Final Project Machine Learning - CSE 847

Gaya Kanagaraj
Sahana Manjunath

Text-To-Image Generation

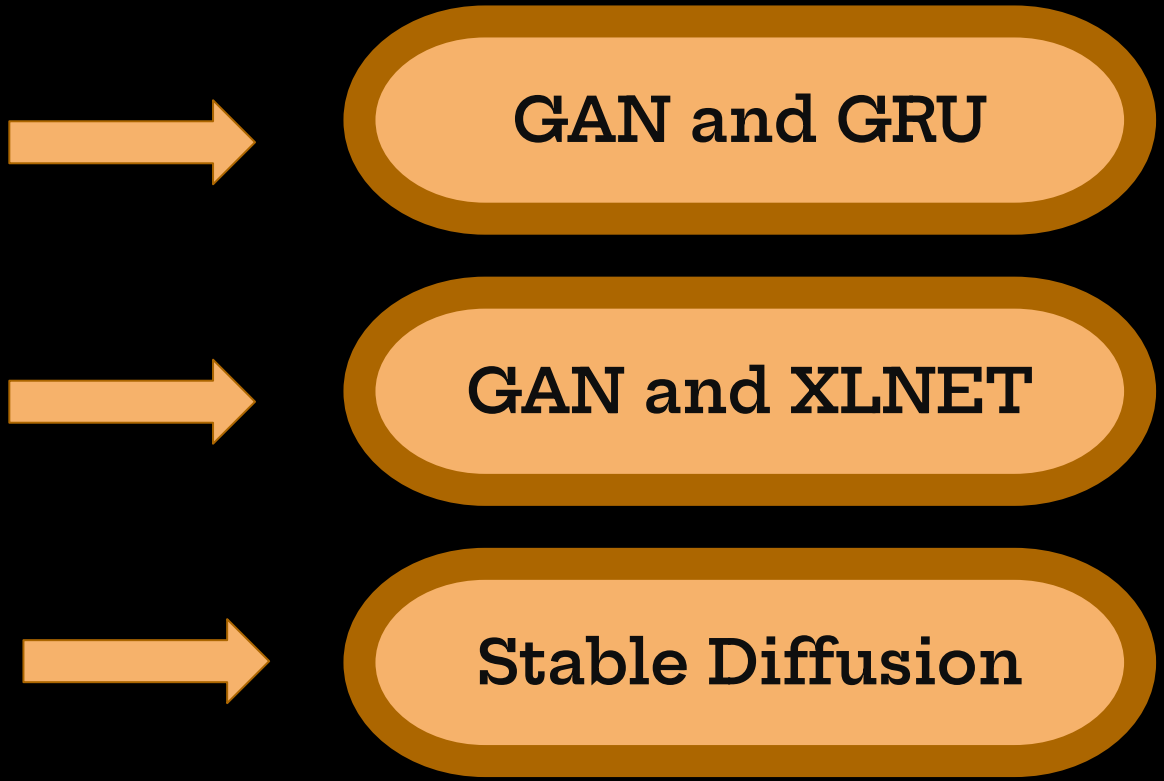
Introduction

Text-to-image generation has gained significant attention in recent years, driven by advancements in deep learning techniques. Traditional approaches, such as GANs (Generative Adversarial Networks) and conditional models, have shown promise in generating images from textual descriptions but often struggle with issues like low image fidelity and mode collapse.

Objective

The objective of this project is to explore and compare two text to image generation techniques - Discriminator Generator model and Stable Diffusion to generate high-quality flower images from textual descriptions. We aim to evaluate the effectiveness of each method in terms of image realism, diversity, and detail

Methodology



Evaluation Metrics

FID Score (Fréchet Inception Distance)

Evaluating the Quality of Generated Images

- Measures similarity between real and generated image distributions using feature embeddings.]
- Mean and covariance of Inception feature representations.
- Lower FID = Higher similarity (better quality).

The FID score is defined as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}),$$

where:

- μ_r and μ_g are the mean feature vectors of the real and generated images, respectively.
- Σ_r and Σ_g are the covariance matrices of the real and generated images, respectively.
- Tr denotes the trace of a matrix.

Evaluation Metrics

IS Score (Inception Score)

Assessing Quality and Diversity

- Evaluates clarity and diversity of generated images based on class probabilities.
- Kullback-Leibler divergence between conditional and marginal distributions.
- Higher IS = Better quality and diversity.

The IS is computed as:

$$\text{IS} = \exp\left(\mathbb{E}_x\left[\text{KL}(p(y|x)\|p(y))\right]\right),$$

where:

- $p(y|x)$ is the conditional probability distribution of class labels y given an image x.
- $p(y)$ is the marginal probability distribution of class labels across all images.
- KL denotes the Kullback-Leibler divergence.

Evaluation Metrics

Human Evaluation

Qualitative Assessment of Generated Images

- Text prompts are passed to the model to generate images.
- Semantic Alignment: Does the image match the text description?
- Realism: Does the image look visually plausible?
- Provided insights into aspects not captured by quantitative metrics.

