HOMEWORK 4 (CPSC 8430 – DEEP LEARNING)
SHRAVANI KONDA

Generative Adversarial Networks

The aim of this assignment is to train two models, a generator and a discriminator by using advanced methods known as DCGAN, Wasserstein GAN, and ACGAN. These methods are used to make training more stable and ensure the generated images both look real and fit the correct categories.

GitHub Link:

https://github.com/shravanik31/Deep-Learning/tree/main/HW4

Dataset:

The dataset used is the CIFAR-10 dataset consisting of 60,000 small color images, each with 32x32 pixels, spread across 10 different categories like cars, birds, and dogs, etc.,

DCGAN (Deep Convolutional Generative Adversarial Network):

DCGAN is a type of GAN that uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively, enhancing the stability of training GANs on image data. DCGAN has been proven effective for generating high-quality synthetic images and is widely used for tasks that require learning feature representations from image data.

Architecture:

Generator:

- The Generator begins by embedding textual input into a dense vector, which is then concatenated with a random noise vector to enrich the input's variability.
- This combined vector is projected and reshaped into a high-dimensional space, forming an initial feature map.
- This map is progressively upsampled through multiple layers of transposed convolutions, each reducing the feature depth while increasing spatial dimensions, facilitated by batch normalization and ReLU activation for smooth training.
- The final layer transforms the upsampled features into a normalized image with the appropriate number of color channels using a Tanh activation function, effectively generating synthetic images based on the text input.

Discriminator:

- The Discriminator's role is to assess whether an image, combined with textual information, is real or generated by the Generator.

- It embeds the textual input into a feature vector similarly to the Generator and processes the image through a series of convolutional layers, incorporating Leaky ReLU and batch normalization for effective learning.
- The extracted image features are then flattened and concatenated with the text features, forming a combined feature set that undergoes a final convolution to predict the image's authenticity.
- This will allow the Discriminator to effectively evaluate the authenticity of images in the context of the provided textual descriptions.

Hyperparameters:

- Epochs: 50
- Learning rate: 0.0002
- Optimizer: Adam
- Batch size: 128
- Noise vector dimension: 100
- Text embedding dimension: 256
- Loss function: Binary Cross-Entropy

Wasserstein GAN (WGAN):

WGAN's loss provides meaningful gradients (better for training) and correlates more closely with image quality than traditional GAN loss functions. It uses a critic instead of a discriminator, which judges the 'realness' of images on a scale, not just real or fake, and is trained to convergence within each generator update step.

Architecture:

Generator:

- It starts with text feature embedding which converts textual input into a vector using a linear layer and a ReLU activation.
- A noise vector is expanded and concatenated with the text feature vector to serve as input for image generation.
- It then maps the combined noise-text vector to a higher dimension and reshapes it to form a starting image-like tensor.
- Then it increases the spatial size of the tensor while reducing the feature depth through several transposed convolutional layers with batch normalization and ReLU activations.
- It ends with a transposed convolutional layer that sets the tensor to the desired image size and color channels, followed by a Tanh activation to normalize pixel values.

Discriminator (Critic):

- It processes text input into a vector using a linear layer followed by a ReLU activation.

- Then it applies a series of convolutions that downsample the image, each followed by batch normalization and LeakyReLU activation for feature extraction.
- It flattens the convolutional features and combines them with the text embedding vector to form a holistic feature representation.
- Then a linear layer reduces the combined features to a single value representing the critic's assessment of the image's realness.

Gradient Penalty:
Gradient penalty in WGAN helps keep the training of the critic in check, ensuring it evaluates images smoothly. It penalizes the critic if the gradients with respect to mixed real and fake images get too large or too small, leading to more stable and consistent learning of how to judge images.

