Unsupervised Representational Learning with Deep Convolutional Generative Adversarial Networks

Hardik Panchal
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070054@iitb.ac.in

Prajwal Kalpande
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070028@iitb.ac.in

Kalpit Borkar
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070029@iitb.ac.in

***Abstract*—Convolutional networks (CNNs) have been widely used in computer vision applications for supervised learning. Here we are presenting a stable implementation of deep convolutional generative adversarial networks (DCGANs) paper which learns good image representations (unsupervised learning) and can also be used as feature extractors. We have shown that DCGAN Discriminator learns useful features which can be used for various tasks in supervised learning and its comparison with various successful models for the classification task.**

***Index Terms*—Convolutional Neural Networks (CNNs), deep learning, generative adversarial networks (GANs)**

I. Introduction

In this paper, we try to understand the working of Generative Adversarial Networks (GANs). We trained DCGAN model on different datasets and validated our results with those obtained in the original research paper. GAN models have been proposed to learn unsupervised data and classify them into several categories, especially unlabelled images. Therefore we can use GANs as feature extractors and use these features for supervised data classifications which is demonstrated later in this paper.

We introduce Deep Convolutional Generative Adversarial Networks (DCGANs) that are more stable than vanilla GAN models when it comes to generating natural images. Next we compare the performance of DCGAN models with regular GAN models for unsupervised image classification and show their effectiveness. Finally, we visualize the learned filters of the trained DCGAN models.

The rest of the paper is organized as follows. In section II we discuss the background of GANs, the different types of models used to generate natural images and how can we visualize the internal structure of Convolutional Neural Networks (CNNs). In section III we discuss our approach towards implementing DCGAN models from vanilla GAN models and the structure of DCGANs. In section IV we discuss all datasets used to train the DCGAN models. In section V, we describe how a DCGAN model trained on ImageNet-1k dataset can be used as feature extractor for classification tasks. The results of training and classification are mentioned in section VI. In section VII we discuss theoretical aspects of DCGANs. In section VIII we conclude this paper. Section IX provides key links for this project.

II. Background and Prior Work

A. Representation Learning from Unlabeled Data

Unlabeled data is classified mainly using unsupervised learning methods such as K-means clustering. Unlabeled images are used to extract features that are used for supervised learning. Unsupervised learning is used on unlabeled data for the classification of images using different clustering techniques and to learn meaningful representations of those clusters. Auto-encoders, ladder structures, and hierarchical clusters are some of the powerful feature extraction methods. These feature extraction methods improve the accuracy of linear classification models. Fig. 1 shows the process for feature extraction using a GAN model.

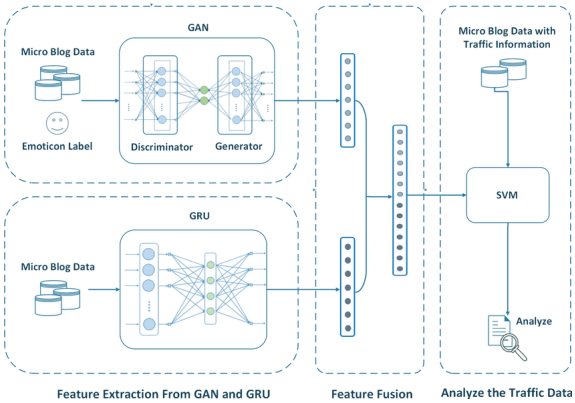


Fig. 1: Feature extraction using GANs
Reproduced from [1]

B. Generating Natural Images

Image-generating models are of two types:

1) Parametric models: Parametric models are used to generate images; however, generating natural images is a difficult task. Variational sampling approaches, iterative forward diffusion processes, and Laplacian pyramid extension approaches have improved the quality of natural images generated. Recurrent networks are quite successful in generating natural images; however, they do not use generators for these tasks. Natural images generated by GANs are noisy and incomprehensible.

2) Non-parametric models: Non-parametric models search on a large database of images and try to match patches of images among them. These models are used for enhancing the resolution of images, creating textures, and restoring deteriorated images or paintings.

C. Visualizing the Internals of CNNs

We can visualize the internals of CNNs by using deconvolutions and filtering the maximal activations. This gives us the purpose of each filter in the network. We can also run gradient descent on specific inputs and learn the image that activates certain filters in the network.

