Project Title

Text-to-Image Generation Using CLIP and Stable Diffusion

---

1. Problem Statement

Creating realistic images from text is one of the most exciting developments in artificial intelligence. Models like Stable Diffusion and DALL·E have shown that it is possible to turn simple text descriptions into detailed, creative visuals.

However, this process is still far from perfect. Models often generate images that do not fully match the input text, or they miss subtle details in descriptions. It can also be difficult to control the visual style or composition of the generated images.

Our project aims to build a working text-to-image system using two powerful pretrained models: **CLIP** (which understands text and connects it to visual meaning) and **Stable Diffusion** (which generates images). By combining them, we want to explore how text meaning can be accurately turned into images and how small changes in model settings affect the results.

---

2. Dataset

We are using the **COCO 2017 Validation Set**, a well-known dataset that includes images paired with human-written captions.

From this dataset, we selected a smaller subset of **5,000 image–caption pairs** that offer a wide range of scenes and objects. This subset keeps the data diverse while being small enough to process efficiently.

The cleaned and preprocessed data is saved in:

data/coco_small/pairs.csv

Each entry links an image path with its caption.

---

3. Model Framework

Our project uses two pretrained models working together inside a PyTorch environment built with the Hugging Face Diffusers library.

Component	Role	Description
CLIP (ViT-L/14)	Text Encoder	Converts each caption into a numerical vector that represents its meaning.
Stable Diffusion v1.5	Image Generator	Uses those text embeddings to generate new images by gradually removing noise from random patterns.

Pipeline Overview

1. Read image–caption pairs from the dataset.
2. Use CLIP to encode each caption into a text embedding.
3. Send those embeddings to Stable Diffusion to generate images.
4. Adjust parameters such as guidance scale and number of inference steps.

5. Save and review the generated images.

This setup lets us observe how well the diffusion model can produce visuals that reflect text meaning and how tuning parameters influences the results.

---

4. Objectives

- Build and test a complete text-to-image pipeline.
- Generate a baseline set of sample images using pretrained models.
- Explore how parameter settings affect image quality.
- Prepare for future evaluations using FID and Inception Scores.
- Optionally add controls for artistic style or color tone.

Member	Responsibility
Gagan Yadav	Leads dataset preparation, data cleaning, and creation of the 5k image–caption subset. Also contributes to writing the methods section of the report.
Sakshi Aryal	Sets up the environment, integrates CLIP embeddings, and supports code documentation for reproducibility.
Rishik Ganta	Handles Stable Diffusion experiments, parameter tuning, and evaluation of generated images. Also assists in preparing figures and visuals for the final report.
Yash Harale	Coordinates the overall workflow, manages GitHub, and contributes to code testing and writing the final analysis and discussion sections.

---

6. Expected Deliverables

- Cleaned dataset subset (pairs.csv)
- A working CLIP + Stable Diffusion notebook
- Five text-to-image sample outputs
- GitHub repository with code and documentation
- A one-page summary of technical steps and findings

---

7. Future Work

If time allows, we would like to explore:

- Using GPT-style text encoders to test if richer text embeddings improve results.
- Adding style control inputs to guide art style or color themes.
- Comparing our diffusion approach with transformer-only image generation models.