Text-to-Image Generation System - Project Proposal

GitHub Repository**:** https://github.com/rahulodedra30/Text2ImageGen

Project Overview

This project aims to develop a custom text-to-image generation system using diffusion models that creates realistic images from natural language descriptions. The system will leverage state-of-the-art deep learning architectures (CLIP + Stable Diffusion) to enable intuitive image synthesis through textual prompts, with applications in creative design, content generation, and AI-assisted visualization.

Objectives

Primary Goal: Build and train a text-conditioned diffusion model capable of generating high-quality 256×256 pixel images from natural language descriptions.

Key Objectives:

1. Curate and preprocess a dataset of 3,000-5,000 image-caption pairs
2. Integrate CLIP text encoder with Stable Diffusion architecture
3. Implement and optimize conditional image generation pipeline
4. Evaluate model performance using quantitative metrics (FID, CLIP score) and qualitative assessment

Dataset

Primary Dataset: We are using Flickr30k dataset from huggingface - lmms-lab/flickr30k

Specifications:

- Size: 31,695 image-caption pairs (6357 images with 5 captions each)
- Train/Validation Split: 80/20 (25,356 training / 6339 validation)
- Image Resolution: 256×256 pixels (resized from variable sizes)
- Caption Quality: 5-50 words per caption, cleaned and normalized

Data Preprocessing Pipeline:

- Images: RGB conversion, resizing to 256×256, normalization to [-1, 1] range
- Captions: Lowercase conversion, special character removal, whitespace normalization, length filtering

Rationale for Flickr30k:

- Diverse real-world photography covering various subjects and scenes
- Multiple captions per image provide rich semantic descriptions
- Manageable dataset size for prototyping and rapid iteration
- Well-established benchmark with high annotation quality

Data Statistics:

- Average caption length: ~12 words
- Image diversity: People (45%), outdoor scenes (30%), objects (25%)
- Caption vocabulary: ~5,000 unique tokens

Model Architecture

We plan to use CLIP (Contrastive Language-Image Pre-training) text encoder, because it effectively aligns text and image representations in a shared embedding space, enabling meaningful cross-modal understanding.

Model: openai/clip-vit-base-patch32
Function: Convert natural language prompts into dense semantic embeddings

Architecture Details:

- Input: Text prompts (max 77 tokens)
- Output: Embeddings of shape [batch_size, 77, 768]
- Pre-trained on 400M image-text pairs for robust semantic understanding

Image Generation: Stable Diffusion

Core Components:

1. VAE (Variational Autoencoder)
    - Compresses 256×256 images to 32×32 latent representations (8× compression)
    - Encoder: Image → Latent space
    - Decoder: Latent space → Image
    - Training Strategy: Frozen

2. UNet (Denoising Network)
    - Iteratively removes noise from latent representations
    - Cross-attention layers integrate text embeddings at each denoising step
    - 860M parameters with skip connections and self-attention
    - Training Strategy: Fine-tuned on Flickr30k dataset

3. Noise Scheduler (DDPM)

- 1,000 timestep diffusion process during training
- Configurable inference steps (25-100 for speed/quality trade-off)
- Implements classifier-free guidance for improved text alignment

Evaluation Metrics

Quantitative Metrics:

1. FID (Fréchet Inception Distance): Measures distribution similarity between generated and real images (lower is better)
2. CLIP Score: Text-image alignment quality (higher is better)
3. Training Loss: Convergence monitoring

Qualitative Assessment:

- Visual quality inspection across diverse prompts
- Semantic alignment between text and generated content
- Diversity of outputs for similar prompts
- Edge case handling (complex descriptions, rare objects)

Roles

Role	Team Members	Responsibilities
Data Engineering	Rahul, Keshika	Dataset curation, cleaning, preprocessing pipeline
Model Development	Pramoth, Haritha	Architecture integration, training pipeline implementation
Experimentation	Pramoth, Rahul, Haritha, Keshika	Hyperparameter tuning, ablation studies
Evaluation & Documentation	Pramoth, Rahul, Haritha, Keshika	Metric implementation, qualitative analysis, code documentation, technical report