Hyperparameters:
- Epochs: 50
- Learning rate: 0.0002
- Optimizer: Adam
- Batch size: 128
- Noise vector dimension: 100
- Gradient penalty: 10
- Loss function: BCELoss (Binary Cross-Entropy Loss)
- Number of discriminator iterations per generator: 5

Auxiliary Classifier Generative Adversarial Network (ACGAN):
It adds a classification layer to the discriminator in a traditional GAN setup, enhancing its ability to judge not just the authenticity of images but also their class membership. This model trains the generator to produce not only convincing but also correctly classified images, improving the specificity and diversity of the output.

Architecture:
Generator:
- It starts with embedding Layer that converts class labels into dense vectors that are combined with noise to guide image generation.
- Then it sequentially increases the resolution of the generated image using transposed convolutions, normalized and activated by batch normalization and ReLU, respectively.
- The architecture ends with a transposed convolution that formats the output image to the correct color scale with a Tanh normalization.

Discriminator:

- The convolutional layers process the input image through multiple layers to downsample and extract features, using batch normalization and LeakyReLU for efficiency.
- It evaluates dual outputs: Real or Fake and Class Prediction.
- In Real or Fake Determination, it outputs a single value via a sigmoid function to classify images as real or fake.
- In class prediction, it uses log-softmax to classify the image into one of the given categories, enhancing the discriminator's ability to guide the generator towards specific classes.

Hyperparameters:

- Epochs: 50
- Learning rate: 0.0002
- Optimizer: Adam
- Batch size: 128
- Noise vector dimension: 100
- No. of classes: 10
- Feature maps for generator: 64
- Feature maps for discriminator: 64
- Loss function: BCELoss (Binary Cross-Entropy Loss) and NLLLoss (Negative Log-Likelihood Loss

Fréchet Inception Distance Calculation: FID is used to assess the quality of images generated by a model compared to real images. FID measures similarity by analyzing the distribution of features extracted by the InceptionV3 network from both real and generated images. Lower FID scores indicate better quality, showing that the generated images more closely resemble the real ones, thus reflecting improved model performance in generating realistic images.

Metrics:

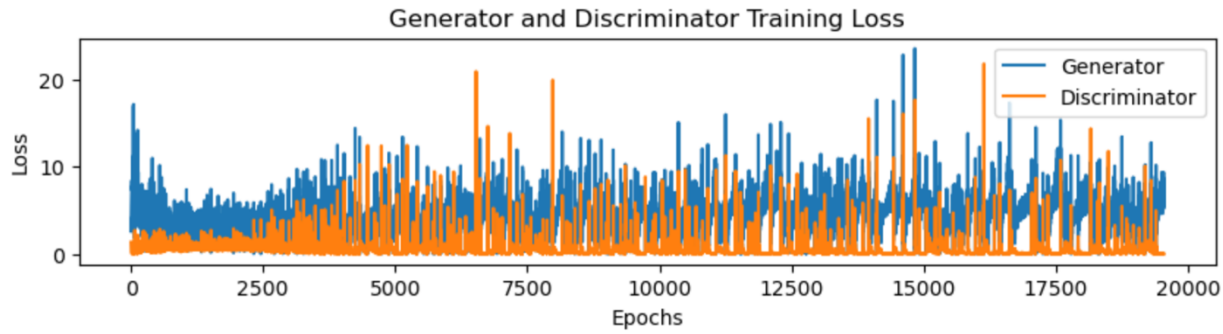
Loss Graph: Generator and Discriminator losses are calculated during training process. The generator's loss measures how effectively it fools the discriminator, while the discriminator's loss assesses its accuracy in distinguishing real from fake images.

