1. Understanding Diffusion

- **Explain what happens during the forward diffusion process, using your own words and referencing the visualization examples from your notebook.**

During the forward diffusion process, an image put as input is transformed into noise, adding more and more "Gaussian Noise". This can be seen from the diffusion process example in Step 4, which shows an example image at different levels of noise.

- **Why do we add noise gradually instead of all at once? How does this affect the learning process?**

Noise is added gradually instead of all at once to give the model a better understanding of the image at different levels. As more and more noise is added, it becomes further and harder to identify compared to the original but the model still knows what the image is. Amount of noise changes the learning process as the different levels noise is added, means the image has different ways it has to decipher and understand the image enough to "denoise" it again.

- **Look at the step-by-step visualization - at what point (approximately what percentage through the denoising process) can you first recognize the image? Does this vary by image?**

I'm able to recognize images around 60% usually, but it definitely does vary by image. Like in the example on step 4, I'm able to recognize the number eight all the way at 80% because it's just two solid colors, and a large recognizable number. The more details and objects in an image the harder it becomes to identify at different noise levels unless you're already familiar with it. At a certain point it just looks like random pixels and colors, but around 60% I can usually see enough detail to make out an image.

2. Model Architecture

- **Why is the U-Net architecture particularly well-suited for diffusion models? What advantages does it provide over simpler architectures?**

U-Net architecture is suited specifically for diffusion models because specifically of it's "encoder-decoder" structure. Allowing you to encode images for downsampling the image, and collect different features, then decode and reconstruct it. The skip connections directly connecting to the encoder/decoders also help keep specific details while working with noise, these features unavailable in other architectures that work.

- **What are skip connections and why are they important? Explain them in relations to our model**

  Like I mentioned in the last question, skip connections connect the layers/feature maps between the encoder and decoder and keep them at the same resolution. This helps recover extra details that downsampling might have caused, in relation to my model this means the images would come out sharper at the end.

- **Describe in detail how our model is conditioned to generate specific images. How does the class conditioning mechanism work?**

The model is conditioned to generate specific images based on specific features or "style" of different classes within the model. The class conditioning mechanism works by first taking class labels and capturing the features within them, a mask determining how much the model follows the class, and conditioning takes both time embeddings holding information about the noise, and class embeddings which are imprinted onto feature maps.

3. Training Analysis (20 points)

- **What does the loss value tell of your model tell us?**

Although I didn't get to directly test my model myself, I know the loss value would reflect how good it is at predicting noise at each step of the time embeddings. The lower loss value would mean it's accurately predicting how the noise changes an image, which also means the images would turn out sharper. But if it was a high loss value, the images would likely be blurry or missing details due to the model not knowing how to handle the noise.

- **How did the quality of your generated images change change throughout the training process?**

- **Why do we need the time embedding in diffusion models? How does it help the model understand where it is in the denoising process?**

Time embedding helps identify what noise level, and part of the denoising process it's in. With each time it has a different noise level assigned to it, allowing the diffusion model to understand what stage of progress it's in.

4. CLIP Evaluation (20 points)

- **What do the CLIP scores tell you about your generated images? Which images got the highest and lowest quality scores?**

CLIP scores basically just compare an image's caption to the generated image, telling it how well the features defined by the class meet the caption, and how "realistic" it is. The highest score images would be the clearest, sharpest, and most identifiable ones while the lower quality scores mean it would've been blurry and unreadable.

- **Develop a hypothesis explaining why certain images might be easier or harder for the model to generate convincingly.**

  I believe it's all to do with complexity, the more there is to work with in an image the more similarities can be found to other things. When there's only one subject focused in an image the AI is able to easily outline and identify how it changes with added/reduced noise

levels. But the smaller details, and more items in an image the harder it becomes for an AI to understand how they work or what's supposed to actually be there.

- **How could CLIP scores be used to improve the diffusion model's generation process? Propose a specific technique.**

I think CLIP could be used to improve the generation by giving it another check to see how accurate it is. If a model generates maybe a sample image that doesn't come close to the caption, it can get sent back through until a more defined and reasonable output is made. A specific technique for this could be implementing it as a measure to re-make an image at the end if it doesn't meet a specific threshold. This would eventually train the AI to know what makes a "bad" or "blurry" image compared to a clear and understandable one.

5. Practical Applications (20 points)

- **How could this type of model be useful in the real world?**

These types of models can easily be used in the real world for things like upscaling images to higher resolutions, quickly making concept arts or piece ideas for creators, or even easily editing things in images and filling in gaps.

- **What are the limitations of our current model?**

  Although I didn't really get to test it that much, the model seems slow and like it would struggle with complex images. Without specific data being loaded either, it wouldn't be able to properly go through class conditioning which means it takes only specific, simple and labeled data for trained images.

- **If you were to continue developing this project, what three specific improvements would you make and why?**

  Since I wasn't able to even fully finish the project, I'm not too sure what improvements the model could use since I haven't gotten to see it's results. Although from what I've understood about the model it's only able to take low-resolution images from specific class labels, leaving it very limited in it's capabilities. I think one of the first improvements that could be made was mentioned in the last section, with using CLIP scores to help train the model further and improve the image clarity and accuracy. Other than that all I can think about is improving the time embeddings, as one of the core parts of the model come from how well it's able to de-noise an image, so the timing and amount of noise is very important to get right.

Bonus Challenge (Extra 20 points)

Try one or more of these experiments:

1. **If you were to continue developing this project, what three specific improvements would you make and why?**

2. **Modify the U-Net architecture (e.g., add more layers, increase channel dimensions) and train the model. How do these changes affect training time and generation quality?**
3. **CLIP-Guided Selection: Generate 10 samples of each image, use CLIP to evaluate them, and select the top 3 highest-quality examples of each. Analyze patterns in what CLIP considers "high quality."**
4. **tyle Conditioning: Modify the conditioning mechanism to generate multiple styles of the same digit (e.g., slanted, thick, thin). Document your approach and results.**

Additional References:

machine learning - Why diffusion model always use U-Net? - Artificial Intelligence Stack Exchange

https://dzdata.medium.com/intro-to-diffusion-model-part-5-d0af8331871