Generative Vision: Unleashing Realistic Image Synthesis with Conditional DCGANs

Weiyue Li *
Halıcıoğlu Data Science Institute
UC San Diego
wel019@ucsd.edu

Charles Ye *
Computer Science and Engineering
UC San Diego
juy022@ucsd.edu

Abstract

The ability to generate realistic synthetic images has garnered significant attention in recent years due to its vast potential applications in various domains. This report delves into the application of Conditional Deep Convolutional Generative Adversarial Networks (cDCGANs) and Conditional Generative Adversarial Networks (cGANs) in synthetic image generation, a field that has gained significant attention recently. By leveraging the capabilities of generative adversarial networks (GANs) and incorporating conditional inputs, the proposed approach enables the generation of images with specific desired attributes. The project involves training cDCGANs and cGANs on large-scale labeled datasets, where the models are conditioned on auxiliary information such as class labels. We also explore the influence of the concatenation of conditional information at different stages. These models are trained on the original labeled dataset and subsequently tested on a combination of real and synthetic images. By evaluating the performance of the classifiers on synthetic images, valuable insights regarding the realism and quality of the generated samples can be obtained. Additionally, the project explores the scalability of the proposed approach by evaluating its accuracy on more complex datasets. Through analyzing the models' performance on challenging and intricate datasets, the study provides insights into the potential of cDCGANs and cGANs to generate high-quality images with accurate labels, thereby alleviating the labor-intensive process of manual annotation. In conclusion, this project contributes to advancing synthetic image generation techniques using conditional GANs, while also exploring the potential for automating the labeling process based on similarity measures.

1 Introduction

The pursuit of generating synthetic images that are indistinguishable from real images has long been a goal in the field of computer vision and artificial intelligence. Recent advancements in generative adversarial networks (GANs) [3] have shown promise in addressing this challenge by training networks in an adversarial manner. This project focuses on the exploration of synthetic image generation using conditional Generative Adversarial Networks (cGANs) [7]. By incorporating condition inputs, such as class labels, into the image generation process, these networks offer the ability to generate images with specific desired attributes. Among the different variations of GANs, the condition deep convolutional GANs (cDCGANs) have shown promise in generating high-quality images by incorporating class labels as conditioning information. However, there is still a need to explore the potential of cDCGANs further and evaluate their performance across different datasets.

One aspect that has been explored is the concatenation of conditional information at different stages of the generator network. In early concatenation, the conditional information is concatenated with the

*Equal contribution

input noise vector or latent representation at the beginning of the generator network. This concatenated vector influences the generation process at every layer, allowing the conditional information to have a direct impact on the low-level features generated by the early layers. In late concatenation, In late concatenation, the conditional information is concatenated with the features extracted from the intermediate layers of the generator network. The generator first processes the initial input through multiple layers to extract intermediate features. These features are then combined with conditional information, guiding the subsequent layers and shaping the generation process. Late concatenation allows the conditional information to have a more localized influence on the features and high-level representations generated by the later layers.

In our project, we explore both early and late concatenation approaches within the context of cDCGANs for synthetic image generation. We aim to evaluate the performance of these techniques on various datasets, including CIFAR-10, CIFAR-100 [5], Intel Natural Scene [8], MNIST , and Fashion-MNIST [1]. By incorporating classification accuracy, Fréchet Inception Distance (FID) scores [4], Inception Scores (IS) [10], and similarity measures between synthetic and real images, we assess the quality, realism, and accuracy of the generated images.

By investigating early and late concatenation in cDCGANs, our project contributes to the advancement of synthetic image generation techniques. We aim to gain insights into the impact of conditional information at different stages of the generator network and explore their effectiveness in generating high-quality images with accurate labels. Additionally, we aim to assess the scalability of these techniques and investigate their potential for automating the labeling process based on similarity measures.

In addition to traditional evaluation methods, this project explores an alternative perspective by investigating the similarity between the generated images and their original counterparts as a measure of accuracy. By leveraging the resemblance between the synthetic and real images, there is potential to automate the labeling process, reducing the burden of manual annotation. This approach not only aims to enhance efficiency but also reduces the reliance on human intervention. Overall, this project contributes to the advancement of synthetic image generation techniques using conditional DCGANs. Through evaluation using image classifiers and analysis of scalability, the study shed light on the potential of cDCGANs and cGANs to generate high-quality images with accurate labels. Furthermore, the exploration of automated labeling based on similarity measures presents a promising avenue for streamlining the annotation process.

