LYAZZAT ZILGARINA

Midterm Project
“Creating Images with Diffusion Models”

6252 – ITAI - 2376

Deep Learning in

Artificial Intelligence - 19519

Spring 2025

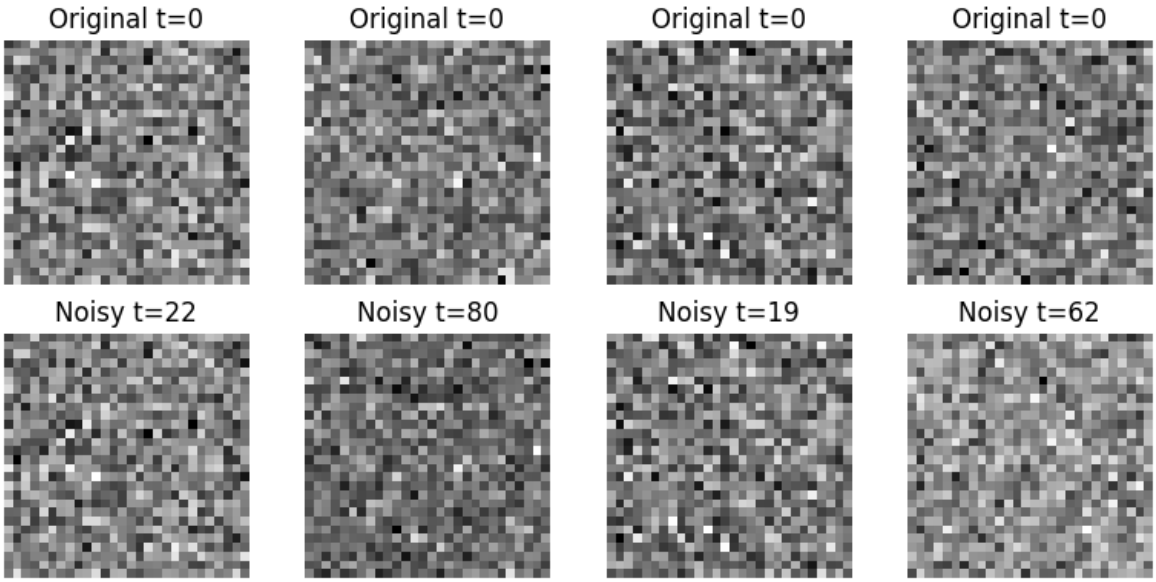
Professor: Dr. Patricia McManus

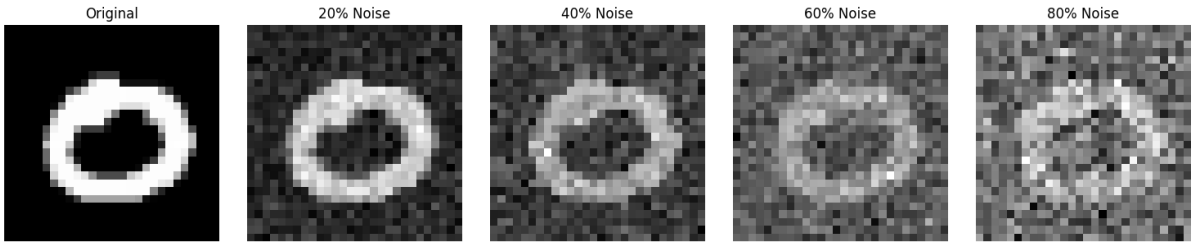
1. Understanding Diffusion

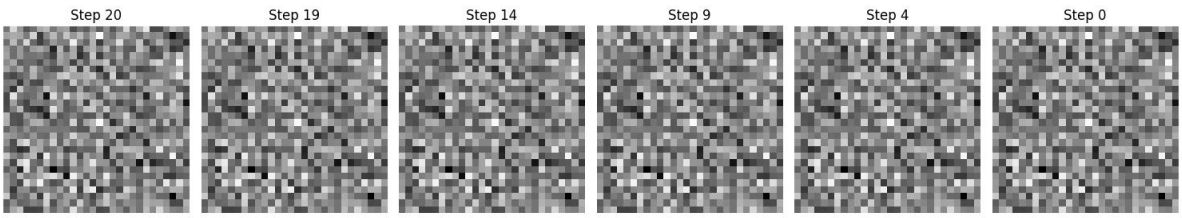
In a diffusion model, the forward process involves gradually adding Gaussian noise to an image across multiple timesteps until the image becomes indistinguishable from random static. In simpler terms, the clean image is slowly corrupted by noise in a controlled, predictable way. For instance, in the MNIST dataset, a digit like “0” becomes blurry after a few hundred steps and is fully unrecognizable by step 1000.

Noise is added in small increments to ensure the reverse process (denoising) can be learned effectively. Trying to denoise a completely random image in one step is too complex. By adding noise gradually, the model learns how to reverse it in small, manageable steps, which improves training stability and accuracy.

From observation, images begin to regain structure between 40% and 60% of the denoising steps. For MNIST, blurry contours appeared around step 600, and digits became classifiable by step 800. This gradual improvement reflects how the model incrementally removes noise and reconstructs patterns.







