Name: Irish Mehta
ASU ID: 1233414002

Mid Semester Assignment Report

Task 1

Aim: Learn about Zero Shot Classification on a sample image, a speciality of CLIP

Challenges: Understanding the usage of logits_per_image method and using those to get the class probability

Results:

Class Descriptions: ["a representation of a cat", "a representation of a dog", "a representation of a giraffe", "a representation of an elephant","a representation of a horse"]

Output: The predicted class is cat and the probability scores are:

cat: 0.9869
dog: 0.0068
giraffe: 0.0047
elephant: 0.0002
horse: 0.0014

Experiment: If we change the class descriptions-
Class Descriptions: ["a blurry photo of a cat","a close-up photo of a furry animal with whiskers","a photo of a four-legged pet", "a photo of a domestic animal looking at camera", "a photo of a mammal with pointed ears"]

Output: The predicted class is cat and the probability scores are:
cat: 0.4946
dog: 0.0276
giraffe: 0.2837
elephant: 0.1887
horse: 0.0055

Conclusion: The probability of classes significantly changes with major changes in prompt descriptions.

Task 2

Aim: Learn how robust CLIP's Zero Shot Learning is to adversarial changes (like adding noise, blurring and rotating the image)

Challenges:

1) Documentation about image editing by going through pillow library
2) Learning how to use intensity to control the level of editing in the input image

Results:

Perturbation Type: noise

Perturbation Type: rotate

Perturbation Type: blur

The predicted class is cat and the probability scores are:
cat: 0.5470
dog: 0.0020
giraffe: 0.0525
elephant: 0.3725
horse: 0.0259

Experiment: When the intensity of perturbations is doubled than that of first attempt

The predicted class is elephant and the probability scores are:
cat: 0.0877
dog: 0.0104
giraffe: 0.0501
elephant: 0.4965
horse: 0.3552

Conclusion: CLIP is quite robust when the intensity of perturbations till a certain extent, beyond which it is unlikely to detect the correct class. For moderate perturbations, it predicts the “cat” label with high probability, but with extremities, it fails.

Part 1 Supplementary Questions

Q1- How does CLIP generate embeddings for both images and text, and why is this dual-encoding approach powerful?

Ans- To generate text embeddings, they create a text encoder based on a modified Transformer architecture. The activations of the highest layer of the transformer are taken as the feature representation of the text and projected into the embedding space. For images, they use a ResNet image encoder with a modification. They allow additional computing across width, depth, and resolution, which outperforms the ResNet image encoder. The power of this approach lies in its use of contrastive learning, aligning text and image embeddings. When an Image model is used separately for classification, it requires very heavy computation and could be more efficient and perform better on unseen examples. However, using dual-encoding, when given a batch of N (image, text) pairs, CLIP is trained to predict which of the N × N possible (image, text) pairings across a batch actually occurred. To do this, CLIP learns an embedding space by jointly training an image encoder and text encoder to maximize the cosine similarity of N real (image, text) pairs in the batch while minimizing that of the N^2 − N incorrect pairings. This enables the model to generalize better, improving performance on zero-shot tasks by learning shared representations across modalities.

Q2- How might the choice of class names affect the classification performance?

Ans- In CLIP's zero-shot classification framework, the choice of class names significantly influences performance. Section 3.1.4 of the original CLIP paper, the authors highlight that CLIP maps images to class names in a shared embedding space, performing classification by selecting the class whose name has the highest similarity to the image. Therefore, the specificity and descriptiveness of class names are crucial. Vague or ambiguous class names may not capture the model's attention effectively, leading to suboptimal classification. Conversely, well-chosen, descriptive class names can enhance the model's ability to accurately associate images with the correct categories, thereby improving classification performance.

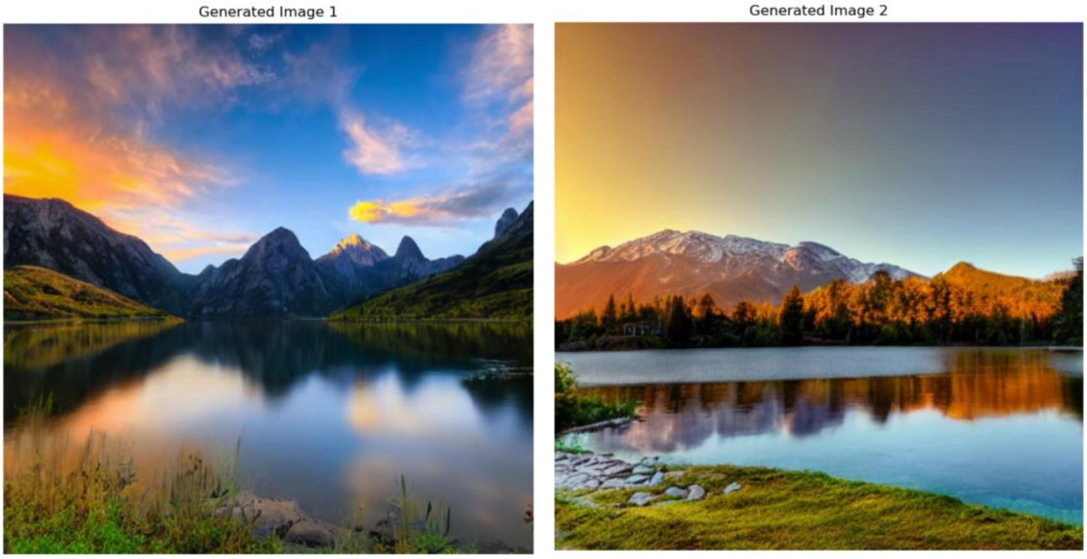
Task 3

Aim: Learn to use a pre-trained Stable Diffusion pipeline on a custom prompt

Challenges:

1) Learning about the hyperparameters required to generate an image

Results: *prompt = "A serene landscape with mountains and a lake at sunset"*



Conclusion: Based on the input prompt and parameters, the model generates single/multiple images and sorts them in order of closeness to the prompt

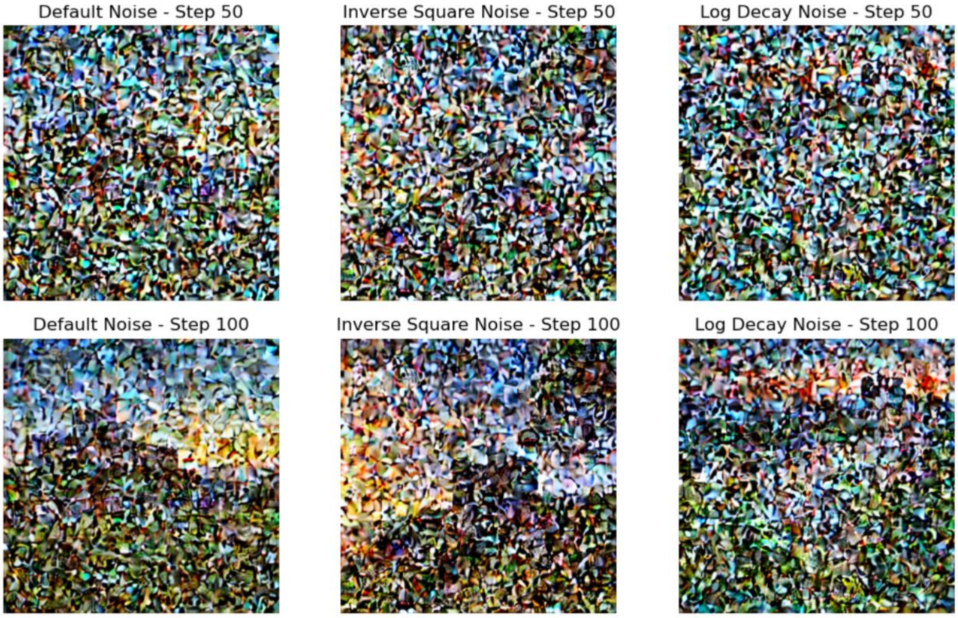
Task 4

Aim: Learn to use a custom noise schedule which can be used in the forward/reverse diffusion process for image generation

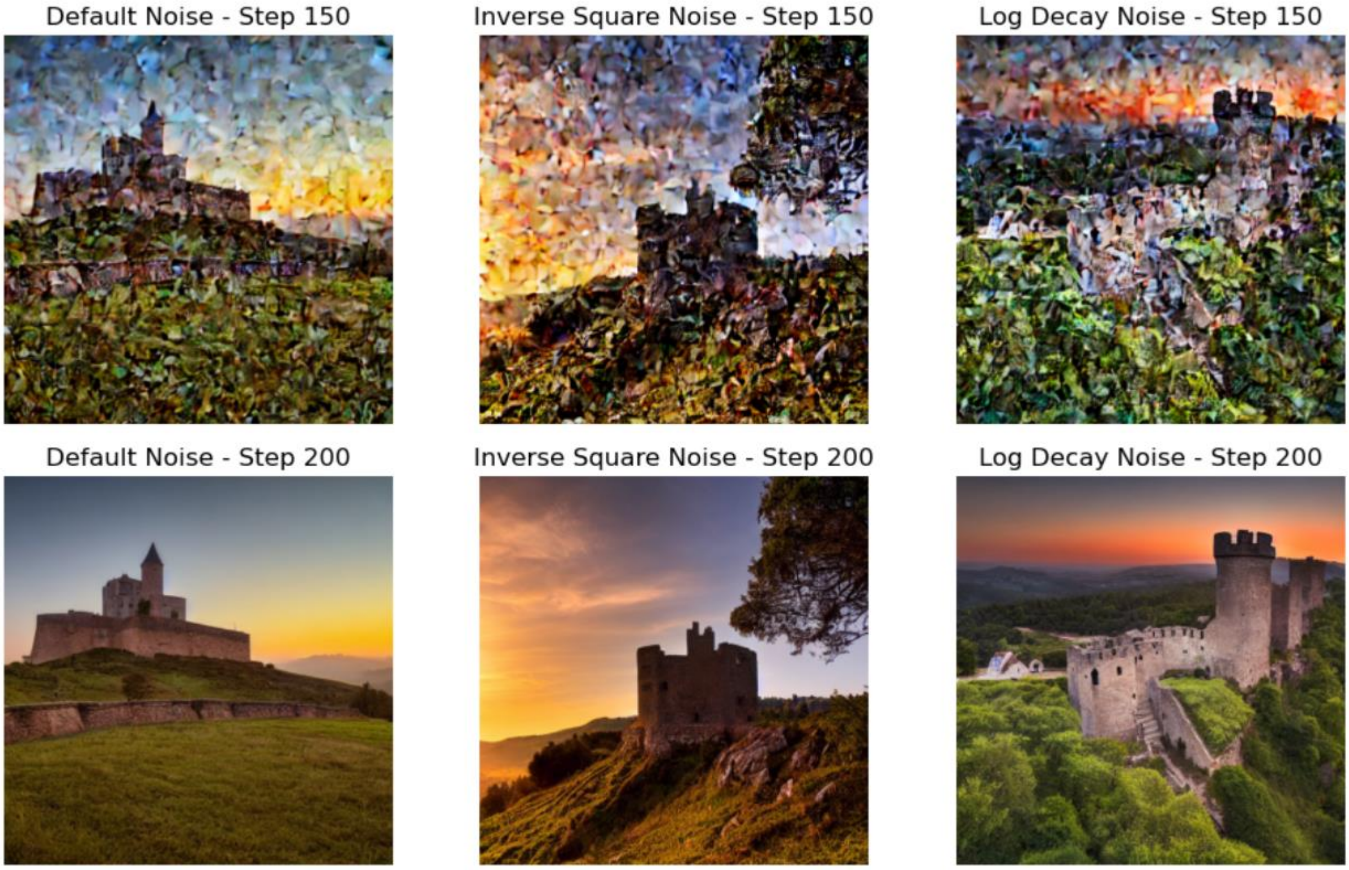
Challenges:

1) Figuring out how to extract the image generated at an intermediate stage
2) Figuring out how noise affects the generated output
3) Go through scheduler documentation (set pipe.scheduler.betas = noise schedule tensor)

Results: Prompt: *"An ancient castle on a hill during sunrise"*


Default Noise - Step 50
Inverse Square Noise - Step 50
Log Decay Noise - Step 50
Default Noise - Step 100
Inverse Square Noise - Step 100
Log Decay Noise - Step 100

Name: Irish Mehta
ASU ID: 1233414002

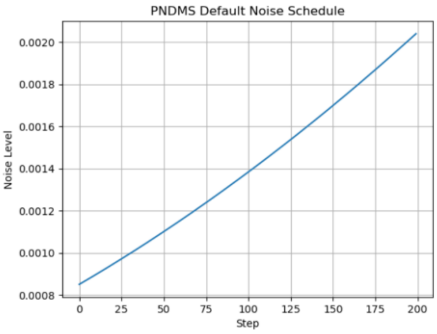

Default Noise - Step 150
Inverse Square Noise - Step 150
Log Decay Noise - Step 150
Default Noise - Step 200
Inverse Square Noise - Step 200
Log Decay Noise - Step 200

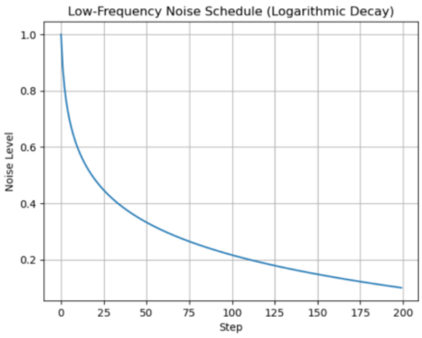
Default Schedule: Balanced representation of both high-frequency details (textures) and low-frequency details (lighting, smooth transitions).

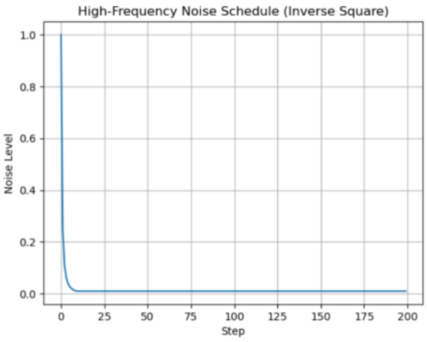
Inverse Square Schedule: Strong emphasis on high-frequency details (sharp textures, edges, shadows), resulting in crisper features but possibly less smoothness in larger areas.

Logarithmic Decay Schedule: Focus on low-frequency details (smooth color gradients, lighting transitions), resulting in smoother images but with less emphasis on fine textures and sharp edges

For reference, these are the noise schedules being tested-







Conclusion: Noise schedules can significantly impact features attributes in the image. However, one thing I noticed is the issue of reproducibility. I cannot set a manual seed when I'm using custom noise, because setting a seed enforce pseudo randomness, while a custom noise schedule requires completely random generation of noise, so when we add custom noise schedule + random seed, we essentially do not see any difference in the intermediate steps