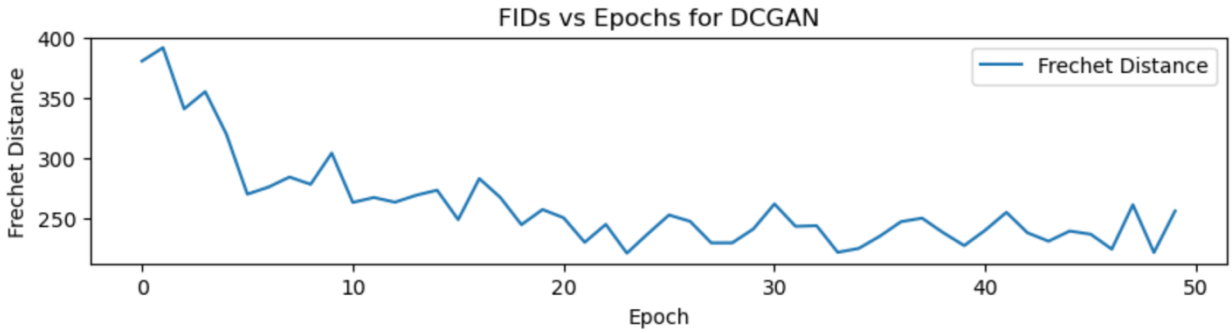
Fréchet Inception Distance (FID) graph: FID compares the similarity between the distributions of real and generated images, with lower values indicating higher quality and realism.

Results:

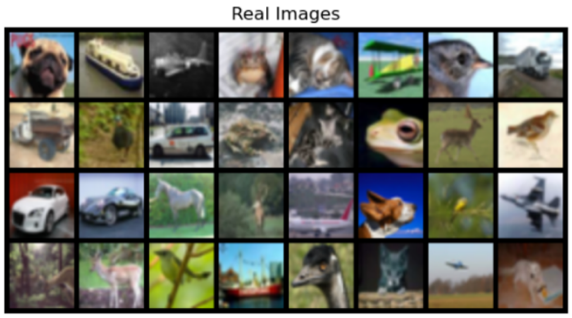
DCGAN:

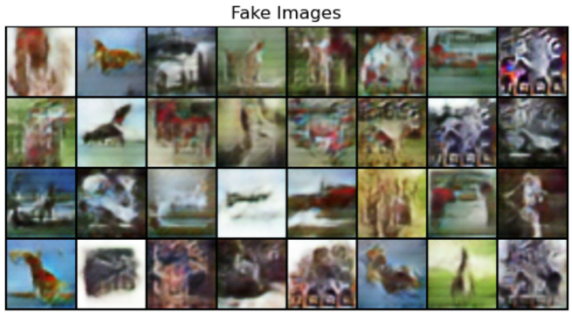
The loss and FID graph for DCGAN is below,