2 Related Work

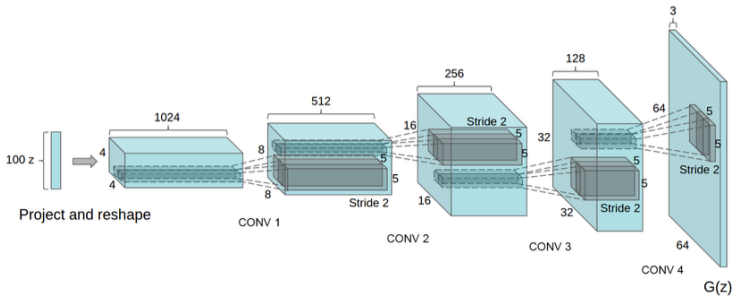


Figure 1: Architecture of DCGANs

Generative Adversarial Networks (GANs) is an unsupervised machine learning model that was originally introduced by [3]. It consists of two deep neural networks, namely the Generator and Discriminator, which collaborate to generate highly realistic images. The objective of the Generator is to improve the quality of synthetic images, while the Discriminator focuses on enhancing its ability to distinguish between real and fake images. Initially, the original GANs architecture did not incorporate convolutional layers, leading to issues with training stability. To address this problem, [9] proposed the DCGANs (Deep Convolutional GANs) architecture as a variation of GANs. DCGANs

replace deterministic spatial pooling layers with stride convolutional layers in both the Generator and Discriminator networks, resulting in enhanced training stability and the generation of higher-quality images. Figure 1 shows the architecture of DCGANs.

In the same year as the introduction of GANs, [7] proposed a variant known as Conditional Generative Adversarial Networks (cGANs). This innovative architecture incorporated real image labels in the forward passes, enabling users to control the specific class of images they desired to generate. This feature proved to be valuable in scenarios where acquiring image data was expensive or restricted. However, similar to the original GANs architecture, this model also did not employ convolutional layers. Figure 2 shows the architecture of cGANs.

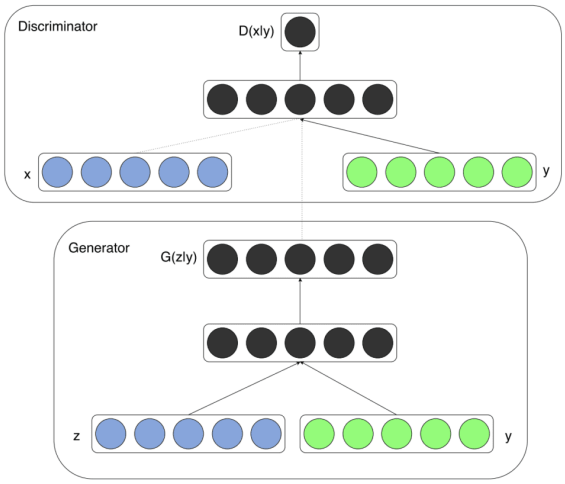


Figure 2: Architecture of Conditional GANs

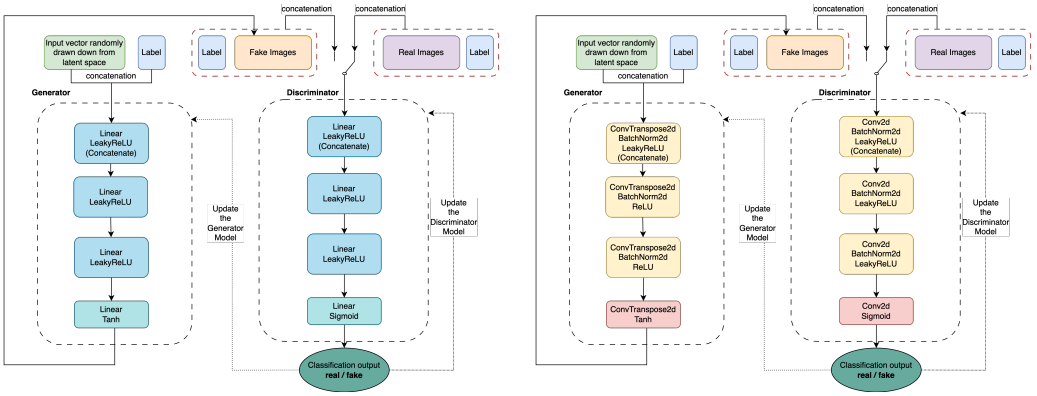
In regular GAN, the generator takes as input a random noise vector (typically sampled from a Gaussian distribution) and generates synthetic samples. The discriminator network is trained to distinguish between the real samples from the training dataset and the synthetic sample generated by the generator. On the other hand, a cGAN extends the GAN framework by conditioning the generator on the additional input information, typically in the form of a class label or other auxiliary data. This means that instead of just taking random noise as input, the generator also takes in a conditional vector that provides specific information about the desired output. The objective function of two-player minmax game would be as follow in Equation 1:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|\mathbf{y})))] \tag{1}$$

