Text-to-Image Generation System - Milestone 3

Group 4 submission: https://github.com/rahulodedra30/Text2ImageGen

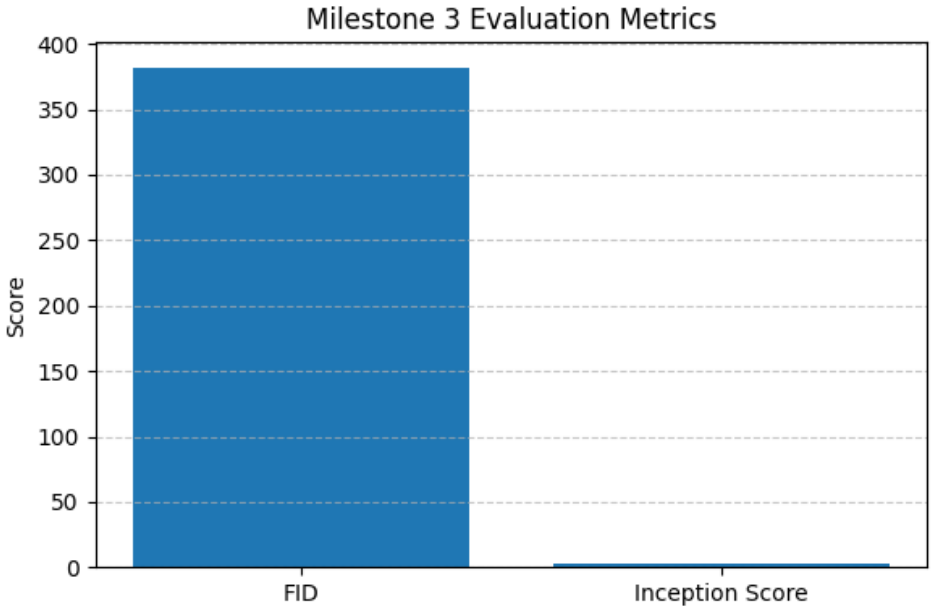
1. Overview

This milestone focused on evaluating the performance of our Stable Diffusion–based text-to-image system. We generated a full evaluation dataset, computed FID and Inception Score, analyzed model behavior, and documented a complete notebook-level workflow analysis.

2. Evaluation Metrics

We calculated two major generative quality metrics:

Metric	Value
FID	381.9514
Inception Score	2.70834



Interpretation:
The model produces recognizable but low-detail images. FID suggests high divergence from the real data distribution, while IS indicates low semantic coherence and limited diversity.

3. Work Summary

Environment Setup

- Installed PyTorch, Diffusers, and evaluation libraries
- Verified CUDA/GPU availability

Directory Setup

- Created folders for raw images, generated images, and evaluation outputs

Model Loading

- Loaded Stable Diffusion 1.5
- Applied DDIM scheduler for improved inference
- Loaded and inspected UNet weights

Baseline Test Generation

- Generated a test image using a simple sanity-check prompt

Bulk Image Generation

- Generated **100 images** using dataset captions for evaluation

Metric Computation

- Calculated **FID** and **Inception Score**
- Exported results to CSV
- Generated a bar plot for visual comparison

4. Notebook Analysis (0–100% Breakdown)

4.1 Environment Setup

The notebook installs CUDA-enabled PyTorch and imports all required libraries including torch, diffusers, PIL, datasets, and utility modules. GPU availability is printed to confirm hardware readiness.

4.2 Directory Setup

Using os.makedirs(), the notebook prepares structured folders for raw images, generated images, and evaluation files, enabling reproducible experiment organization.

4.3 Model Loading & Scheduler Selection

Stable Diffusion 1.5 is loaded using the Diffusers pipeline.
A **DDIM scheduler** is selected to produce smoother and faster denoising steps.
UNet weights are optionally loaded for partial fine-tuning or checkpoint testing.

4.4 Single-Prompt Baseline Testing

A prompt - “A golden retriever playing in green field”—is passed to the pipeline. This verifies:

- Text encoding
- Latent denoising
- VAE decoding
- Inference timings

4.5 Bulk Image Generation for Evaluation

The notebook loads 100 captions from the dataset and generates corresponding images. Each output is saved into a structured evaluation directory.
This sample set is used for metric computation.

4.6 Strange Prompt Inspection

The first prompt that produced distorted outputs is printed.
This step helps debug semantic alignment failures.

4.7 Metric Tool Installation

The notebook installs:

- torchmetrics
- pytorch-fid
- scipy

These libraries support:

- FID computation
- Inception Score
- Image preprocessing

4.8 FID & Inception Score Calculation

Real images and generated images are compared using FID.
Generated images are passed through an Inception network to compute IS.
The results are saved in evaluation_scores.csv.

4.9 Final Metrics Plot

A bar plot comparing FID and IS is created and saved to metrics_plot.png. This serves as the final visual summary of Milestone 3 performance.

5. Next Steps Before Final Milestone

To significantly improve results:

- Increase dataset size to 4–5k samples
- Train for 20–30 epochs instead of short runs
- Switch to LoRA fine-tuning for efficient, high-quality updates
- Use advanced schedulers such as DPM++, Euler, or DDIM
- Optimize LR, batch size, and prompt formatting
- Add CLIP Score and Precision/Recall metrics
- Produce side-by-side comparisons for demo
- Explore 512×512 training for higher fidelity

6. Conclusion

Milestone 3 successfully delivers a complete evaluation and detailed notebook analysis. Although results show early-stage performance, all issues are clearly identified, and clear optimization paths are set for the final milestone.