IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

AI BASED APPROACH FOR REGULARIZED DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

ADITYA B[1], AKSHAY KUMAR C R[2], MOIN MANZOOR[3], ARYA KARN[4], RAKSHITA[5]

Student, Department of Computer Science & Engineering, Atria Institute of Technology, Bengaluru, India[1,2,3,4]

Assistant Professor, Department of Computer Science & Engineering, Atria Institute of Technology, Bengaluru, India[5]

Abstract: Generative Adversarial Networks, or GAN for short, is a productive modeling model using in-depth learning methods, such as convolutional neural networks.
In recent times, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention.
We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain constraints, and demonstrate that they are a strong candidate for unsupervised learning.
Training on various image datasets, such as anime we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability to be used for various purposes such as advertising.

I. INTRODUCTION

Deep neural networks are used mainly for supervised learning: classification or regression.
Generative modelling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.
In the context of computer vision, one can leverage the practically unlimited amount of unlabelled images and videos to learn good intermediate representations, which can then be used on a variety of supervised learning tasks such as image classification.
GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. There has been very limited published research in trying to understand and visualize what GANs learn, and the intermediate representations of multi-layer GANs.
Our method is an unsupervised approach based on DCGAN which can do good image representations. We train our network through adversarial learning and later reusing discriminator and encoder part of the model as a feature extractor for image classification tasks. For evaluating our model performance as a feature extractor, we add a linear classifier at the top of the network and train it with some labelled data for images classification. The results show that RDCGAN can achieve high accuracy at classification compared to DCGAN and other methods.

II. PURPOSE

• We propose and evaluate a set of constraints on the architectural topology of Convolutional GANs that make them stable to train in most settings. We name this class of architectures Deep Convolutional GANs (DCGAN)
• We use the trained discriminators for image classification tasks, showing competitive performance with other unsupervised algorithms.
• We visualize the filters learnt by GANs and empirically show that specific filters have learned to draw specific objects.
• We show that the generators have interesting vector arithmetic properties allowing for easy manipulation of many semantic qualities of generated samples.

III. RELATED WORK

A. REPRESENTATION LEARNING FROM UNLABELED DATA.

Unsupervised representation learning is a well-studied problem in general computer vision research and also in the context of images. A classic approach to unsupervised representation learning is to cluster on data (for example, using K-means) and use clusters for improved classification scores. In the context of images, hierarchical clustering of image patches

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

(Coates & Ng, 2012) can be done to learn strong image representations. Another popular method is to train autoencoders (convoluted, stacked (Vincent et al., 2010), separating what and where components of the code (Zhao et al., 2015), ladder structures (Rasmus et al., 2015)) encode an image into a compact code. and decodes it to reconstruct the image as accurately as possible. These methods have also been shown to learn good feature representations from image pixels. Deep belief networks (Lee et al., 2009) have also been shown to work well in learning hierarchical representations.

B. GENERATING NATURAL IMAGES

Generative image models are well studied and fall into two categories: parametric and nonparametric. Non-parametric models usually do map from a database of existing images, often patches of images, and are used in texture synthesis (Efros et al., 1999), super-resolution (Freeman et al., 2002), and -image (Hays & Efros, 2007). Parametric models for image generation have been extensively investigated (eg MNIST figures or for tissue synthesis (Portilla & Simoncelli, 2000)). However, creating natural images of the real world has not had much success until recently. A variable sampling approach to rendering (Kingma & Welling, 2013) has had some success, but samples often suffer from blurriness. Another approach generates images using an iterative forward diffusion process (Sohl-Dickstein et al., 2015). Generative Adversarial Networks (Goodfellow et al., 2014) produced images that suffered from being noisy and incomprehensible. A laplacian pyramid extension of this approach (Denton et al., 2015) showed higher quality images, but they still suffered from making objects appear jittery due to the noise generated when chaining multiple models. An iterative network approach (Gregor et al., 2015) and a non-convolutional network approach (Dosovitskiy et al., 2014) have also had some success in producing natural images recently. However, they did not make use of generators for supervised missions.