To evaluate the quality and fidelity of generated images, two commonly used metrics are the Inception Score (IS) and the Fréchet Inception Distance (FID) score. The Inception Score, proposed by [10], measures the quality and diversity of generated images using an Inception-v3 classifier. It considers both the clarity of individual images and the diversity of classes they cover. On the other hand, the FID score, introduced by [4], measures the similarity between the distribution of real and generated images using feature representations extracted from an Inception-v3 network. The lower the FID score, the closer the generated images are to the real data distribution.

ResNet18 is a popular deep convolutional neural network architecture that was introduced by [2]. It is specifically designed for image classification tasks and has become a widely adopted model in computer vision research and applications. ResNet18 is characterized by its deep structure, consisting of 18 layers, and its unique use of residual connections. These connections allow the network to learn residual mappings, enabling the model to efficiently train and handle the challenges of training very deep networks.

In this project, we aim to enhance the performance of traditional Conditional Generative Adversarial Networks (cGANs) by incorporating the architectural improvements introduced by Deep Convolutional GANs (DCGANs). Specifically, we implement Conditional DCGANs (cDCGANs) that leverage the power of convolutional layers to improve the quality of generated images. To evaluate



(a) Architecture of Conditional Generative Adversarial Networks (cGANs)

(b) Architecture of Conditional Deep Convolutional Generative Adversarial Networks (cDCGANs)

Figure 3: Architectures of cGANs and cDCGANs

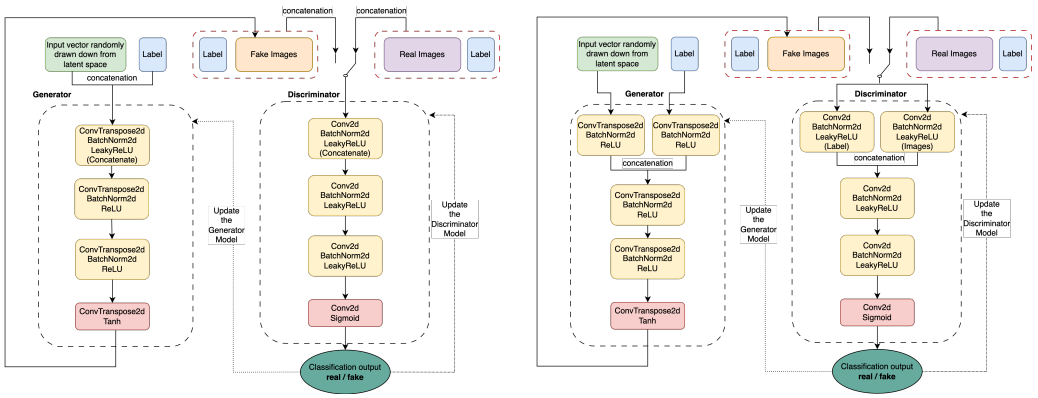
the effectiveness of our approach, we employ multiple assessment methods. Firstly, we utilize the Inception Score (IS) to measure the quality and diversity of the generated images. Additionally, we employ the Fréchet Inception Distance (FID) to quantify the similarity between the distributions of real and generated images. Lastly, we perform image classification using a pre-trained ResNet18 model to assess the perceptual quality and realism of the generated images. By employing these evaluation metrics, we can obtain a comprehensive understanding of the performance of our cDCGANs model and validate its effectiveness in generating high-quality and realistic images.