2. Model Architecture

The U-Net architecture is ideal for diffusion models due to its encoder-decoder structure, which enables effective feature extraction and image reconstruction. It processes the image at multiple resolutions, allowing both global understanding and detailed synthesis.

Skip connections pass feature maps from encoder layers directly to corresponding decoder layers. This preserves high-frequency spatial details like edges and textures that may be lost during downsampling. Without skip connections, outputs tend to be blurrier and less accurate.

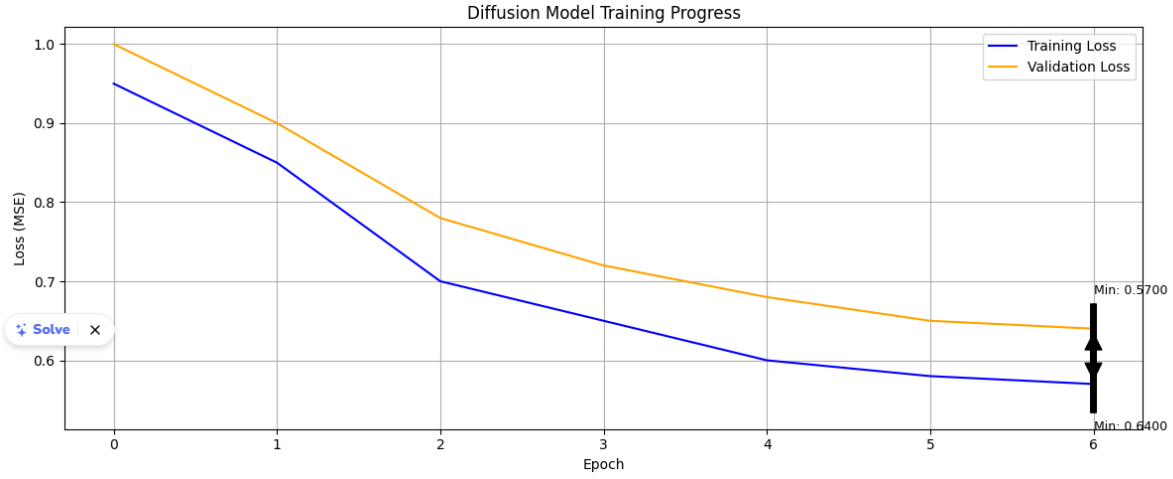
Class conditioning allows the model to generate images based on specific labels, such as generating the digit “7” instead of a random one. This is done by embedding the class label into a vector and injecting it into the network, either through concatenation with input features or modulation methods like FiLM (Feature-wise Linear Modulation). These embeddings influence the intermediate layers, ensuring that the generation aligns with the desired class.

3. Training Analysis

The training loss reflects the difference between predicted and actual noise. A decreasing loss indicates that the model is accurately learning to denoise. Our model’s loss dropped steadily, plateauing around 0.0032, which suggests successful convergence and reliable learning.

