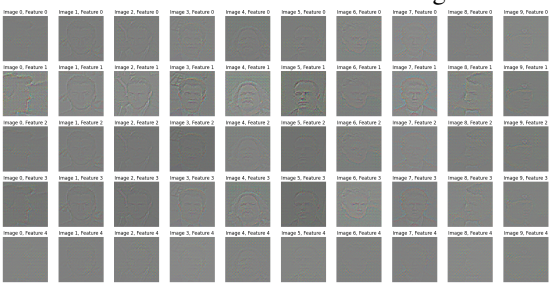

Figure 2: The first 5 trained filters of the final convolutional layer of the discriminator. Shown that there are meaningful shapes and lines learned by the discriminator [10].

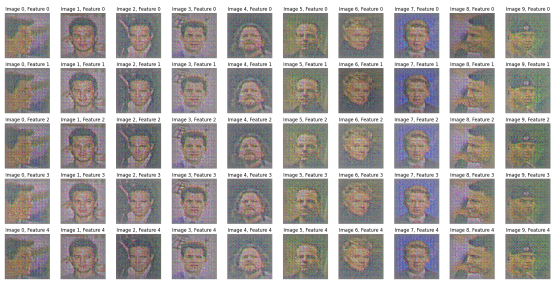

Figure 3: Random filters on the same set of photos. Very little to no discrimination between the pixels of the images [10].

4.0.3 Result 3: Latent Space Exploration

Starting from different points in latent space and walking from one point to another displays the drastic differences between the points 4. The additions or removals of shapes and entire parts of the image show that there is a heterogeneity of what can be produced and the generator is not simply memorizing a specific image that might be able to trick the discriminator.

4.0.4 Result 4: Generator Manipulation

Similar to word vectors, the latent vectors that serve as the base for generating an image can be manipulated with simple arithmetic. In the figure 7, it is seen that you can take a "smiling woman" and subtract a "neutral women" then add a "neutral man" to get a smiling man as a result.

4.1 Results beyond original paper: ablation experiments

Further experiments were done to challenge the results of the paper 2. The motivation for most if not all of the experiments were to understand the choices that the paper made, since many of the decisions were not clearly expanded upon.

Figure 4: Visualization of different points in latent space

Figure 5: Picking 5 different pairs of points in latent space and walking between the two in 9 uniform steps. The shapes of the churches are changing drastically as you go from one point to another

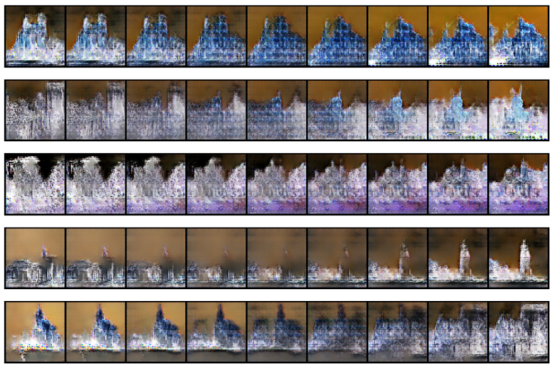

Figure 6: Similar results with faces, especially notable that there is change of hair, gender and orientation of the faces. Once again shows that there is little memorization of data learned by the generator

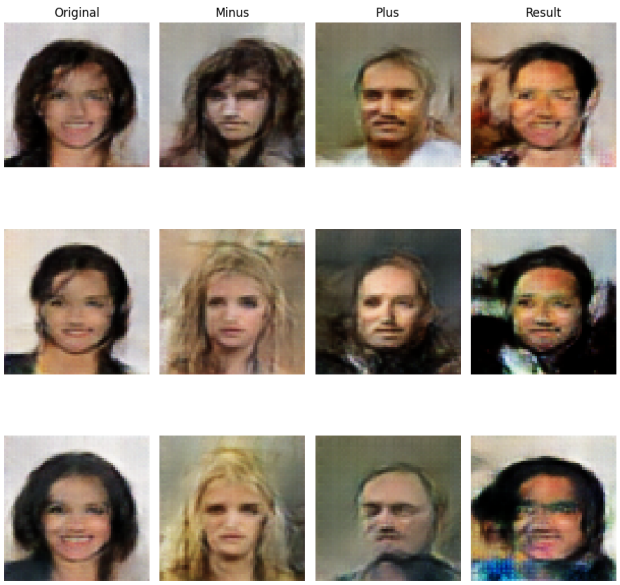

Figure 7: Original is an image of a image generated by a latent vector that learned to generate smiling women, subtracting an image of a neutral women, adding an image of a neutral man. The result is an image of a smiling man.

Notably the choice to replace the pooling layers with strided convolutions was brought up in "Striving for Simplicity: The All Convolutional Net" [11], of which this paper also adopted. For the experimentation with dense layers, the paper states that "the first layer...could be called fully connected" and "the last convolutional layer is flattened and fed into a single sigmoid layer" [10]. So, modification were made to turn these into the fully connected layers. Tanh was used as the activation function in the output layer of the generator to "quickly saturate and cover the color space or trianing distribution", as Tanh is bounded bounded between [-1,1] which the data has been normalized to. The choice of using leaky ReLU in the discriminator was to work better with "higher resolution modelling" [10].