3 Method

The main objective of our project is to enhance the performance of cGANs by modifying both the generator and discriminator network architectures. Figure 3 (a) illustrates the architecture of the original cGANs model. The original cGANs network did not use convolutional layers. Therefore, we adopt the idea of DCGANs architecture (Figure 1) and use convolutional layers in the cGANs. We call the new proposed architecture cDCGANs, which is illustrated in Figure 3 (b). We hypothesize that using more powerful convolutional layers instead of the current fully connected layers will enable cDCGANs to generate higher-quality images with greater classification accuracy when subjected to the same classifier. In addition to the cDCGANs model, we have slightly changed the architecture of how the noises and labels are concatenated in the Generator and how the images and labels are concatenated in the Discriminator. This new architecture, cDCGANs2, is presented in Figure 4.

The training algorithm for the cDCGANs follows the traditional GANs training structure. The only difference is that we add labels to both the Generator and Discriminator during the training process as specified in Algorithm 1.

For the image classification tasks on each architecture, we trained three classifiers: one on all real images, one on all fake images generated by our models, and one on a combination of fake and real images. By comparing the accuracy of these classifiers, we gain valuable insights into the quality of the images generated by our models and their ability to resemble real images. Our proposed technique, cDCGANs, differs from previous work in the use of convolutional layers instead of fully connected layers in the generator and discriminator networks. This modification allows cDCGANs to capture spatial information and generate more realistic images. The use of convolutional layers has been shown to be effective in improving the performance of image generation models. By incorporating convolutional layers into the cGANs architecture, we aim to leverage their benefits and enhance the quality of generated images. Additionally, the introduction of labels to both the generator and discriminator enables better alignment between the generated samples and the desired target, leading to improved classification accuracy. The strength of our method lies in its ability to generate high-quality images and improve classification performance, making it a promising approach for various image-generation tasks.



(a) cDCGANs (b) cDCGANs2

Figure 4: Architecture comparison between cDCGANs (early concatenation) and cDCGANs2 (late concatenation)

Algorithm 1 cDCGANs Training Algorithmn

procedure cDCGANsTraining
    Initialize parameters
    Initialize variables
    for Epoch of training **do**
        Switch to training mode for both Generator and Discriminator
        for Batch (images, labels) in dataloader **do**
            Generate new noises.
            Generate fake images using the Generator with noises and real labels
            Generate fake output using the Discriminator with fake images and real labels.
            Generate real output using the Discriminator with real images and real labels
            Calculate the D_loss.
            Update the Discriminator
            Generate fake output using the Discriminator with fake images and real labels.
            Calculate the G_loss
            Update the Generator
        end for
    end for
end procedure

In terms of concatenation in different stages, early concatenation and late concatenation refer to two different approaches for incorporating conditional information into the generator network. Here are the differences between these two concatenation methods:

3.1 Early Concatenation

In early concatenation, the conditional information is concatenated with the input noise vector or latent representation at an early stage of the generator network. The concatenated vector serves as the initial input to the generator, and it propagates through the layers of the network. The conditional information is combined with the noise vector or latent representation from the beginning, influencing the generation process at every layer of the generator network. Thus, this approach could allow the conditional information to have a direct impact on the low-level features generated by the early layers of the generator.

3.2 Late Concatenation

In late concatenation, the conditional information is concatenated with the features extracted from the intermediate layers of the generator network. The conditional information is concatenated

with the features at a later stage of the generator, typically after several layers of processing. The generator network first processes the initial input (e.g., noise vector or latent representation) through multiple layers to extract intermediate features. These intermediate features are then concatenated with conditional information to influence the subsequent layers and guide the generation process. Therefore, late concatenation could allow the conditional information to have a more localized influence on the features and high-level representations generated by the later layers of the generator.

4 Experiments

The aim of this experiment is to explore the effectiveness of conditional Deep Convolutional Generative Adversarial Networks (cDCGANs) and conditional Generative Adversarial Networks (cGANs) in generating synthetic images and labels to train downstream image classification models, reducing the need for manual labeling and alleviating the labeling bottleneck commonly encountered in large-scale image datasets by leveraging the generated synthetic data. We will access the quality of the generated images using the Fréchet Inception Distance (FID) and Inception Score metrics, and evaluate the suitability of synthetic images for training image classification models such as ResNet, Softmax Regression, and other image classification models by measuring the accuracy of the resulting models.

