PROMPT BLENDING PROJECT

This project creates a Gradio interface to create a visually pleasing image by blending two text prompts . The system intelligently combines the content of both prompts while maintaining a visual appeal using CLIP, Groq, and Stable Diffusion.
The goal of this is to finally create meaningful images which contain elements from both the prompts.

For Eg : on entering SNOW and CAKE as prompts, this was the image as output:

COMPONENTS

1)Prompt Enhancement

The first step after user enter prompts is their enhancement using **Groq's Llama3-70B.** This step:

- Adds descriptive details to the prompt (which the user might not always give) so as to create better images. Doing this, guides the model to produce more accurate, visually rich images. Descriptors like lighting, environment, mood, and

composition give Stable Diffusion clearer context, improving image quality. Without these details, the model may generate bland or generic visuals.

- Apply token weighting..This was primarily done because Stable Diffusion(model which we finally use to generate images) understands token weighting. **Token weighting** in Stable Diffusion lets you emphasize certain words or concepts in a prompt by assigning them higher weights using syntax like (word:1.6). A higher weight (e.g., 1.6) tells the model to pay more attention to that word.This is a crucial step because without token weighting ,the model might put focus on any one of the other descriptive details. It is therefore necessary to let the model know which part of the enhanced prompt is of maximum focus.

Finally the simple two prompts snow and cake are enhanced to this:

```
Enhanced Prompt A: Softly lit (1.1) snowflakes (1.1) gently falling (1.1) on serene (1.1) winter (1.1) landscape (1.1)
Enhanced Prompt B: A beautifully lit, golden brown cake (cake:1.65) on a rustic wooden table.
```

IMAGE BEFORE PROMPT ENHANCEMENT:

2) Stable Diffusion Image Generation

In this project, I use Stable Diffusion v1.5 to turn two text prompts into one blended image.

- **Turning Prompts into Embeddings**
  First, both enhanced prompts are turned into something called embeddings. These are just number-based representations that help the computer understand the meaning of the prompts. You can think of embeddings like a map where each word has a position, so similar ideas are closer together

- **Mixing the Prompts with Alpha**
  To combine the two prompts, I use a method called linear interpolation using the formula:

  *blended_embeddings = alpha * embeddings_a + (1 - alpha) * embeddings_*

  This means I mix the two sets of embeddings using a slider called alpha. If alpha is 0.5, both prompts are equally mixed. If alpha is closer to 1, the image leans more toward the first prompt. If it's closer to 0, it leans more toward the second prompt.But instead of using just one alpha value, I try many values between 0.4 and 0.6. This small range makes sure that the final image includes features from both prompts, not just one. If alpha is too close to 0 or 1, the image might only reflect one prompt, which we don't want.

- **Generating the Image**
  Once the blended embeddings are ready, they are passed into the Stable Diffusion model, which then creates a final image based on this mix. This lets us create a smooth, balanced image that feels like it belongs to both original ideas.

This whole process helps turn two separate concepts into one visually pleasing and meaningful image.

3) Choosing the right value of alpha / Aesthetic Score

The process of blending two prompts into a single, coherent image involves more than just combining their embeddings. One of the key challenges is ensuring that the final image reflects both prompts fairly, without one dominating the other. This is where the parameter alpha plays a crucial role—it controls the relative influence of each prompt's embedding in the blended representation.Choosing a single fixed alpha can lead to biased results, especially when one prompt is more visually pleasing than the other.

To address this, a range of alpha values from 0.4 to 0.6 are taken, generating an image for each. This range avoids extreme bias toward either prompt and increases the chances of producing a visually and semantically balanced image. The best image out of all the generated images for different values of alpha is done through aesthetic scoring using CLIP (Contrastive Language-Image Pretraining). A list of some type of aesthetic words is made which also contains the original prompt_a and prompt_b with additional descriptors like *"aesthetic, visually pleasing"*. The purpose of this list is to help find the best image using the logic that how similar is the final image to these words. More similar the image is to them, the better the image. Prompts provided by the user are also contained in the image so as to calculate similarity of the final image with prompts provided by the user.

To evaluate each generated image:

1. The image is passed through CLIP's image encoder to produce a feature vector representing its visual content.

2. The aesthetic prompt is passed through CLIP's text encoder to produce a text feature vector.

3. Both vectors are normalized, and the cosine similarity between them is computed. This measures how closely the image aligns with the prompt in a high-dimensional semantic space.

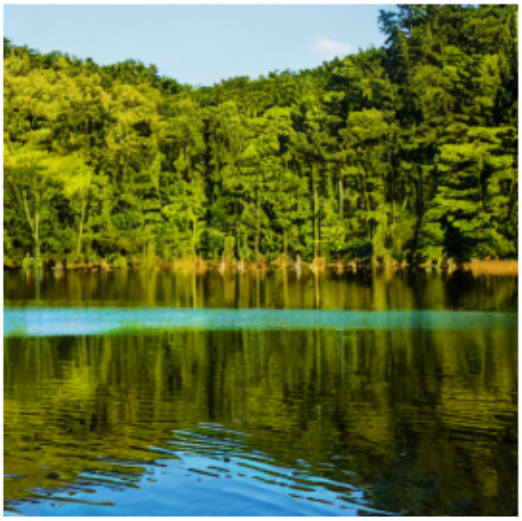
The cosine similarity score ranges from -1 to 1, with higher values indicating a higher similarity. This score effectively quantifies how well each image captures both the meaning and artistic intent of the combined prompts. The image with the highest similarity score is selected as the final output.