Task 5

Aim: Exploration of latent space for image interpolation

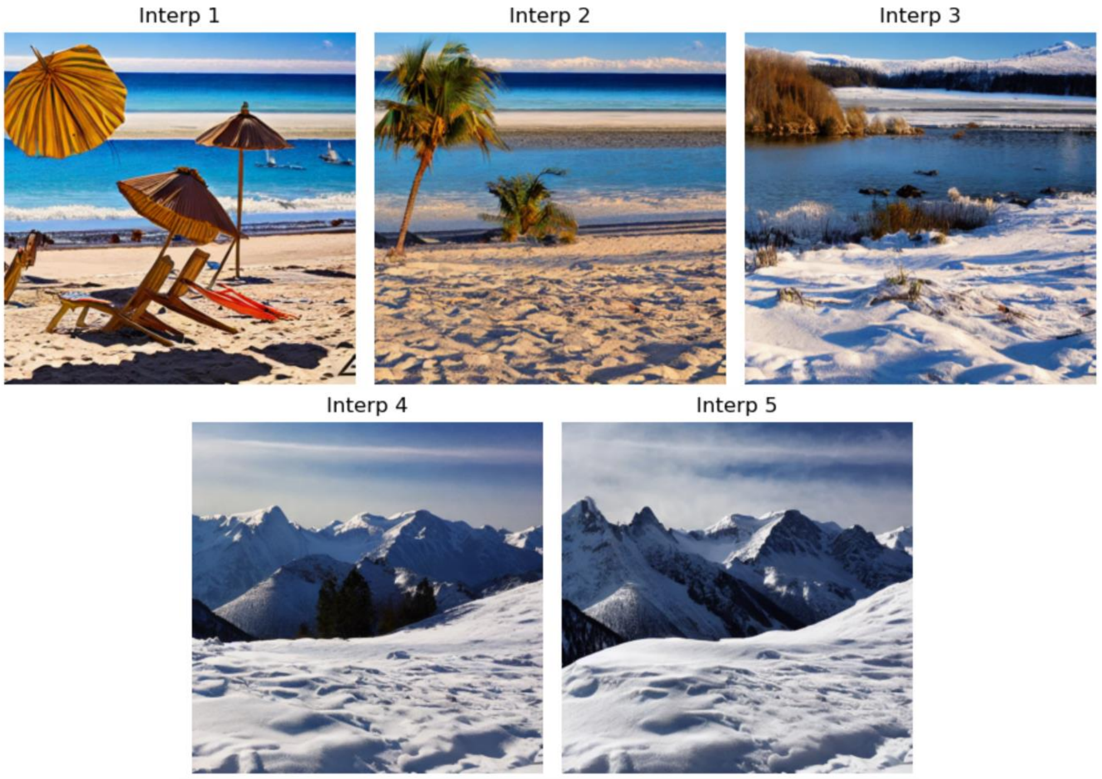
Challenges:

1) Generating the interpolated embedding
2) Using embedding to generate an image instead of a prompt

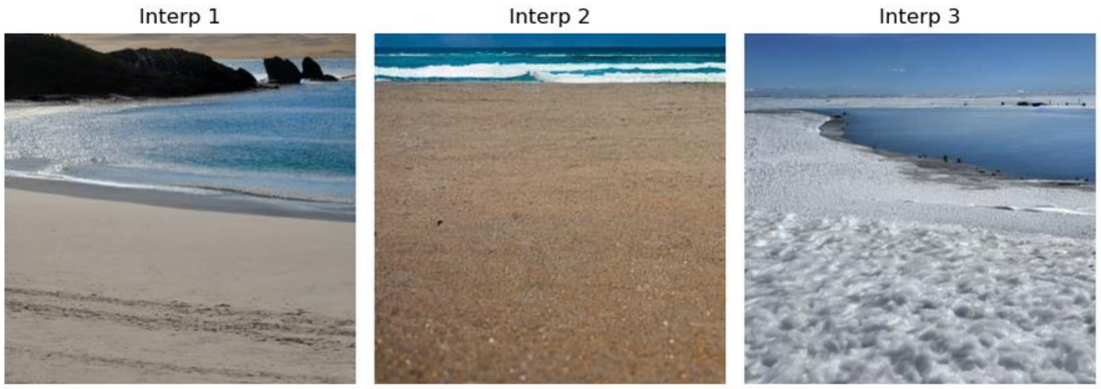
Results:

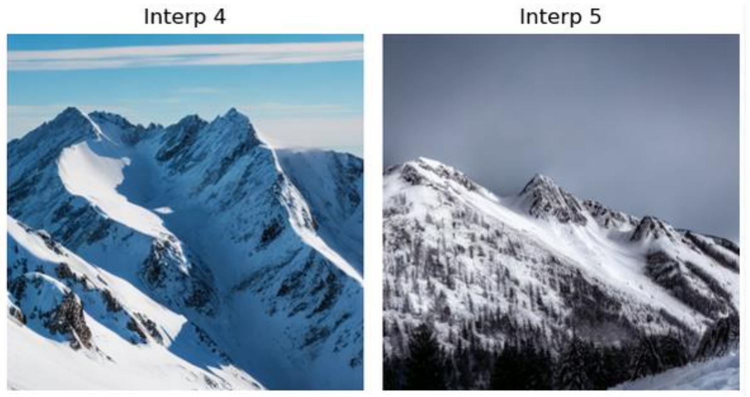
Prompt 1: *"A sunny beach"*

Prompt 2: *"A snowy mountain"*



Previous Attempt (also successful)





Conclusion: Interpolation of embeddings helps understand intermediate transitions, and the models' capability to combine attributes from both the prompts. In the above examples, the intermediate pictures move from a sand beach to a snow beach and then to snowy mountains.