4.1 Datasets

We have deployed our models on various datasets such as CIFAR-10 and CIFAR-100 [5], and Intel Natural Scene [8].

1. **CIFAR-10**: It consists of 60,000 color images, each with a resolution of 32×32 pixels, divided into 10 mutually exclusive classes. The dataset is split into 50,000 training images and 10,000 test images. Each class contains an equal number of images. The ten classes in the CIFAR-10 dataset include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks.
2. **CIFAR-100**: The CIFAR-100 dataset is an extension of the CIFAR-10 dataset. It also consists of 60,000 color images, but it encompasses a larger and more diverse set of classes. The images are 32×32 pixels in resolution, and they are divided into 100 fine-grained classes. Each superclass contains five fine-grained classes. The dataset is split into 50,000 training images and 10,000 test images.
3. **Intel Natural Scene**: It consists of around 25,000 images of size 150×150 distributed under 6 categories (buildings, forest, glacier, mountain, sea, and street). There are approximately 14,000 images in the training set, 3,000 in the test set, and 7,000 in the validation set.

The above datasets consist of RGB images with three channels, which were the primary focus of our study. However, to ensure the generality of our model on single-channel images, we also tested it on the MNIST by [6] and Fashion-MNIST by [11] datasets.

1. **MNIST**: The MNIST dataset is a widely used benchmark dataset in image classification tasks. It consists of a collection of 70,000 grayscale images of handwritten digits from 0 to 9. Each image has a resolution of 28×28 pixels. The dataset is divided into 60,000 training images and 10,000 test images.
2. **Fashion-MNIST**: The Fashion-MNIST dataset is an alternative, but more challenging, dataset compared to MNIST. It is aimed at benchmarking machine learning algorithms for image classification tasks. It comprises 60,000 grayscale images of fashion items, such as clothing and accessories, divided into 10 classes. The images also have a resolution of 28×28 pixels. Similar to MNIST, Fashion-MNIST provides a training set of 60,000 images and a test set of 10,000 images.

Table 1 provides detailed statistics on the datasets used in this project.

4.2 Results

Due to limited computational resources, we only used the suggested learning rate of 0.0002 and momentum term β_1 of 0.5, according to [9] in the original DCGANs paper to train all of our generative

Table 1: Dataset Statistics

Dataset	Number of Images	Number of Channels	Number of Classes
MNIST	70,000	1	10
Fashion-MNIST	60,000	1	10
CIFAR-10	60,000	3	10
Intel Natural Scene	25,000	3	6
CIFAR-100	60,000	3	100

models. Table 2 shows the results of the testing accuracy, FID score, and IS score on all five datasets. Note that for 3-channel images (CIFAR-10, CIFAR-100, and Intel Natural Scene), we trained the generative models for 100 epochs and used ResNet-18 as the classifier. For 1-channel images (MNIST and Fashion-MNIST), we trained the generative models for 10 epochs and used Softmax Regression as the classifier since these datasets are relatively simple. Moreover, we did not calculate the FID and IS scores for 1-channel images because grayscale images lack color information, which is a crucial aspect of assessing the realism and diversity of generated images. Additionally, the pre-trained models used by FID and IS scores were trained on large-scale datasets that contain a wide variety of complex and diverse features, which is not representative of grayscale MNIST and Fashion-MNIST.

Table 2: Results of cGANs and cDCGANs on Different Datasets

Dataset	Model	Acc (Fake)	Acc (Mixed)	Acc (Real)	FID	IS
CIFAR-10	cGANs	0.8877	0.8198	0.7940	0.1416	1.5933
CIFAR-10	cDCGANs	0.8081	0.7522	-	**0.0446**	**3.8459**
CIFAR-10	cDCGANs2	0.9997	0.9922	-	0.0960	2.7535
CIFAR-100	cGANs	0.9859	0.9129	0.7487	0.1194	1.6095
CIFAR-100	cDCGANs	0.9014	0.7518	-	**0.0655**	**3.3468**
CIFAR-100	cDCGANs2	0.9900	0.9793	-	0.1036	2.7136
Intel	cGANs	0.9133	0.8380	0.7090	0.2203	1.4002
Intel	cDCGANs	0.8123	0.7907	-	**0.0268**	**2.6772**
Intel	cDCGANs2	0.9720	0.8960	-	0.0340	2.5714
MNIST	cGANs	0.9952	0.9961	0.8218	N/A	N/A
MNIST	cDCGANs	0.8941	0.9883	-	N/A	N/A
MNIST	cDCGANs2	1.0	1.0	-	N/A	N/A
Fashion-MNIST	cGANs	0.8942	0.9957	0.7821	N/A	N/A
Fashion-MNIST	cDCGANs	0.7717	0.9775	-	N/A	N/A
Fashion-MNIST	cDCGANs2	1.0	0.9975	-	N/A	N/A

