1. Mini Project: Diffusion Models in Computer Vision

Cascaded Shallow model

Desirèe Charles
Art Ó Liathain

December 2025

2. Table of Contents

Contents

List of Figures

3. Introduction

Recent advances in generative modeling have shown that high-quality image synthesis can be achieved not only through large, monolithic architectures, but also through the composition of multiple simpler models. Cascaded generative approaches decompose the image generation task into a sequence of stages, where each stage is responsible for adding structure, resolution, or detail. By progressively refining an image rather than generating it in a single step, these models can reduce training complexity, improve interpretability, and allow individual components to be analyzed or modified independently.

Each stage serves a different purpose, the first is a generation model to generate a coarse 16x16 resolution image. The second stage is a learned upsampler that upsamples the 16x16 coarse output images to 32x32. The last stage is a refiner which refines the output of the learned upsampler using perception loss to an accurate 32x32 representation. These models cascade from one to another to approximate a CIFAR-10 image while each retaining a shallow number of layers favouring wide by shallow models over the traditional deep and narrow of current state of the art.

The goals of this paper are:

- To implement and train a working cascading image generator on CIFAR-10
- To implement a wide variety of models to compare in the different stages
- To compare results across model size, training schedule and sample quality, using Kernel Inception Distance (KID) and qualitative visual inspection

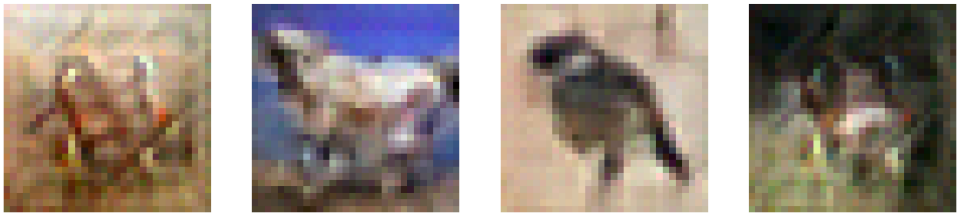

Figure 1: Generated samples.

Figure 2: Target samples.

4. Background and Literature Review

Within this project a wide variety of models have been selected due to their prominence in state of the art computer vision. This paper assumes a base understanding of computer vision models as the focus of the paper is cascading image generation.

4.1. Generation models

4.1.1. Denoising Autoencoders (DAE)

Denoising Autoencoders learn to reconstruct clean data from corrupted inputs, forcing the model to capture meaningful structure in the data distribution rather than simply memorizing inputs [1]. By training on progressively noisier samples, DAEs learn a mapping that moves samples toward higher-density regions of the data manifold. This denoising objective forms the conceptual foundation for diffusion-based generative models, where generation is performed through iterative noise removal.

4.1.2. Diffusion Models

Diffusion models define a generative process as the reversal of a gradual noising procedure applied to training data [2]. During training, noise is incrementally added to data over a sequence of timesteps, while the model learns to predict and remove this noise. The general architecture used for diffusion models are Unets [3] which are a natural choice to map corrupted data to reverse process parameters. At inference time, the model generates samples by starting from pure noise and iteratively denoising, producing high-quality and diverse images with stable training dynamics.

4.1.3. ResNet

Residual Networks introduce skip connections that allow information to bypass intermediate layers, mitigating issues such as vanishing gradients in deep neural networks. By learning residual functions rather than direct mappings, ResNets enable more stable optimization and improved feature propagation [4]. These properties make them well-suited for generative models, where preserving low-level spatial information across layers is critical.

4.2. Upsampler Models

4.2.1. PixelShuffle

PixelShuffle is an efficient upsampling operation that rearranges channel information into spatial resolution, increasing image size without introducing checkerboard artifacts common in transposed convolutions [5]. This approach enables computationally efficient super-resolution while maintaining spatial consistency in generated images.