C. VISUALIZING THE INTERNALS OF CNNS

One constant criticism of using neural networks has been that they are black-box methods, with little understanding of what the networks do in the form of a simple human-consumable algorithm. In the context of CNNs, Zeiler et. al. (Zeiler & Fergus, 2014) showed that by using deconvolutions and filtering the maximal activations, one can find the approximate purpose of each convolution filter in the network. Similarly, using a gradient descent on the inputs lets us inspect the ideal image that activates certain subsets of filters (Mordvintsev et al.).

IV. DESIGN OVERVIEW

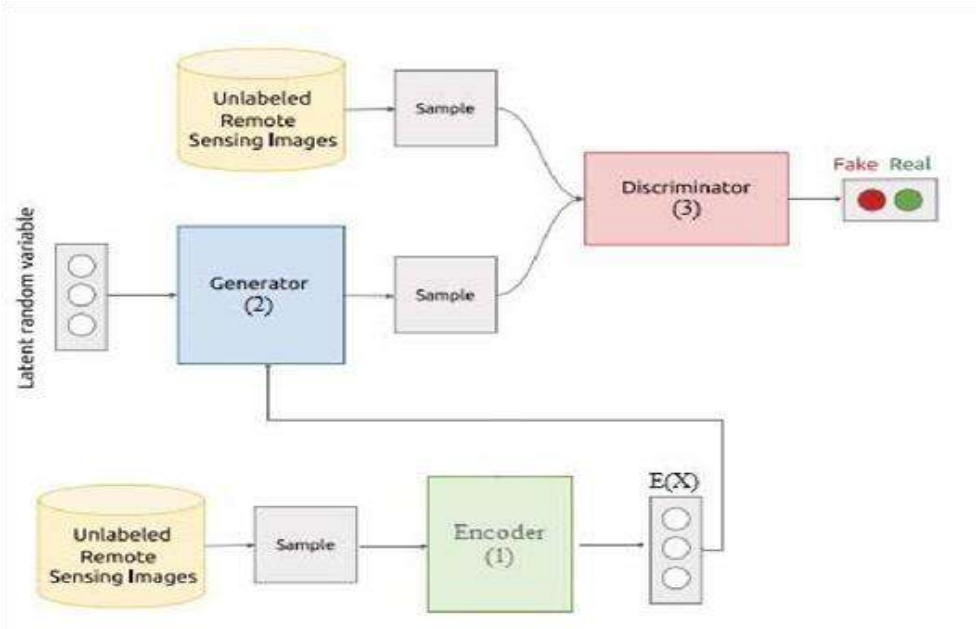

Figure1. design of the model

Generate random noise in a probability distribution such as normal distribution.
Make it as an input of our Generator neural network. It will output generated fake data . These steps can also mean that we sample some data from the distribution that the generator learned. We will notate the noise as (z_n) and the generated data as G(z_n). G(z_n) means the result of the noise processed by the generator G.
We combine the generated fake data with the data that is sampled from the dataset (which is real data). Make them to be input of our Discriminator. We will notate it as D. Discriminator will try to learn by predicting whether the data is fake or not. Train the neural network by doing forward pass and followed by back propagation. Updates the D weights

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

Then, we need to train the Generator. We need to make the G(z_n) or the fake data generated from random noises as the input of the D. Note that this steps only input the fake data into the Discriminator. Forward pass the G(z_n) in D. By using the Discriminator neural network, on doing forward pass, predict whether the fake data is fake or not (D(G(z_n))). Then do back propagation where we will only update the G weights.
Repeat these steps until we can see that the generator provide good fake data or maximum iteration has been reached.