We observe that the classification accuracy on all synthesis images or mixed images of our cDCGANs2 model outperforms cGANs and is followed by cDCGANs (see Figure 9). However, both the FID score and the IS score for cDCGANs outperform those for cDCGANs2 and cGANs. Although the classification accuracy of cDCGANs may not be comparable to the other two models, its low FID score and high IS score indicate that the generated images closely match the distribution of real images in terms of visual appearance, structure, and overall feature distribution.

We also observe that, for the same model and dataset, the classification accuracy of all synthetic images usually outperforms that of mixed images, followed by all real images. This observation suggests that our generative models were able to extract important features from the real images, resulting in synthetic images with less noise and a higher information content. As a result, the classification models benefit from the enhanced quality of the synthetic data, leading to improved accuracy.

There are potential reasons why our cDCGANs2 model exhibits remarkably high classification accuracy. One possible factor could be the way it concatenates noises and labels in the generator and images and labels in the discriminator. Unlike cDCGANs, where concatenation occurs directly in the forward pass, the cDCGANs2 forward pass involves extracting information from noises/images using

a convolution block and extracting information from encoded labels using another convolution block. These extracted features are then concatenated with the remaining convolutional layers. Consequently, the initial step in the cDCGANs2 model may have compromised its generalizability to diverse images compared to the cDCGANs model.

Figure 5 and Figure 6 illustrate the differences among the sample images generated by the three models in our study. It is evident that the image quality of traditional cGANs is significantly vaguer compared to cDCGANs and cDCGANs2. However, the images generated by cDCGANs2 exhibit less diversity than those generated by cDCGANs. This disparity is particularly pronounced in the CIFAR sample, where the overall image complexity is lower compared to Intel Natural Scene.

In the case of MNIST and Fashion-MNIST, cDCGANs have generated a resemblance to the original datasets. These results in Figure 7 highlight the potential of cDCGANs in synthesizing high-quality images across diverse domains, showcasing their versatility and effectiveness in the field of generating image modeling. Considering the aforementioned factors, despite cDCGANs having lower classification accuracy compared to cDCGANs2, its superior FID score and IS score suggest that it produces highly realistic images that closely resemble real images in terms of visual appearance, structure, and overall feature distribution. Therefore, cDCGANs can be considered the best model among the three in our study.

However, we noticed that the FID and IS scores for cDCGANs and cDCGANs2 are the closest among all three 3-channel datasets, even though the classification accuracy for cDCGANs2 outperformed the others. A potential reason for this could be the larger image size of Intel Natural Scene compared to CIFAR (150x150 vs 32x32). This larger size allows the late concatenation model (cDCGANs2) to generate diverse samples while still achieving high classification accuracy on the synthetic images. Thus, for future studies, it would be worthwhile to use the proposed cDCGANs2 architecture to test on more datasets that are more complex than CIFAR, in order to further explore its capabilities and performance.

References

[1] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] G. Ian, P.-A. Jean, M. Mehdi, X. Bing, W.-F. David, O. Sherjil, ..., and B. Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[5] A. Krizhevsky, V. Nair, and G. Hinton. CIFAR dataset. `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

[6] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[7] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[8] K. Puneet. Intel image classification dataset. `https://www.kaggle.com/datasets/puneet6060/intel-image-classification`, 2020.