4.3. Refiner Models

4.3.1. Vision Transformers

Vision Transformers (ViTs) model images as sequences of patches and use self-attention to capture long-range dependencies across the entire image [6]. This global receptive field allows refiners to reason about spatial relationships and semantic consistency, making ViTs particularly effective for correcting structural inconsistencies and improving perceptual coherence.

4.4. Cascaded diffusion models

Cascaded generative models approach image synthesis by breaking the generation process into multiple sequential stages, each operating at an increasing level of resolution and detail [7]. Prior work has shown that generating images in this progressive manner simplifies the learning problem, as early stages focus on global structure while later stages specialize in

adding fine-grained details. In the context of diffusion models, cascades typically consist of an initial low-resolution generator followed by learned super-resolution models that condition on the outputs of previous stages. More broadly, cascaded approaches highlight how complex image distributions can be approximated through a composition of simpler models rather than a single, highly complex network, providing both practical performance benefits and increased interpretability.

This is an approach with merit but is prone to issues such as error propagation throughout the pipeline. There are many ways in which this has been approached, teacher forcing, conditional augmentation [7] to name a few are approaches used to manage the loss carrying from one stage to the next during training. While results have been shown for cascaded models there is much room for growth and exploration such as alternative models that only diffusion which would allow more diverse methods of training.

5. Method

5.1. Dataset and task

Our project uses the CIFAR-10 dataset, which consists of 60,000 training images of size 32x32 across 10 object categories. In this project, the labels are ignored, and the task is image generation. Images are loaded and converted to tensors in (0,1). No significant augmentation is applied, as the quality of generative images is the primary concern rather than classification performance.

CIFAR-10 was the chosen dataset as it allows quick generation due to the 32x32 size, it is also a heavily tested framework which allows for easier comparisons with the current state of the art.

5.2. Cascaded Model Training Schedule

The generator is implemented as a three stage pipeline, which base input is a 16x16 noisy image:

- The generation stage: 16x16 coarse image
- The upsampler stage: 32x32 upsampled image
- The refiner stage: 32x32 refined image

All stages inherit from Baseclass, which enforces they implement preprocess() and forward(). These functions are essential to be able to properly handle the different models as each model handles inputs and training differently.This abstraction makes it possible to easily switch the different underlying models (like changing from a simple convolutional refiner to ViT refiner).

5.3. Training

The training has three stages:

5.3.1. Stage 1: Coarse Downsampled Image Generation

The first stage of training is the “warm up” stage for the coarse model. This is required as the following refiner and upsampler layers are based on the output and the early learning outputs would propagate errors throughout the pipeline. The loss function for the generation stage is MSE.

5.3.2. Stage 2: Upsampler and Refiner

In stage 2, coarse generation stage is frozen and the upsampler and refiner models are trained. Due to the unstable nature of the generation stage a teacher forcing schedule is used. This decides if the refiner and upsampler get a ground truth downsampled image or the coarse stage output. The scheduler for this lowers the possibility of giving the ground truth to the upsampler linearly over the number of stage 2 epochs. The start and end points are configurable. The losses in this phase include: an MSE loss for the upsampler output compared to a resized ground truth image, a refiner loss that is comprised of MSE and a perception loss.

5.3.3. Stage 3: Fine Tuning

During the last stage, all three stages get unfrozen and are trained together with the same loss combinations. In this stage the gradients are not detached from the coarse stage allowing the gradients to flow from the upsampler and refiner to the coarse model to allow for more fine tuned training of the weights to work in tandem with the cascading models.

5.3.4. Stage Information

Each phase is assigned its optimizer and learning rate scheduler, the settings are made through a configuration. The Config class is responsible for defining the core hyperparameters, the number of layers in the coarse block, the dimensions and heights of the upsampler and refiner, and the teacher forcing start and end probabilities. This configuration was designed with ablations studies in mind to allow for comprehensive testing and comparison.