Kaggle's Flower Dataset

Dataset Composition:

- Training Set: 29,390 images
- Validation Set: 5,780 images
- Test Set: 5,775 images

Image: High-quality flower images.

Text Description: E.g., "Prominent purple stigma, petals are white in color."

MODEL: GAN and GRU

Objective:

- Build a Generative Adversarial Network (GAN) from scratch to generate flower images conditioned on text descriptions.

Model Architecture:

- Text Encoder:
  - Embeds text descriptions into fixed-length vectors using a bidirectional GRU with attention.
- Generator:
  - Input: Noise (z) + Text embedding.
  - Architecture: Transposed convolutions with batch normalization and ReLU.
  - Output: 256 X 256 IMAGE
- Discriminator:
  - Input: Image + Text embedding.
  - Architecture: Convolutional layers with progressively increasing filters.
  - Output: Probability of the image being real.

Training Details:

- Dataset: Full training dataset (29,390 images).

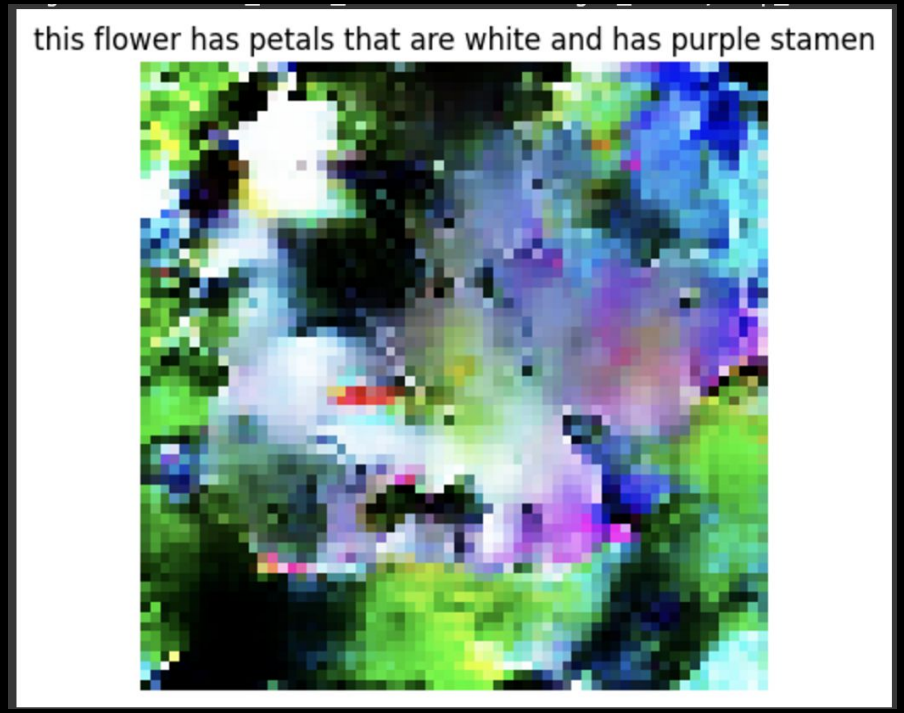
Epochs: 30 (limited by computational constraints).

- Loss:
  - Generator Loss: Fool discriminator into classifying generated images as real.
  - Discriminator Loss: Classify real and generated images correctly.
- Optimizer: Adam (b1 = 0.5, b2= 0.999).

MODEL: GAN and GRU

Key Observations:

- Generated images were not realistic (blurred color blends).
- Likely insufficient training due to limited epochs.
- Longer training (e.g., 1,000 epochs) could significantly improve results.



MODEL: GAN and XLNET

Model Architecture:

- **Text Encoder:**
  - It transforms text into embeddings using xlnet-base-cased pretrained model.
- **Generator:**
  - **Input**: Combines random noise vector and text embedding to conditionally generate images (Input to 1st layer).
  - **Layer-wise Architecture**: Consists of 11 layers, including transposed convolutions, and batch normalization.
  - **Regularization**: Uses Batch Normalization and Dropout to stabilize training and prevent overfitting.
  - **Output:** Generates a 64 x 64 RGB image with Tanh activation, conditioned on the text description.

- **Discriminator:**
  - **Input**: Takes an image (64x64) and text embedding, conditioning the image classification on the text description.
  - **Layer-wise Architecture**: Consists of 7 layers, including convolutional layers, batch normalization, and LeakyReLU activations.
  - **Text-Conditioning**: The text embedding is concatenated with the image features in the final layers to improve classification.
  - **Output**: Produces a probability of the image being real or fake (real/fake classification), with the final output flattened into a vector.

MODEL: GAN and XLNET

Training Configuration:

- **Dataset** - Full training dataset (29,390 images).
- **Epochs** - 20 (limited by computational constraints).
- **Loss** -
  - **Generator Loss:** Fool discriminator into classifying generated images as real.
  - **Discriminator Loss:** Classify real and generated images correctly.