[9] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[11] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

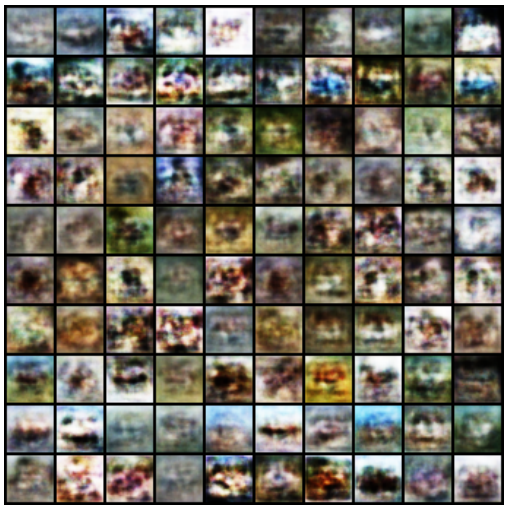
5 Supplementary Material

5.1 Resources

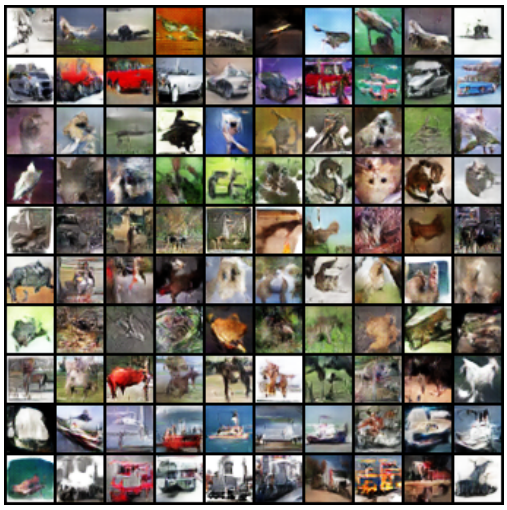
A video presentation of this project could be found here. The source code to run the experiments in this project could be found here.

5.2 Sample Images Generated

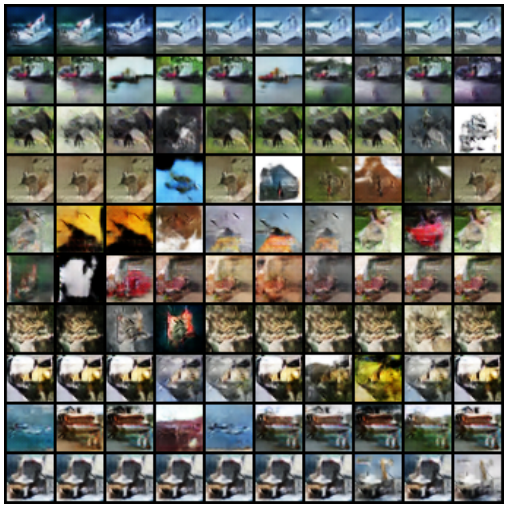
Figure 5 illustrates the samples generated by three models (cGANs, cDCGANs, and cDCGANs2) after 100 epochs of training on the CIFAR-100 dataset. It is evident that the quality of samples generated by cGANs is not competitive compared to those of cDCGANs and cDCGANs2. While the pixels in the samples generated by cDCGANs2 may appear clearer, they exhibit lower variability and diversity compared to the samples generated by cDCGANs. Additionally, Figure 6 displays the samples generated by the three models after 100 epochs of training on the Intel Natural Scene dataset. Due to the larger size (150x150) and the presence of more diverse features in the Intel Natural Scene dataset, cDCGANs2 was able to generate samples with greater diversity.