Task 6

Aim: Exploration of latent space to modify particular attributes

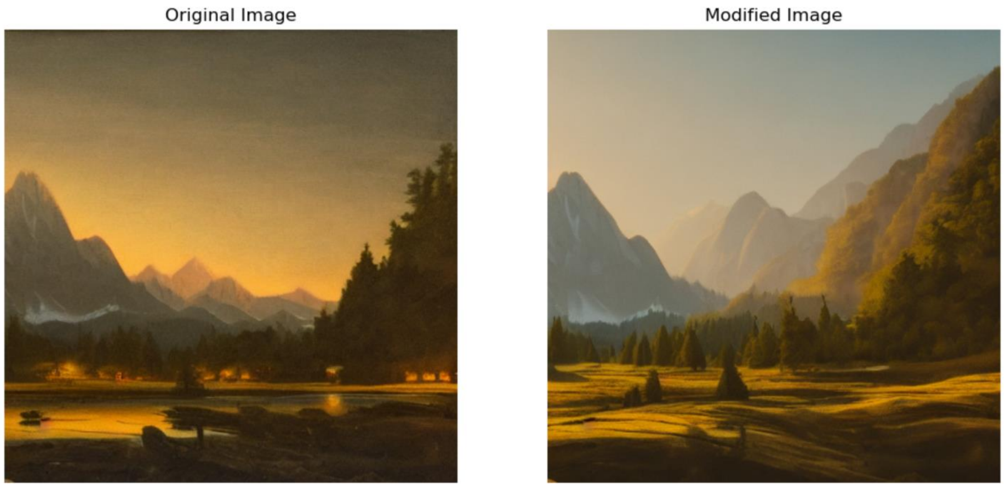
Challenges:

1) Learning about classifier free guidance using stable diffusion
2) Modifying the latents to incorporate the predicted noise (at every step, new latents= old latents – predicted noise)

Results:

Prompt 1: *"A serene landscape with mountains in night"*

Prompt 2: *"A serene landscape with mountains during daylight"*



Conclusion: The output images are noticeably similar, with the exception of a certain change to the mountain structure. But the attribute has changed from night time to daytime. We can achieve better results with a thorough fine-tuning and using more iterations

Task 7

Aim: Finding out the optimal guidance scale parameter for image generation

Challenges:

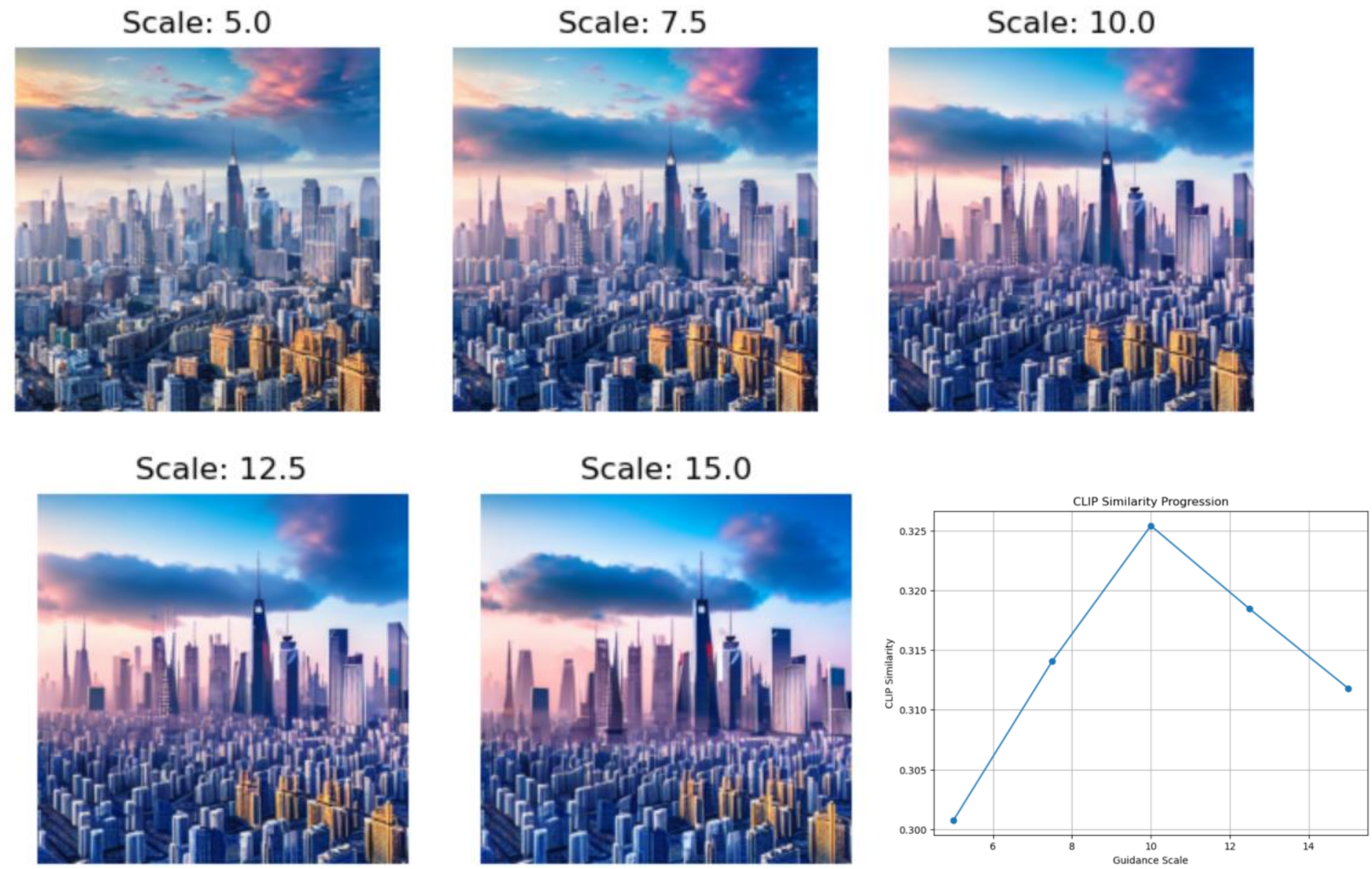
1) Manual calculation of cosine similarity using clip embeddings of both prompt and generated image