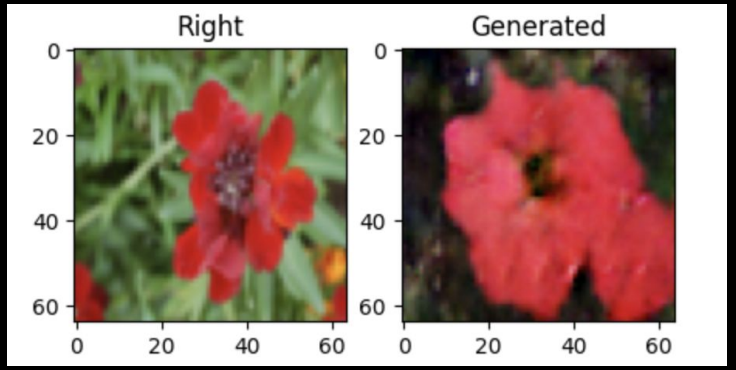
- **Optimizer** - Adam
- **Platform** - Google Colab Pro A100 GPU

MODEL: GAN and XLNET

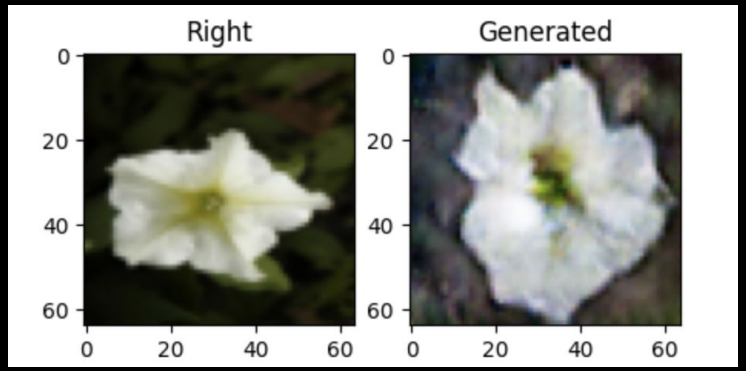
Key Observations:

- **Generated Images:** The images were conceptually correct but lacked sufficient clarity and diversity.
- **Training Duration:** Extending the training period with the same architecture did not significantly improve image quality.
- **Architecture Limitation:** The current model architecture lacks the complexity needed to capture fine details and generate more diverse, clearer images. A more advanced architecture is required to address these issues.

Red flower with rounded-shaped petals



White flower with yellow in center



MODEL: Stable Diffusion Fine-Tuning

Objective:

- Fine-tune the pretrained Stable Diffusion model to generate flower images based on textual descriptions.

Model Architecture:

- Text Encoder:
  - Pretrained CLIP model
  - Architecture: Vision Transformer (ViT) with 12 layers and 768 hidden units.
  - Outputs 768-dimensional text embeddings for up to 77 tokens.

- Diffusion Scheduler:
  - Manages forward (noise addition) and reverse (denoising) diffusion processes.
  - Uses a cosine noise schedule with 1,000 timesteps.

- U-Net: Core generative component of the diffusion process.

- Downsampling Path:
  - 4 blocks with residual convolutions and strided convolutions.
  - Middle Block:
  - Bottleneck with residual and attention layers for global context.
- Upsampling Path:
  - 4 blocks with transposed convolutions and residual layers.
- Cross-Attention:
  - Aligns text embeddings with visual features throughout the model.

MODEL: Stable Diffusion Fine-Tuning

Training Details:

- Dataset: 2,000 randomly selected images due to resource constraints.
- Epochs: 10 (45 minutes per epoch).
- Optimizer: AdamW (learning rate 1e-6 , gradient clipping at 1.0).

Key Observations:

- Generated images were of high quality due to the pretrained nature of the model.
- Limited dataset and epochs restricted further improvements.

MODEL: Stable Diffusion Fine-Tuning

Prompt : The petals of the flower has a hair-like texture, and consist of various shades of purple and blue.

Real Image

Generated Image

MODEL: Stable Diffusion Fine-Tuning

Prompt : This flower is white and yellow in color, with petals that are oval shaped.

Real Image

Generated Image

Results Overview

	FID Score	IS Score
GAN and XLNet	1355.40	Mean - 1.0 Std - 0.0
Stable Diffusion	737.77	Mean - 1.37 Std - 0.21

Conclusion

In conclusion, fine-tuning Stable Diffusion on the flowers dataset produced high-quality images with excellent clarity and a strong understanding of textual prompts. While GAN and XLNet generated conceptually correct images, they struggled with clarity and diversity. GAN and GRU performed the worst due to architectural limitations, making them less effective for generating detailed and varied images. In the future, this approach can be extended to more generic datasets, enabling the generation of diverse and high-quality images across a broader range of topics.

Thanks , Any questions?