V. SYSTEM ARCHITECTURE

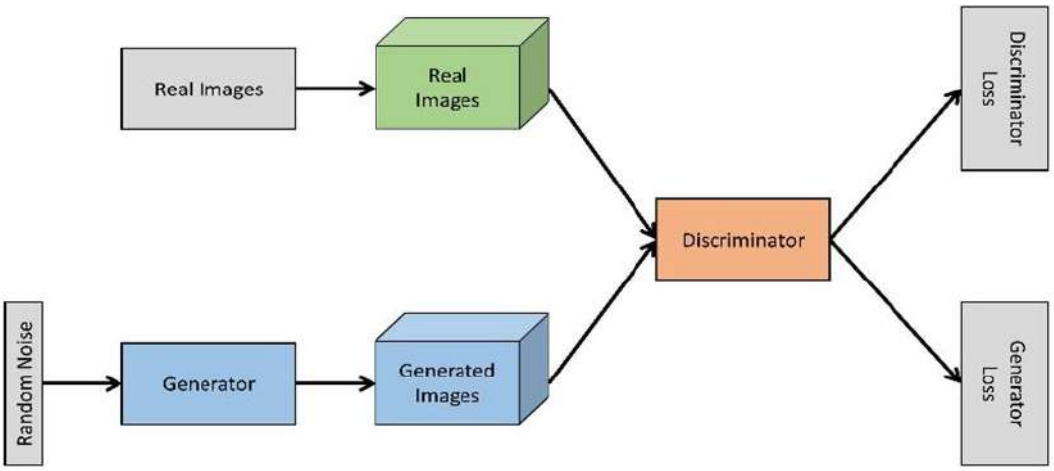

Figure2. Architecture of the model

VI. DATAFLOW DIAGRAM

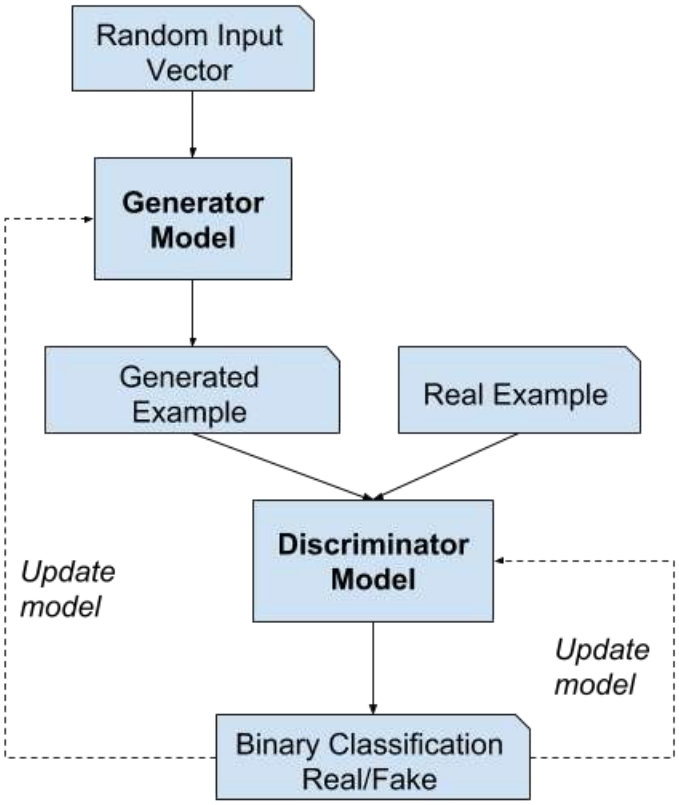

Figure3. DATAFLOW DIAGRAM

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

VII. IMPLEMENTATION

Generator and discriminator train in implementation.
Here are the steps involved in training the Generator.

- We generate a batch of images using the generator, pass the into the discriminator.
- We calculate the loss by setting the target labels to 1 i.e. real. We do this because the generator's objective is to "fool" the discriminator.
- We use the loss to perform gradient descent i.e. change the weights of the generator, so it gets better at generating real-like images to "fool" the discriminator.

Here are the steps involved in training the Discriminator.