Results:

Prompt: "A futuristic city skyline at morning"

Negative Prompts: "lowres, bad anatomy, bad hands, cropped, worst quality, low quality, jpeg artifacts, watermark, blurry, out of frame”

Quality Enhancement Prompts: ", high resolution, highly detailed, masterpiece, best quality, professional"



Conclusion: Although the images look similar here, there are minor differences which are accounted for when calculating the CLIP similarity as shown in the graph. Moreover, adding two enhancements- negative prompt as well as quality enhancement prompts have significantly improved the quality of the generated images. These quality enhancements are generalized and not specific to any input prompt

Part 2 Supplementary Questions

Q1- Can you design a noise schedule that prioritizes certain image features or styles?

Ans: Yes, a noise schedule can be designed to prioritize certain image features or styles in diffusion models. By controlling how noise is added and removed during the diffusion process, it’s possible to emphasize or retain certain characteristics. For example, a custom noise schedule can be designed to preserve high-frequency details (like edges or textures) as shown in Task 4. Moreover, we can also allowing lower-frequency features (like color and composition) to create images with smoother color transitions and realistic effects as shown in Task 4. Similarly, certain noise schedules can bias the model to follow a particular pattern or abstraction for specific styles, retaining stylistic elements. This approach requires control over the denoising steps.

Q2- What insights can be gained by interpolating between latent vectors of different prompts? How does this relate to the model’s understanding of the concepts?

Ans: Interpolating between latent vectors of different prompts shows the model's understanding of how concepts blend or transition in their learned space. This technique allows us to explore how the model encodes features like style, objects, or abstract ideas across different prompts. As latent vectors are interpolated, the resulting images show gradual changes, blending attributes from both prompts. For instance, interpolating between “a cat” and “a dog” might show hybrid creatures, indicating how the model generalizes concepts of animals. This reflects the model’s high-level conceptual understanding, as it smoothly transitions between representations while preserving the core aspects of both prompts. Such interpolations provide insight into the structure of the latent space and how the model organizes related concepts, offering an insight into its internal "knowledge" framework.

Task 8

Aim: Utilization of different modalities to control image generation

Challenges:

1) Using a modified diffusion model- Control Net Architecture on Stable Diffusion- to improve generation using image and text
2) Prompt modification to improve the image quality

Results:

Prompt: *" A house with a garden"*

Negative Prompts: *"lowres, bad anatomy, bad hands, cropped, worst quality, low quality, jpeg artifacts, watermark, blurry, out of frame"*

Quality Enhancement Prompts: *", high resolution, highly detailed, masterpiece, best quality, professional"*

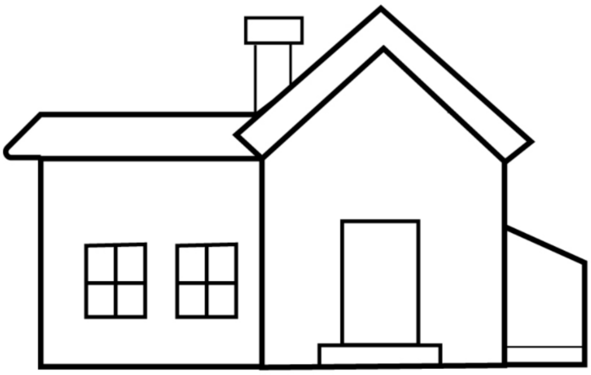


Figure 1- Reference "Sketch" passed as input

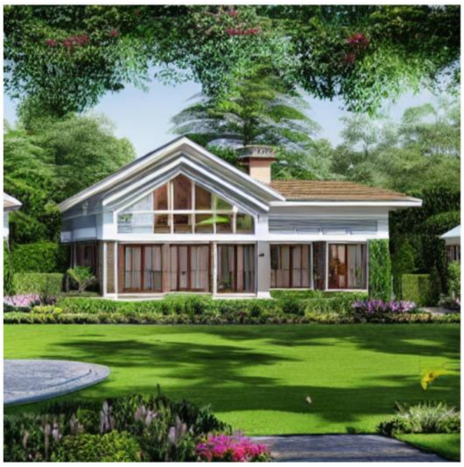
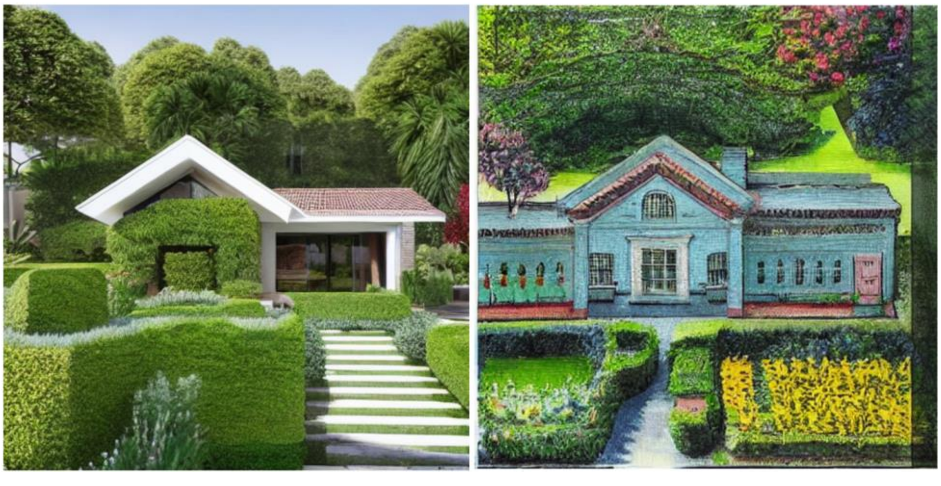