III. Approach

In the past years, scaling the GAN models by adding more CNN layers to the models has been unsuccessful. However, the following proposed architecture results in stable training, even for high-resolution images. This new architecture mainly consists of the following three changes in the structure.

Firstly, we replace all max-pooling layers in the network with convolutional layers. This allows the neural network to learn its own spatial upsampling and downsampling trends for generator and discriminator parts of the model respectively.

Next, we eliminate all fully connected dense layers which are on top of the convolutional features. We achieve this by implementing global average pooling layers, which increases the model stability but leads to slower convergence speed. A trade-off is achieved by connecting the highest convolutional features to the input of the generator and the output of the discriminator. Fig 2. below shows the modified model architecture, with the first layer being a fully connected layer and the last layer flattened being fed into a sigmoid function.

Finally, we use batch normalization techniques to normalize the data such that it has zero mean and unit variance. This results in stabilization of the training process of the model and deals with the issues that arise due to poor initialization of data. This prevents the GAN from collapsing to a single image which is a common point of failure during the process training the model. Batch normalization is applied to the output layer of the generator part of the model and the input layer of the discriminator.

Fig. 3 shows the architecture of the DCGANs used in this paper. The proposed architecture for a stable DCGAN model is summarized as follows:

1) Replace all max-pooling layers with discriminators and generators.
2) Apply batch normalization to the output layer of the generator and the input layer of the discriminator.
3) Remove all fully connected dense layers from the model.
4) Use LeakyReLU for discriminators and ReLU for generators; however, use Tanh for the output layer.

IV. Datasets

We train DCGAN models on the following three datasets: Tiny Imagenet, Large-scale Scene Understanding (LSUN), and Faces dataset.

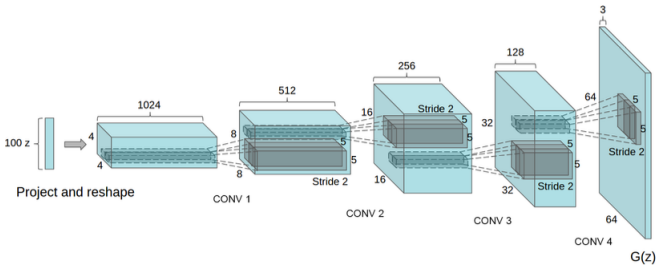


Fig. 2: Generator architecture of DCGAN model

A. Tiny Imagenet

Tiny Imagenet is a small sample of data from the complete dataset Imagenet [2]. Imagenet contains natural images of size 64x64 used for unsupervised learning. The following image in Fig. 4 shows some samples from the Tiny Imagenet dataset:

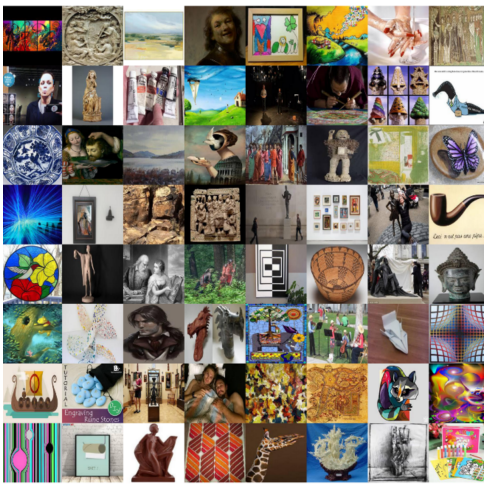


Fig. 3: Tiny Imagenet dataset sample

B. Large-scale Scene Understanding (LSUN)

The following image in Fig. 5 shows some samples from the Large-scale Scene Understanding (LSUN) dataset [3]:

C. Human Faces

The Faces dataset contains 3 million images of unlabeled human faces [4]. The following image in Fig. 6 shows some samples from the Human Faces dataset:

D. SVHN

The following image in Fig. 7 shows some samples from the SVHN dataset [5]:

Fig. 4: LSUN dataset sample

Fig. 5: Human Faces dataset sample

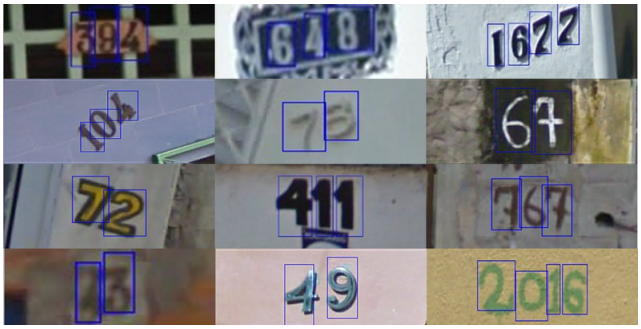


Fig. 6: SVHN dataset sample

V. Experiments

A. Classifying CIFAR-10 using GANs as feature extractor

To judge the performance of DCGAN models for unsupervised learning, we use DCGAN models as feature extractors and use these features to evaluate the performance of linear models for supervised tasks. In this paper, we mainly used linear SVM models for this process.

To evaluate DCGANs as feature extractors, we train the DCGAN models on the Tiny Imagenet dataset and then use the extracted features from the discriminator's convolutional layers. After passing these features through max-pooling layers, we flatten and concatenate these features and apply a linear SVM model for classification. For this experiment, the model was never trained on CIFAR-10, but the model was evaluated on it. This shows the robustness and effectiveness of the features extracted by the DCGAN models from the given dataset.

B. Classifying SVHN digits using GANs as feature extractor

We extract the features of the unlabeled dataset using DCGANs and use these features to classify the digits of the StreetView House Numbers (SVHN) dataset since the labeled data is limited. The process is similar to that of the CIFAR-10 experiment. Some training samples are randomly selected and trained on a linear SVM model with the same feature extraction pipeline that was used in the CIFAR-10 experiment.

VI. Results

A. Classifying CIFAR-10 using GANs as feature extractor

Fig. 14 shows the generated images with the model trained on Tiny Imagenet dataset.

B. Classification results for CIFAR-10

Fig.15 shows the classification report of the CIFAR-10 dataset using the DCGAN model. Here we can see that an accuracy score of 84 % is achieved on the validation dataset, which agrees with the results of the paper.

C. Classification results for SVHN

Fig.16 shows the classification report of the SVHN dataset using DCGAN model. Here we can see that an accuracy score of 57 % is achieved on the validation dataset, which is less than mentioned in the paper(72 %) because we had to settle with a smaller train dataset due to the limitation of computation power.

D. Images generated with the model trained on various datasets

Fig. 17 shows the generated images with the model trained on the Human Faces dataset. Fig. 18 shows the generated images with the model trained on the LSUN dataset.Fig. 8 shows real vs. fake images generated by the DCGAN model when trained on the Imagenet dataset.Fig. 9 shows real vs. fake images generated by the DCGAN model when trained on the Human Faces dataset. Fig. 10 shows real vs. fake images generated by the DCGAN model when trained on the LSUN dataset.

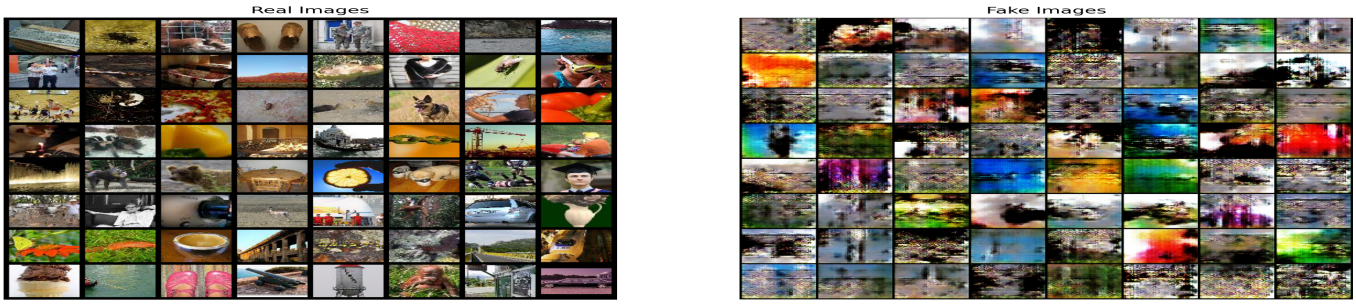


Fig. 7: Real vs. fake images generated by the DCGAN model when trained on the Tiny Imagenet dataset

