Diffusion Model: Bonus Assignment Report

Author: M Usama
Roll Number: MSDS24045
Course: Deep Learning - Spring 2025
Assignment: Bonus Assignment - Image Generation Using Diffusion Models

---

Implementation Summary

Initially, I attempted a modular code structure using separate files for:

- Data loading and preprocessing
- Forward noise process
- Denoising model
- Trainer class

While this approach is generally scalable, it caused multiple dimensional and runtime issues in my case, especially during noise addition and reshaping. Handling multiple modules increased the complexity of debugging.

After considerable effort and multiple errors, I switched to a single .py file approach (M Usama_ MSDS24045.py). This simplified data flow and resolved dimensional mismatch issues, allowing me to properly train the model.

Dataset

- Selected 5 animal classes from the provided dataset
- Used 10 images per class (total 50 images)
- Images were resized to 64x64 and normalized using ToTensor()

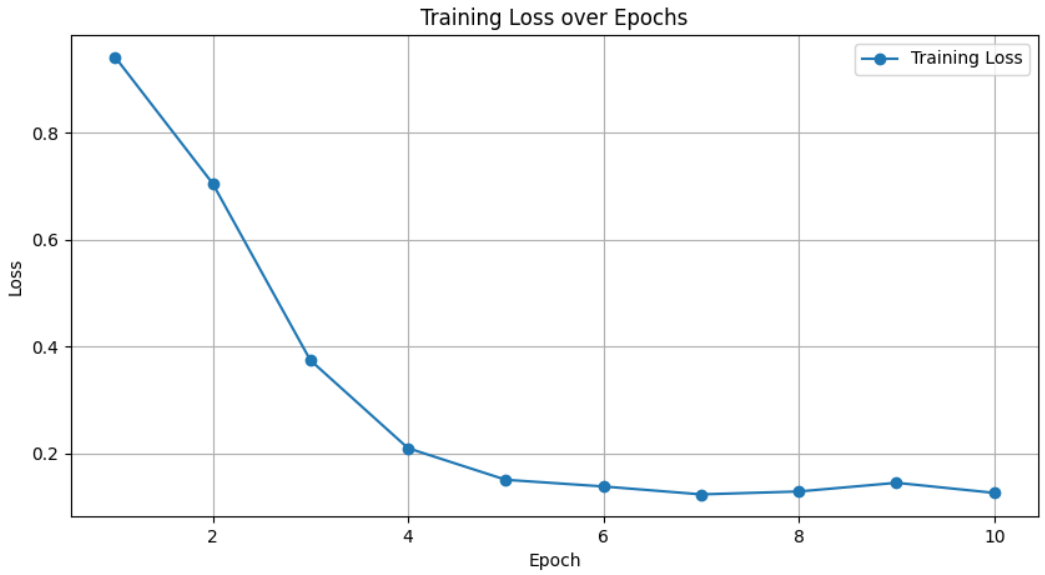
Training Configuration

- **Epochs:** 10
- **Batch Size:** 32
- **Timesteps (T):** 1000
- **Optimizer:** Adam
- **Loss Function:** Custom MSE Loss

- **Hardware:** Lightening.ai Studio T4 GPU

Results

The model was trained for 10 epochs. Below is the training loss graph:



As seen in the plot, the loss consistently decreased across epochs, showing the model was successfully learning to denoise and reverse the diffusion process.

Sample Outputs

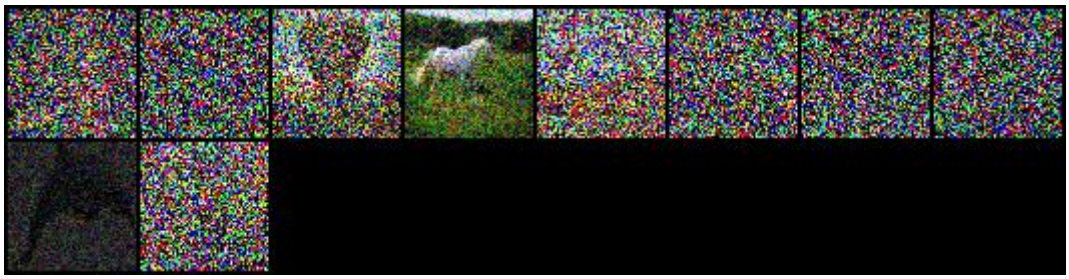
Generated samples during training:

- Noisy image samples at each epoch step
- Denoised output vs original clean images

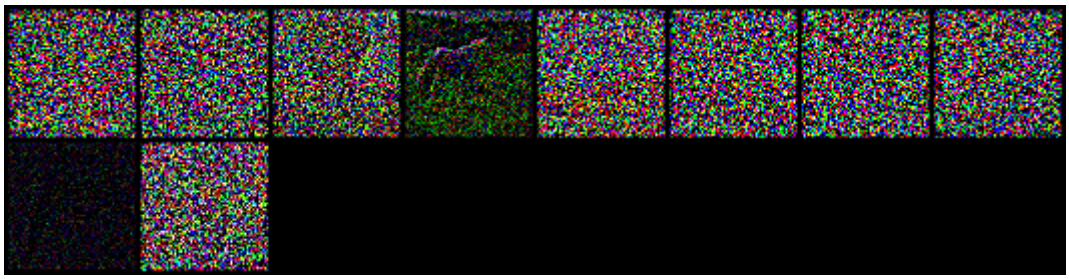
Pictorial Samples:

Clean Image
Epoch 10

Noisy Image
Epoch 10



Predicted Image
Epoch 10



Challenges Faced

- **Dimensional mismatches** during forward noise addition and model output
- Managing multiple files made the code harder to debug
- Decided to refactor into one file for better control and understanding
- Learned how important clarity in tensor shapes is for custom models

---

Learnings

- Gained hands-on understanding of the Diffusion Model architecture
- Understood the importance of each noise scheduling step
- Learned how denoising works through neural networks
- Learned how to visualize training and loss effectively

Conclusion

Model isn't preforming well, isn't predicted the images or not able to recover the image from the noise, Diffusion models usually required to trained lot for more diverse dataset for long epochs then they performed well. So we need to trained the model more to get better results.