Figure 2- Generated Image

Multiple attempts: I tried multiple attempts but most of them failed (i.e some minors issues)



Conclusion: We can utilize this framework to generate completely random but structurally coherent images of human body, building, scenery and non-living objects by utilizing multi-modal information

Name: Irish Mehta
ASU ID: 1233414002

Task 9

Aim: Refining an image through iterations and CLIP similarity

Challenges:

1) Using CLIP similarity score to update gradients in latent space
2) Setting up an optimizer and tuning its learning rate

Results:

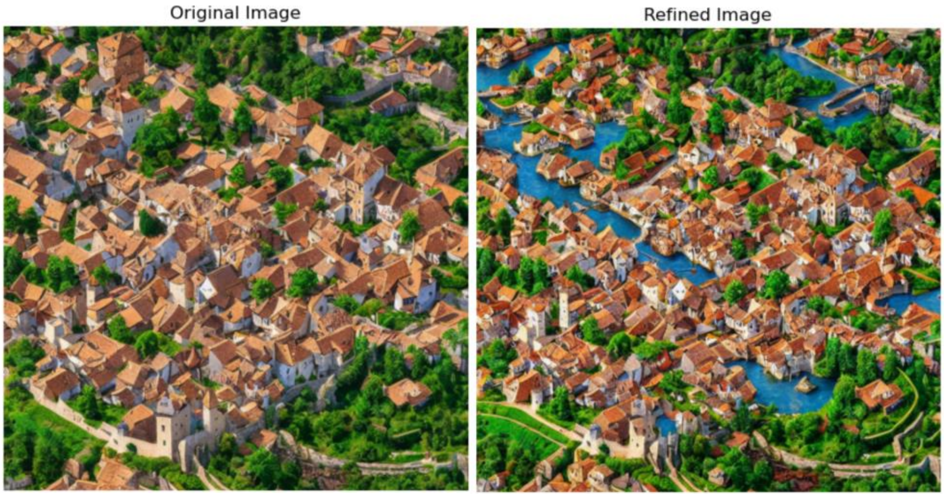
Prompt: *"A dragon flying over a medieval village"*

Negative Prompts: *"lowres, bad anatomy, bad hands, cropped, worst quality, low quality, jpeg artifacts, watermark, blurry, out of frame"*

Quality Enhancement Prompts: *", high resolution, highly detailed, masterpiece, best quality, professional"*

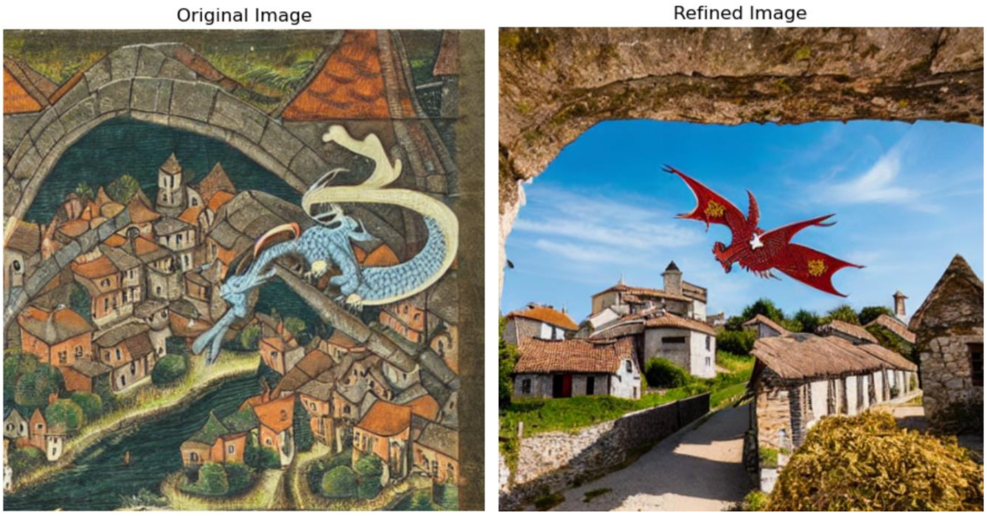
Tunable Parameters-

Attempt 1: With modified prompt quality, increased latent vector size and without finetuning over 25 iterations