5 Discussion

As seen in the ablation studies the parameters and structure presented in the paper are the best ones that have been found in our limited search. The difficulty of balancing the training and loss of the generator and discriminator led to a lot of the issues with performance as changes were made. This is due to the fact that GANs inherently are very sensitive to little changes and finding the parameters to allow for proper training is very difficult, without very extensive searching

Modification	Discriminator Loss Mean	Generator Loss Mean	Discriminator Loss Lowest	Generator Loss Lowest
LR = 0.0001	0.4288	2.9485	0.1112	0.0010
LR = 0.00001	0.0901	3.9478	0.0225	2.3076
Beta = 0.9	0.0604	9.3882	3.9020	0.7723
Latent space dimension = 10	0.0034	51.3121	1.8072e-27	12.9624
Latent space dimension = 1000	1.4002e-11	69.5420	2.0620e-31	69.1848
Pooling layer	0.9232	1.5402	0.0855	0.6468
Fully connected layer	0.6347	3.1991	0.0123	0.0101
ReLU in discriminator	0.7094	2.9053	0.1499	0.0055
Sigmoid in generator	3.1770e-12	62.1561	9.9577e-28	62.1007
ReLU in generator	1.8046e-8	44.9723	1.3315e-22	36.7360
32*32 output dimension	0.8215	1.8039	0.0743	0.0549
128*128 outputs dimensions	0.0666	5.0097	0.0026	1.3692

Table 2: Mean and best loss of generator and discriminator. It is seen that generator loss is consistently higher than discriminator.

it would be difficult to provide better choices than those provided in the paper. Oftentimes, it's seen in our experiments that the imbalance results in a very high generator loss and low discriminator loss. Other papers like Wasserstein GAN [3] have shown that there are modifications that can be made to further stabilize GANs, although it should be noted this paper was published before Wasserstein GAN, but it would be interesting to see how an addition of Wasserstein loss could further stabilize the training process.

Leveraging insights from our DCGANs experimental analysis, the potential for future explorations and enhancements presents exciting opportunities to address key challenges and broaden their applications. The study addressed the remaining issues of model instability, particularly the tendency of models to exhibit oscillating modes when trained for extended periods. A potential improvement could involve integrating reinforcement learning techniques with DCGANs to further enhance their learning efficiency and stability. Research suggested that combining DCGANS with deep Q-network (DQN) reinforcement learning helps overcome the issues of unbalanced samples and insufficient samples.[4] The results show that the model achieved state-of-the-art performance and further improved the efficiency of the generator. Additionally, exploring the application of DCGAN to other domains, such as 3D modelling and virtual reality environments, could yield valuable insights and advancements. One study trained GAN architectures including GAN and DCGAN on the smallNORB dataset to generate 3D images. The results show that DCGAN gave plausible and stable results for all the classes while other GAN structures gave undesirable outputs for some classes.[12] This could impact the way we generate and interact with 3D objects and environments, potentially impacting fields like gaming, simulation, and virtual reality.

5.1 What was easy

The popularity of the paper gives rise to many tutorials and implementations that allowed us to easily create the models described in the paper. The outline of the parameters and architecture also made it easy to understand where we can continue exploring in our own ablation studies. The visualization tasks provided clear direction and offered interesting and very reproducible results that we can leverage as well.

5.2 What was difficult

Although the experiments were clearly laid out the lack of code from the authors made it very difficult to set up the experiments and debug the code. Additionally the datasets presented in the paper were either too large for our resources (Imagenet-1k) or inaccessible (faces) [10]. This made it difficult to faithfully recreate their experiments and led to differences in performance results.

6 Distribution of work

Lucy worked on the visualization experiments while Daniel performed the ablation experiments. Kendall performed further training on the different datasets that were listed. Lucy and Kendall wrote the paper.

References

[1] Celebfaces attributes (celeba) dataset.

[2] Amanpreet Singh (apsdehal). imagenet-1k.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[4] A deep learning and reinforcement-learning-based system for encrypted network malicious traffic detection. Jin yang, gang liang, beibei li, guozhu wen, tianyu gao.

[5] Harshit Dwivedi. Understanding gan loss functions, Aug 2023.

[6] Sylvain Gugger. mnist.

[7] Nathan Inkawhich. Dcgan tutorial — pytorch tutorials 2.1.1+cu121 documentation.

[8] Weilin Huang Yuanjun Xiong Yu Qiao Limin Wang, Sheng Guo. Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns.

[9] Hao Su Jonathan Krause Sanjeev Satheesh Sean Ma Zhiheng Huang Andrej Karpathy Aditya Khosla Michael Bernstein Alexander C. Berg Li Fei-Fei Olga Russakovsky, Jia Deng. Imagenet large scale visual recognition challenge.

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[11] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2015.

[12] Analytics Vidhya. Applying generative adversarial network to generate novel 3d images.