Diffusion-Based Image Denoising

Generative AI – Mini Project

Group Members:
Saksham Wane(2411CS04)
Sourav Kumar (2411CS14)
Prantik Maji (2411CS16)
Arpit Kumar (2411CS23)

Problem Statement & Objective

- **Goal:** Apply a pretrained diffusion model (DDPM) to denoise synthetically noised images from CIFAR-10 and understand the image generation process across inference steps.
- **Motivation:**
  - Image denoising is crucial for low-light photography, medical imaging, and enhancing degraded visual data.
  - Diffusion models have emerged as powerful generative models capable of high-quality image synthesis.
- **Objectives:**
  - Add synthetic Gaussian noise to test images.
  - Iteratively denoise using a pretrained DDPM.
  - Visualize and evaluate the step-wise reconstruction.
  - Analyze output quality using metrics like PSNR and SSIM.

Background & Reference Paper

- **Denoising Diffusion Probabilistic Models (DDPMs):**
  - Generative models that learn to reverse a noising process.
  - Training involves learning to denoise step-wise from pure noise to a clean image.
- **Key Paper:**
  - Ho et al., 2020 — *"Denoising Diffusion Probabilistic Models"*.
  - Introduced DDPMs as an alternative to GANs, offering stable training and high-quality outputs.
- **Core Equation:**

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$$

- **Where:** ϵ_θ is the noise estimator network, typically a U-Net.

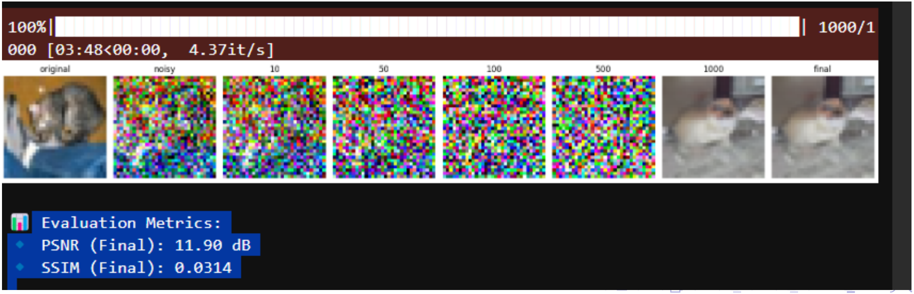
Dataset & Tools

- **Dataset: CIFAR-10**
  - 60,000 32x32 color images across 10 classes.
  - We use the **test set** (10,000 images) for our experiments.
  - Images were normalized and Gaussian noise ($\sigma = 0.2$) was added for denoising evaluation.
- **Tools and Libraries:**
  - **PyTorch:** Core deep learning framework for loading and preprocessing CIFAR-10.
  - **HuggingFace Diffusers:** Provided pretrained DDPM and scheduler for reverse diffusion.
  - **scikit-image:** Used to compute PSNR and SSIM for evaluation.
  - **Matplotlib:** Visualization of denoising steps and timelines.

Methodology

- **Approach:**
  - Used a pretrained DDPM (UNet2DModel from HuggingFace) trained on CIFAR-10.
  - Synthetic Gaussian noise was added to CIFAR-10 test images.
  - The pretrained DDPM was applied in reverse (denoising) mode over 1000 time steps.
- **Implementation Pipeline:**
  1. Load original CIFAR-10 test image and normalize.
  2. Add Gaussian noise with standard deviation $= 0.2$.
  3. Initialize DDPM scheduler and model.
  4. Perform denoising over 1000 steps, saving snapshots at step 10, 50, 100, 500, and 1000.
  5. Calculate final PSNR and SSIM.
- **Key Tools:** PyTorch, HuggingFace Diffusers, Matplotlib, scikit-image.

- **Visual Timeline:** Captured denoised image at intervals to show progression.
- **Selected Steps:** Step 10, 50, 100, 500, 1000, and final output.
- **Quantitative Evaluation:**
  - PSNR (Final): **11.90 dB**
  - SSIM (Final): **0.0314**
- **Findings:**
  - Minor structural improvements are noticeable over time.
  - Noise reduction is visible, but finer details are not fully restored.



Analysis & Insights

- **Progressive Restoration:**
  - The denoising process clearly reduces noise, especially after 100 steps.
  - Major improvements observed around step 500, indicating non-linear convergence.
- **Quality Metrics Interpretation:**
  - Low PSNR suggests limited pixel-wise accuracy.
  - Very low SSIM reveals poor structural similarity to the original image.
- **Possible Causes:**
  - Pretrained DDPM was not fine-tuned for higher noise levels or specific image details.
  - Lack of conditioning (e.g., class labels or guidance) affects denoising specificity.
- **Conclusion from Analysis:**
  - DDPM shows strong potential but requires contextual guidance or domain-specific tuning for high-fidelity results.

Challenges & Limitations

- **Model Limitations:**
  - The pretrained DDPM is not optimized for arbitrary noise levels.
  - Absence of class-conditioning limits recovery of semantic structures.
- **Performance Bottlenecks:**
  - Full 1000-step inference is computationally expensive.
  - Real-time or interactive applications are not feasible in current setup.
- **Evaluation Constraints:**
  - PSNR and SSIM do not capture perceptual quality fully.
  - Metrics may undervalue visually decent reconstructions with minor spatial shifts.
- **Improvements for Future:**
  - Explore classifier-guided or diffusion transformers.
  - Apply training-time augmentation or fine-tuning on noisy image sets.

Conclusion & Takeaways

- **Summary:**
  - Implemented DDPM-based image denoising using a pretrained model.
  - Visual and quantitative evaluation showed noise reduction but limited restoration fidelity.
- **What We Learned:**
  - Diffusion models effectively smooth and denoise images over multiple iterations.
  - Model architecture and conditioning play a critical role in output quality.
- **Future Scope:**
  - Experiment with conditional DDPM or control-net guidance.
  - Use real-world noisy datasets for evaluation and training.
  - Compare against GAN-based or hybrid denoising pipelines.

References

- Ho et al. (2020). *Denoising Diffusion Probabilistic Models*.
- CIFAR-10: `https://www.cs.toronto.edu/~kriz/cifar.html`
- Diffusers Library: `https://huggingface.co/docs/diffusers`
- Code Repo `https://github.com/VaruosKumar/GenAI-Mini-Project`