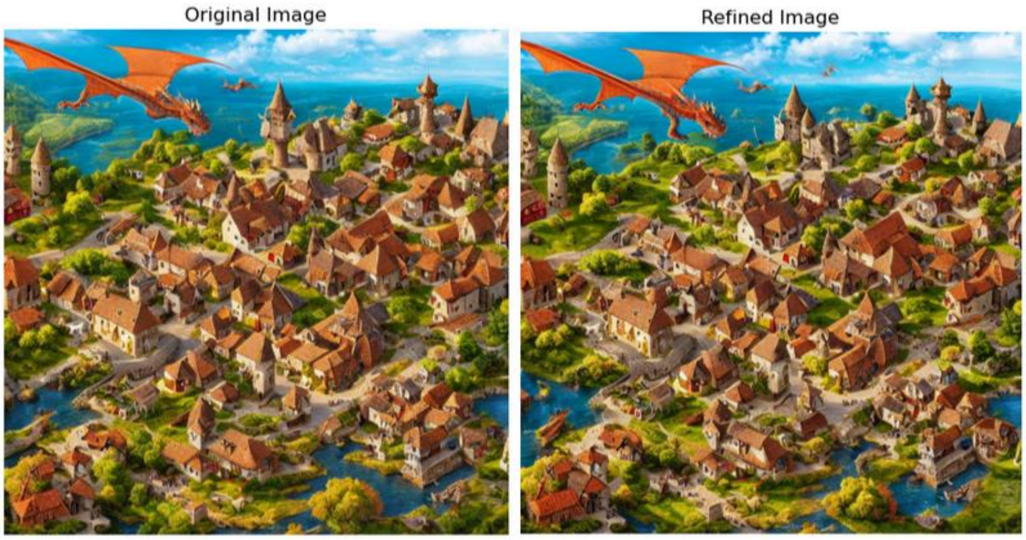


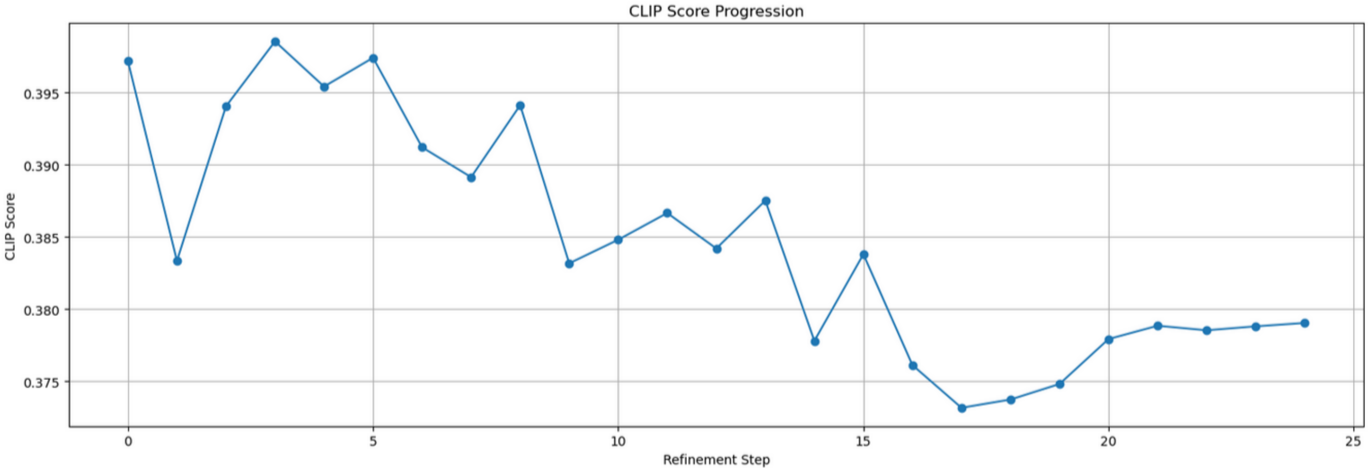
Insight- The model is not able to balance between quality and sticking to the prompt. I can see tiny parts of a dragon at different places, but I suspect that the model did not create a dragon because it could not generate a high-quality object as mentioned in the negative prompts

Attempt 2: Sticking to the original prompt with finetuning on guidance



Attempt 3: Trying the same approach as attempt 2 with different parameters





Conclusion: Given the complexity of the prompt and requirement of a creative approach, the fact that the model was able to create something even remotely close is really interesting. However, that being said, the generation quality is extremely random. I ran the code atleast 10 times, and only 2-3 times did I see a well formed “dragon”. I suspect it has a lot to do with the parameter tuning I chose, but even with experimentation, it didn’t change much

Part 3 Supplementary Questions

Q1- Can this approach (Controllable Image Generation) be extended to incorporate more modalities (e.g., color palettes, reference images)? What challenges might arise?

Ans: Yes, this approach can be extended to incorporate more modalities (including 3D models). To do this, we can condition the diffusion model on additional inputs like color palettes or reference images (like we added a sketch) so that the generation process can be guided more precisely. A color palette can control the image’s overall aesthetic, while reference images can enhance style and details, and 3D models can influence the spatial structure and depth of objects.

However, it is extremely challenging. Multiple modalities increase the model’s complexity and require major additions to the architecture to handle the multi-modal embeddings simultaneously. We would need to encode and align different types of data, such as text, sketches, colors, and images, and make sure that we are not over-complicating the data for the model to understand. This could lead to increased memory and computational costs, especially when working with high-dimensional inputs like 3D models. Moreover, ensuring the different modalities complement each other without overwhelming or confusing the model is crucial, as conflicting data can lead to incoherent outputs.