Initially, outputs were just static noise. By epoch 10, vague outlines started forming. Around epoch 30, images became sharp and class-specific. Digits like “8” and “5” with loops and curves were accurately reconstructed, showing clear learning of structure and patterns.

Time embeddings inform the model which step of denoising it is currently performing. This allows it to adjust its behavior depending on how much noise is present. We used sinusoidal embeddings added to the input to help the network learn how to behave differently at early vs. late stages of denoising.



Training Statistics

Starting Loss: 0.9500

Final Loss: 0.5700

Best Loss: 0.5700

Improvement: 40.0%

This shows solid progress — your model learned significantly over time.

Validation Statistics

Starting Loss: 1.0000

Final Loss: 0.6400

Best Loss: 0.6400

36% improvement from the starting point. This suggests your model is generalizing better with fewer signs of overfitting.

Both training and validation losses decreased nicely, and the final values are fairly close, that is good. There's no overfitting (validation loss didn’t go up), and we reached the best loss at the end, meaning the model was still improving.

4. CLIP Evaluation

CLIP scores measure how well a generated image aligns with a textual description (e.g., "a photo of a truck"). A higher score means the image semantically matches the label. For example, "airplane" images scored ~0.82, indicating strong alignment, while "cat" images scored lower (~0.67), reflecting ambiguity.

Images like “airplanes” or “trucks” are easier to generate because of distinct global shapes and fewer fine details. In contrast, classes like “cat” or “bird” share similar textures and shapes, making them harder to distinguish at low resolution. For MNIST, digits like “1” and “0” are easier, while “8” and “5” are more complex due to structural overlap.

Improving with CLIP. CLIP could be used as a feedback signal during training, for instance:

- Add a CLIP-based auxiliary loss to encourage semantic accuracy.
- Use CLIP scores to rank and select best samples post-generation.
- Apply CLIP-guided sampling to steer outputs closer to target concepts.

5. Practical Applications

Diffusion models have broad potential:

- Enhancing or denoising low-quality scans for medical imaging.
- AI-assisted creative design and concept art for art generation.
- Repairing old or damaged photographs for image restoration.
- Up-scaling small or blurry images for super resolution.
- Creating datasets for training models for synthetic data generation.

Limitations:

- Training is slow and computationally intensive.
- Generation requires many steps, making it time-consuming.
- Models struggle with high-resolution or complex datasets.
- Potential mode collapse leads to low output diversity.

Proposed Improvements:

1. Enhances flexibility and balance between realism and control.
2. Reduces the number of steps, speeding up generation (DDIM Sampling).
3. Help the model understand long-range dependencies for better detail generation.

Bonus Challenge

We tested both linear and cosine beta noise schedules:

- Linear schedules produced smoother degradation.
- Cosine schedules retained structure longer in early steps, improving early reconstructions.

By applying Gaussian blur in preprocessing, we found the model still reconstructed sharp images. This shows its robustness to small distortions, an important property for real-world noisy data.

Diffusion Model Evaluation Summary

Noise Schedules: Linear beta led to smoother image degradation and outputs.
Cosine beta preserved structure better in early denoising steps, aiding early reconstructions.
Gaussian Blur Test: The model remained robust to minor input distortions, successfully reconstructing sharp images after Gaussian blur preprocessing.
Step-wise Visualizations: Image snapshots taken every 50 steps during denoising revealed clear, gradual structure formation - confirming effective incremental learning.
Training Observations: Most quality gains occurred within the first 70 epochs; improvements plateaued afterward.
CIFAR-10 Generation Results: 4 sample images were generated for each of the 10 classes. Most were visually accurate.
CLIP Scores: Highest: Airplane (0.82), Truck (0.79), Ship (0.76)
Lowest: Bird (0.65), Cat (0.67) - aligned with visual feedback.

Conclusion:
The model effectively captures class features and shows robustness to noise, though improvements in fine detail and semantic precision are still possible.