- We expect the discriminator to output 1 if the image was picked from the real MNIST dataset, and 0 if it was generated using the generator network.
- We first pass a batch of real images, and compute the loss, setting the target labels to 1.
- Then we pass a batch of fake images (generated using the generator) pass them into the discriminator, and compute the loss, setting the target labels to 0.
- Finally, we add the two losses and use the overall loss to perform gradient descent to adjust the discriminator

VIII. RESULTS

Snapshot 1:
1st generated image [fake score > real score]

Figure4. generated image 1

Snapshot 2:
5th generated image [fake score > real score]

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

Figure5. generated image 5

Snapshot 3:
10th generated image [fake score = real score]

x	data
t	text
z	noise
G	generator network
D	discriminator network
T	dimension of the text
I	dimension of the image
$\mathcal{N}$	the dimension of the noise
φ	encoder
$\mathcal{L}$	loss
s	score

Algorithm 1 Training algorithm.

Input: images x, texts t, $\hat{t}$, training epochs E
for $n = 1$ **to** E **do**
  $h \leftarrow \varphi(t)$, $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z,h)$ {Generate fake image}
  $s_r \leftarrow D(x,h)$, $s_w \leftarrow D(x,\hat{h})$, $s_f \leftarrow D(\hat{x},h)$
  $\mathcal{L}_D \leftarrow \mathcal{L}_{s_r} + \mathcal{L}_{s_w} + \mathcal{L}_{s_f}$, $\mathcal{L}_G \leftarrow \mathcal{L}_{s_f}$ {Loss of D, G}
  $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
  $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
end for

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

Figure6. generated image 10

Snapshot 4:
20th generated image [fake score < real score]

Figure7. generated image 20

Snapshot 5:
25th generated image [fake score < real score]

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115209

Figure8. generated image 25

IX. CONCLUSION AND FUTURE ENHANCEMENT

- We propose a more stable set of architectures for training generative adversarial networks.
- We would create a website and implement an interface for our newly generated images.
- We give evidence that adversarial networks learn good representations of images for supervised learning and generative modelling.
- There are still some forms of model instability remaining
- We noticed as models are trained longer they sometimes collapse a subset of filters to a single oscillating mode.
- After the model is trained we plan to create an advertising website and use our model to its benefit.

REFERENCES

1. Mehran Mehralian, Babak Karasfi, RDCGAN: Unsupervised Representation Learning With Regularized Deep Convolutional Generative Adversarial Networks, 10.1109/AIAR.2018.8769811
2. Alec Radford & Luke Metz, Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv:1511.06434v2 [cs.LG] 7 Jan 2016
3. Yu, Fisher, Zhang, Yinda, Song, Shuran, Seff, Ari, and Xiao, Jianxiong. Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365, 2015
4. Srivastava, Rupesh Kumar, Masci, Jonathan, Gomez, Faustino, and Schmidhuber, Ju ̈rgen. Under- standing locally competitive networks. arXiv preprint arXiv:1410.1165, 2014.
5. Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806, 2014.
6. Denton, Emily, Chintala, Soumith, Szlam, Arthur, and Fergus, Rob. Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751, 2015.
7. Dosovitskiy, Alexey, Fischer, Philipp, Springenberg, Jost Tobias, Riedmiller, Martin, and Brox, Thomas. Discriminative unsupervised feature learning with exemplar convolutional neural net- works. In Pattern Analysis and Machine Intelligence, arXiv:1406.6909
8. Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, Thomas Brox ,Learning to Generate Chairs, Tables and Cars with Convolutional Networks ,arXiv:1411.5928
9. Theis, L., van den Oord, A., and Bethge, M. A note on the evaluation of generative models. arXiv:1511.01844, Nov 2015. URL http://arxiv.org/abs/1511.01844.
10. Zhao, Junbo, Mathieu, Michael, Goroshin, Ross, and Lecun, Yann. Stacked what-where auto- encoders. arXiv preprint arXiv:1506.02351, 2015.