(a) cGANs

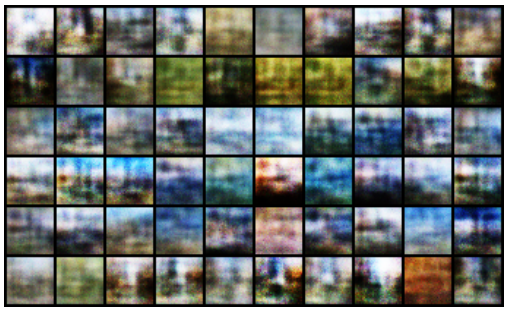


(b) cDCGANs

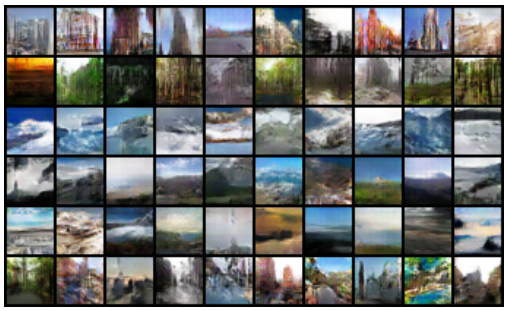


(c) cDCGANs2

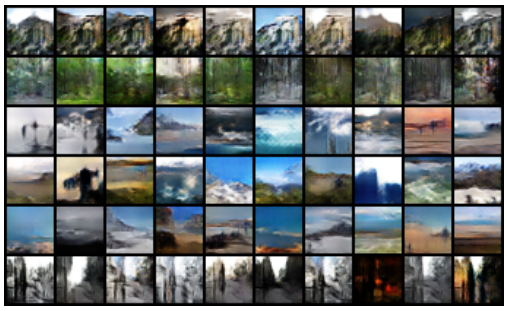
Figure 5: Samples for 3 models on CIFAR-10



(a) cGANs



(b) cDCGANs



(c) cDCGANs2

Figure 6: Samples for 3 models on Intel Natural Scene

Figure 7 showcases the ability of cDCGANs to generate clear and diverse samples within a limited 10-epoch training period on grayscale datasets such as MNIST and Fashion-MNIST. On the other hand, for cDCGANs2 (late concatenation), due to the simplicity of these grayscale image datasets, it produced highly similar results for each class. This observation explains the high classification accuracy achieved by cDCGANs2 on the grayscale synthetic images.

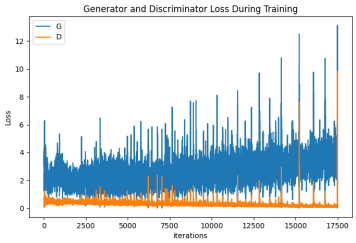
(a) MNIST

(b) Fashion-MNIST

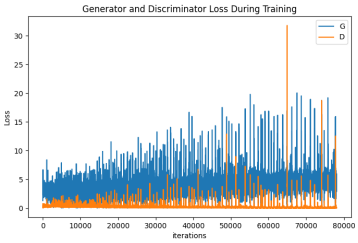
Figure 7: Samples of cDCGANs on grayscale images

5.3 Training Loss for Generator and Discriminator

Figure 8 shows the training loss for both Generator and Discriminator for our cDCGANs.



(a) Intel Natural Scene

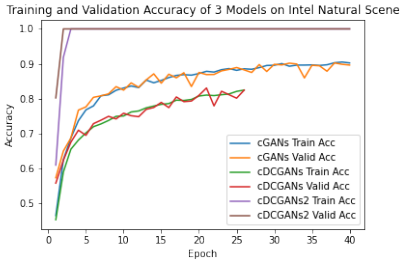


(b) CIFAR-10

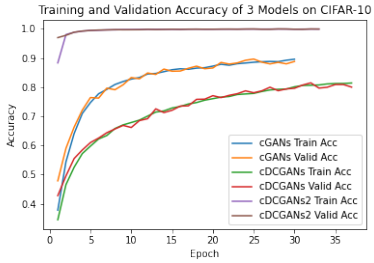
Figure 8: Traning loss for generator and discriminator

5.4 Training and Validation Accuracy

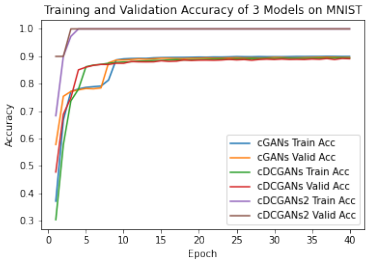
Figure 9 displays the training and validation accuracy for the three models on each dataset.



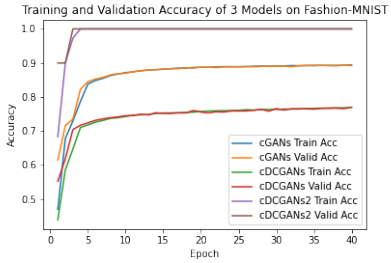
(a) Intel Natural Scene



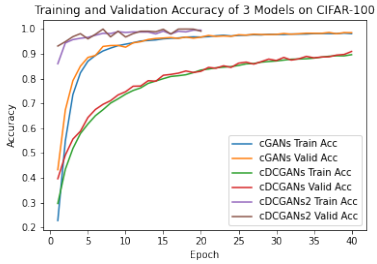
(b) CIFAR-10



(c) MNIST



(d) Fashion-MNIST



(e) CIFAR-100

Figure 9: Training and validation accuracy for 3 models on 5 datasets