In summary, this approach ensures that blending is not just mathematically balanced but also visually pleasing and a perfect blend of both.

IMAGES AT EXTREME VALUES OF ALPHA AND FOR ALPHA BETWEEN 0.4-0.6

PROMPTS: LAKE AND PARK

α=0.20
Score=0.53

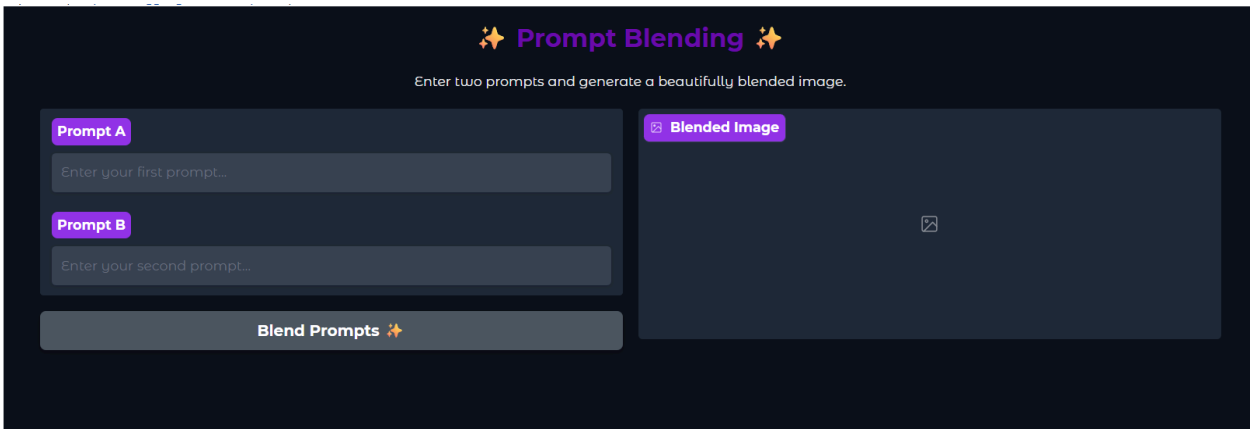


α=0.52
Score=0.55

α=0.77
Score=0.53

Gradio Interface

Finally, Gradio app creates a simple web interface where users can enter two text prompts. When the "Blend Prompts " button is clicked, it returns a blended image. The image is then displayed on the right side of the interface.



Real-World Challenges and Problem Faced

When I first combined two user prompts directly in Stable Diffusion, the images often skewed heavily towards one concept—either I'd get a snowy landscape with barely a hint of cake, or a cake piled so high it buried the snow entirely.This was possibly because any one of two prompts alone made more visually pleasing image.

- Without careful handling, subtle elements vanish. Combining "snow" and "cake" naively makes the model chase one dominant feature, leading to incoherent blends.

- Lack of descriptive context produced bland visuals—snow looked flat, cake textures lacked depth.
- Generating dozens of images for different blend ratios hampered quick experimentation and consumed GPU memory.

To resolve these, I introduced prompt enhancement, token weighting, and a guided interpolation strategy, each tuned for clarity and efficiency.

Mindset Behind Prompt Engineering

Having the right prompts for generating image is very important. Here's how I think about it:

Keep It Short and Clear: I try to keep prompts around 10 words. If the prompt is too long, the model gets confused. If it's too short, the image may not turn out well.

Highlight What Matters: I use special weights to show what's most important. For example, I give the main subject a stronger weight like (snow:1.7), so the model focuses on it. Less important details like background or mood get a lighter weight, like (calm:1.1).

Use Descriptive Words: I pick strong and clear words like "soft lighting" or "wide-angle view" to help the model understand the look and feel I want in the image.

Stay Consistent: I use a low temperature (0.3) in the language model to keep the enhanced prompts steady and not too random.

This way, I turn simple user prompts into short, powerful instructions that guide the image model clearly.

Linear Interpolation: Present and Future Perspectives

Current Strategy: I blend two embeddings with:

*blended_embeddings = alpha * embeddings_a + (1 - alpha) * embeddings_b*

iterating alpha from 0.4 to 0.6 in 20 steps. This narrow window maintains balance—neither prompt overtakes the other. capturing the essence of both inputsBest alpha is found using the similarity between final image and the aesthetic words we provided..

Future Aspect:

- Smart alpha selection: Analyze embedding similarity to focus on alpha range.
- Non-Linear Blends: Use curved or attention-weighted paths through embedding space to enhance creative blending.
- Generate desired images for any two types of prompts (even if prompts are very far away in latent space)
- A better method to score our image on the basis of aesthetics and how much it contains both the prompts.

These extensions would help in making better images.