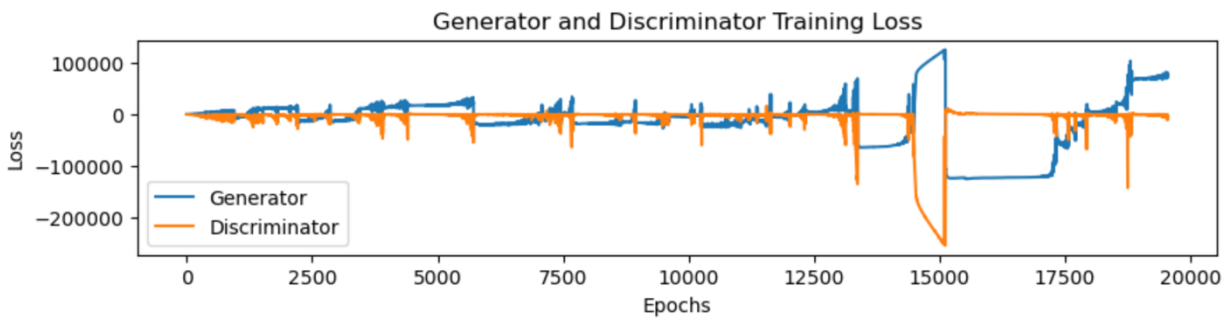
The images generated are below,

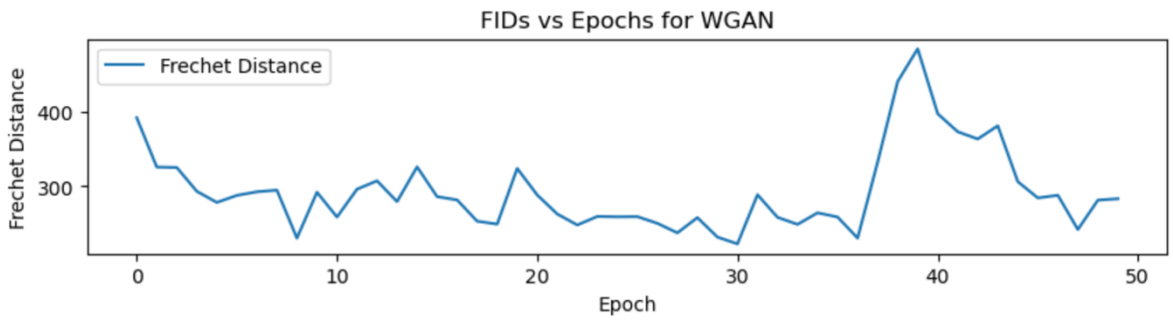




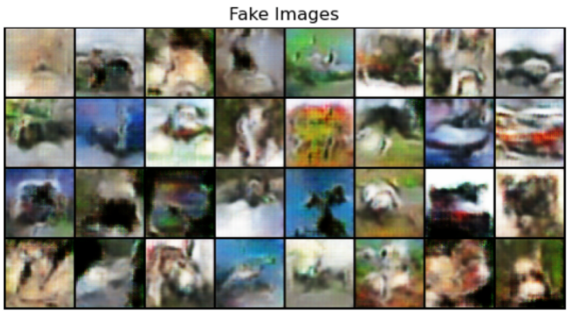
WGAN:

The loss and FID graph for WGAN is below,



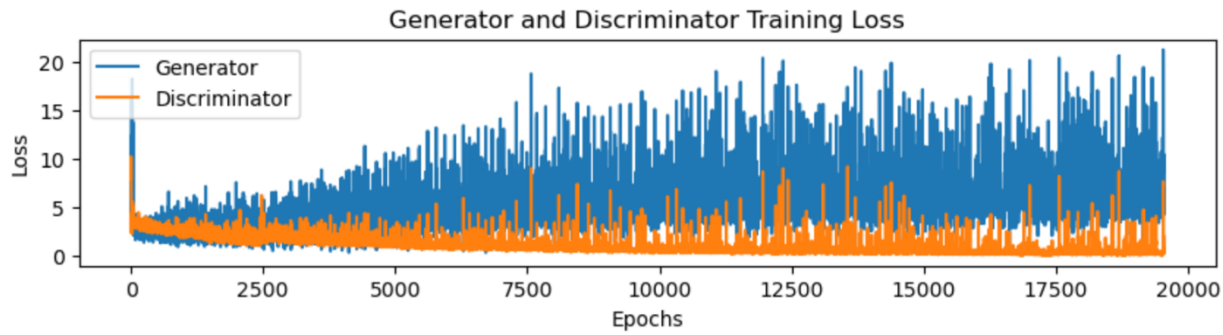


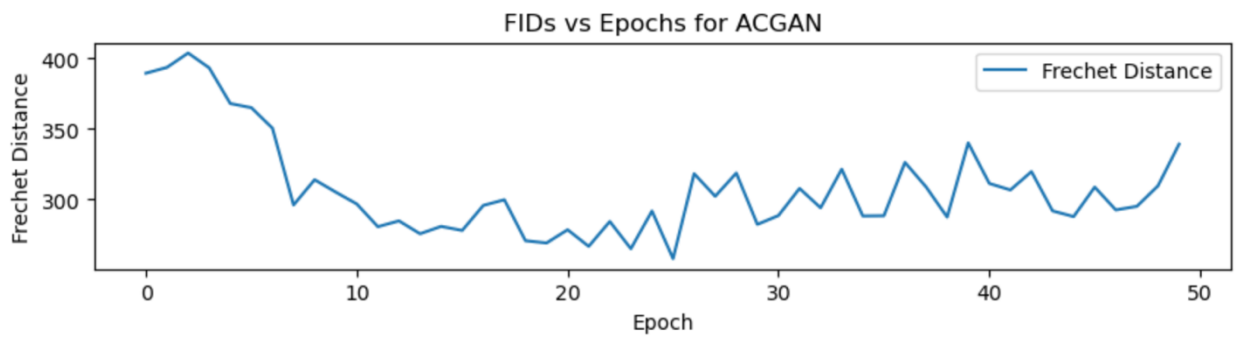
The images generated are below,



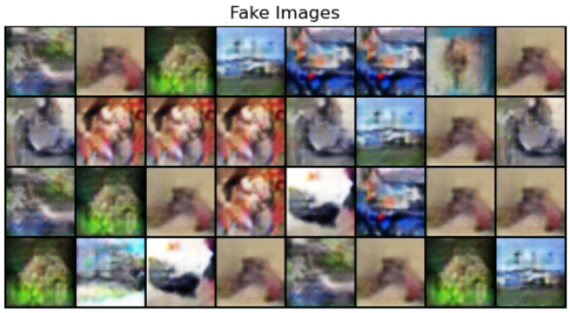
ACGAN:

The loss and FID graph for ACGAN is below,

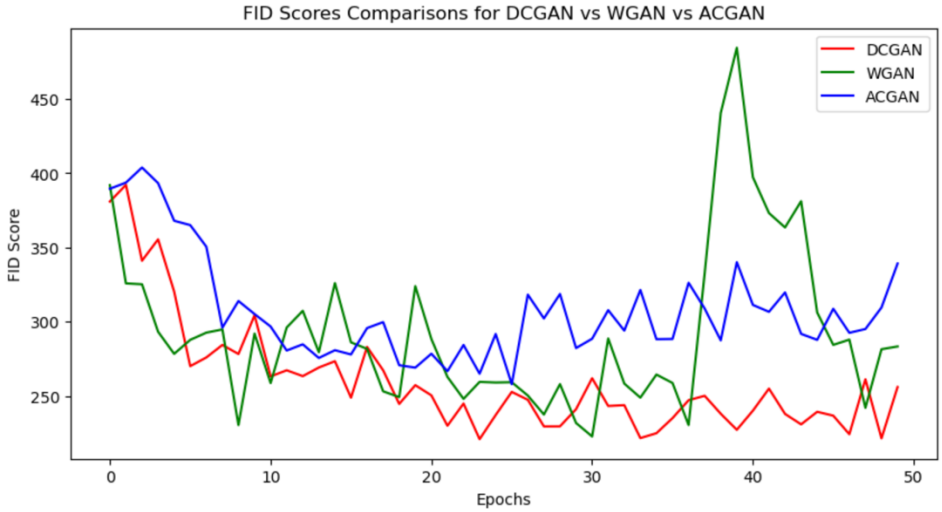




The images generated are below,



FID Comparison Analysis:



- The graph shows a comparison of Frechet Inception Distance (FID) scores for three different Generative Adversarial Network models: DCGAN, WGAN, and ACGAN, over the course of 50 epochs.
- Lower FID scores indicate better image quality produced by the model. The DCGAN generally maintains the lowest FID scores throughout the epochs, suggesting it consistently generates higher-quality images.
- WGAN shows more fluctuation and generally higher FID scores, indicating less stable performance in image quality.
- ACGAN while outperforming WGAN at the start, demonstrates a spike in variability and ends with higher FID scores, hinting at potential overfitting or instability in the later stages of training.
- Overall, DCGAN appears to have a performance advantage in generating more realistic images as shown by FID.