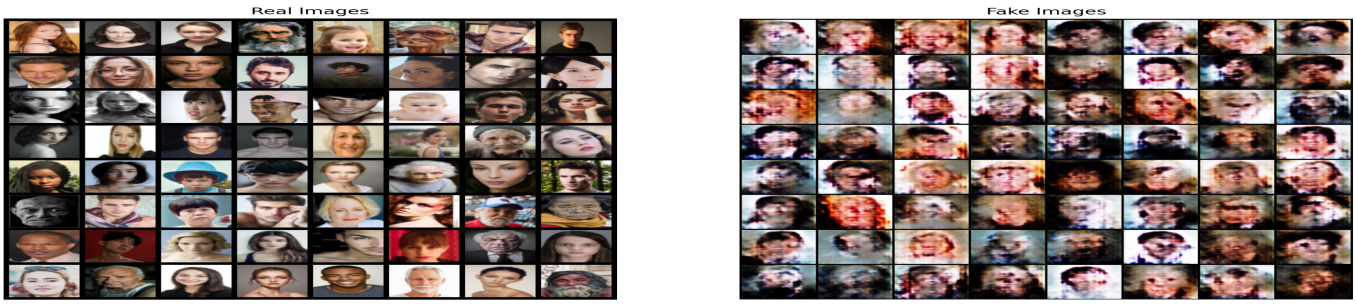


Fig. 8: Real vs. fake images generated by the DCGAN model when trained on the Human Faces dataset

Fig. 9: Real vs. fake images generated by the DCGAN model when trained on the LSUN dataset

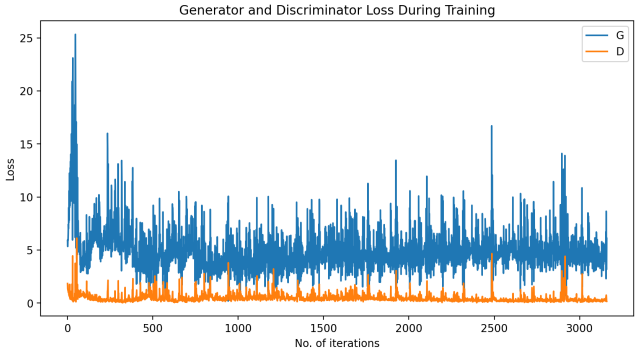


Fig. 10: Loss plot for Tiny Imagenet dataset

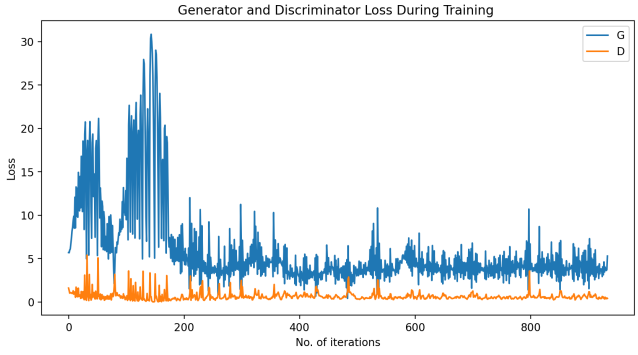


Fig. 11: Loss plot for Human Faces dataset

E. Loss plots for datasets

Fig. 11 shows the loss plot for training the DCGAN model on the Imagenet dataset. Fig. 12 shows the loss plot for training the DCGAN model on the Human Faces dataset. Fig. 13 shows the loss plot for training the DCGAN model on the LSUN dataset. The trend is common across all three datasets, which is that the loss profile saturates after a certain number of iterations and it doesn't improve any further.

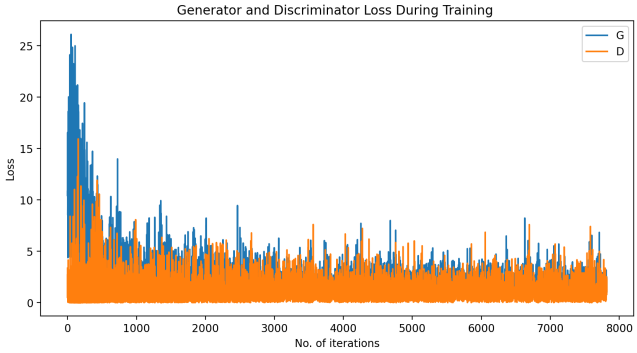


Fig. 12: Loss plot for LSUN dataset

	precision	recall	f1-score	support
airplane	0.80	0.84	0.82	95
automobile	0.92	0.94	0.93	98
bird	0.72	0.85	0.78	85
cat	0.71	0.80	0.75	89
deer	0.82	0.80	0.81	103
dog	0.84	0.79	0.81	107
frog	0.90	0.78	0.84	115
horse	0.88	0.92	0.90	96
ship	0.96	0.85	0.90	113
truck	0.90	0.91	0.90	99
accuracy			0.84	1000
macro avg	0.84	0.85	0.84	1000
weighted avg	0.85	0.84	0.85	1000

Fig. 13: Classification report CIFAR-10

	precision	recall	f1-score	support
0	0.81	0.61	0.69	152
1	0.80	0.65	0.72	485
2	0.63	0.58	0.61	358
3	0.32	0.40	0.36	189
4	0.59	0.58	0.58	190
5	0.49	0.57	0.52	175
6	0.47	0.55	0.51	122
7	0.56	0.62	0.59	151
8	0.39	0.53	0.45	90
9	0.35	0.45	0.40	88
accuracy			0.57	2000
macro avg	0.54	0.55	0.54	2000
weighted avg	0.60	0.57	0.58	2000

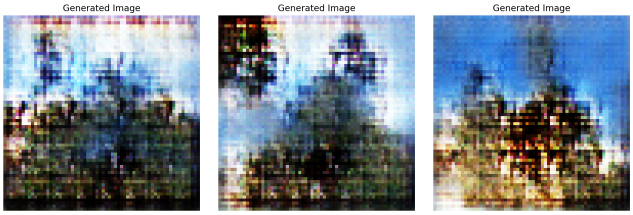
Fig. 14: Classification report for SVHN dataset

VII. Discussion

A. Vector arithmetic on face samples

Unsupervised representation of data obtained from DCGANs has a linear structure in representation space. This is observed by applying vector arithmetic on face samples of the Human Faces dataset. This representation space is known as the Z space, the vectors as Z vectors.

This can be seen from the figure, (a) and (b) show two faces generated by feeding two noise vectors as input to trained Generator. (c) shows image generated by feeding sum of earlier noise vectors to Generator. (d) is obtained by doing pixel averaging of (a) and (b). We can clearly see the linear structure from the similarity of (c) and (d).



(a) (b) (c)

Fig. 15: Generated images from model trained on LSUN dataset

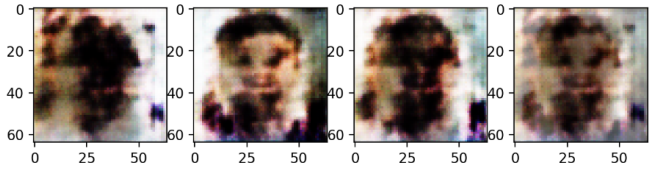


Fig. 16: Vector arithmetic on Generated Faces

VIII. Conclusion

We trained DCGAN on various datasets and observed their training loss. Later, we used these trained models as feature extractors and performed classification. We can see that DCGAN learns good features that can be used to supervised learning and are a more stable architecture for training GANs. DCGANs have made impressive strides in producing high-quality images and are anticipated to have a big impact across a range of industries, including the arts, entertainment, and medical. DCGANs have a lot of potential for developing computer vision and machine learning, despite the difficulties that still need to be solved.

IX. Key Links

Link to the GitHub repository containing all code files: link
Link to the demo video of our project: link

Acknowledgment

We would like to express our sincere gratitude to Professor Amit Sethi for their invaluable guidance and support throughout the duration of this project. Their extensive knowledge, constructive feedback, and unwavering encouragement have been instrumental in the successful completion of this project. Their insightful suggestions and recommendations have been critical in improving the quality of this project.

References

[1] Sentiment-based traffic condition analysis - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-framework-of-our-approach-1-Feature-extraction-the-CGAN-is-trained-on-a_fig2_326091769
[2] https://paperswithcode.com/dataset/tiny-imagenet
[3] https://www.kaggle.com/datasets/ajaykgp12/lsunchurch
[4] https://www.kaggle.com/datasets/ashwingupta3012/human-faces
[5] https://www.kaggle.com/datasets/stanfordu/street-view-house-numbers
[6] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ArXiv. /abs/1511.06434.