6. Experimental analysis and evaluation

6.1. Baseline configuration

The baseline configuration for all empirical tests is:

Category	Configuration
Dataset	**Batch size:** 128
Diffusion Parameters	**Timesteps (T):** 25 \ **Beta start:** 1e-4 \ **Beta end:** 0.02 \ **Timestep embedding dim:** 32 **Time embedding:** 128
Coarse Generator	**Resolution:** 16 × 16 \ **Model type:** Diffusion \ **Input channels:** 3 \ **Output channels:** 128 \ **Latent dimension:** 256 \ **Layers:** 6 \ **Skip connection every:** 2 layers \ **DAE sigma:** 0.5 \
Upsampler	**Model type:** PixelShuffle \ **Input channels:** 3 \ **Hidden channels:** 256 \ **Upscale factor:** 2 \ **Residual blocks:** 2
Refiner	**Model type:** Vision Transformer (ViT) \ **Input channels:** 3 \ **Feature channels:** 256 \ **Residual blocks:** 2
Training Schedule	**Coarse epochs:** 35 \ **Upsampler + Refiner epochs:** 35 \ **Joint training epochs:** 30
Optimization	**Learning rate (all stages):** 2e-4 \ **LR step size:** 15 \ **LR decay (gamma):** 0.5
Teacher Forcing	**Phase 2:** 0.9 ⟶ 0.1 \ **Phase 3:** 0.1 ⟶ 0.0

These were selected based on earlier tests and served as a baseline for all future tests

6.2. Ablation studies

The goal of the ablation study was to identify if there was a correlation between parameter count and performance and what combination of cascading models produce the best results.

To conduct fair tests the only features that were altered between tests were, the models used in the cascading Ex from diffusion -> resnet -> vit to dae -> pixel shuffle -> vit. These were

selected to allow a diverse range of models to be tested and to explore the correlation between parameter count and models between different cascades.

6.3. Evaluation metrics

6.3.1. Kernel Inception Distance (KID)

KID quantifies the correspondence of the generated images to the real CIFAR-10 images in the Inception network's feature space by means of their distributions. A lower value of KID indicates that the generated samples are, on average, even closer to the real image, as far as the visual features learnt are concerned. In our case, KID is calculated regularly throughout the training period by using a constant number of real and generated samples, and we present both the final and mean KID over the recorded assessments.

6.3.2. LPIPS (perceptual distance)

LPIPS uses deep features (VGG) instead of raw pixel differences to evaluate the similarity between the two images. It uses a pretrained model to calculate the perception loss. This is useful because MSE favours average values but that leads to heavy blur and homogeneous colours over sharp edged. In our processing, LPIPS is incorporated into the refiner loss alongside MSE, encouraging outputs that are not only numerically close to the target but also visually coherent.

6.3.3. Time per epoch

The duration of the epoch is recorded to capture computational cost and training efficiency. This helps put improvements in KID and visual quality into perspective against training time, which is important when model size or diffusion schedule is varied across configurations.

6.4. Results

Add tables

7. Conclusion

In light of the results it is clear that there is no strong bearing on the types of models used in sequence to generate images. This suggests that although there have been methods that effectively use cascading models to great effect [7], there is a need for specific integration methods between models rather than an abstract stage based approach. When each stage is defined abstractly, there can only be a weak abstract connection between each model reducing the effectiveness of the approach. This paper highlights shows that while there is merit in cascading models, there is a need for care when selecting each models to ensure they connect and flow well from one to another sharing gradients to allow for more accurate training.

Overall, this work shows that cascaded generative models can serve as a viable and flexible framework for image synthesis, but their effectiveness depends on thoughtful architectural and training choices. Future improvements are likely to come from tighter stage integration and more principled design rather than simply increasing model complexity.

8. References

Bibliography

[1] Y. Bengio, L. Yao, G. Alain, and P. Vincent, “Generalized Denoising Auto-Encoders as Generative Models.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/1305.6663

[2] J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/2006.11239

[3] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/1701.05517

[4] A. Thakur, H. Chauhan, and N. Gupta, “Efficient ResNets: Residual Network Design.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/2306.12100

[5] O. Zamzam, “PixelShuffler: A Simple Image Translation Through Pixel Rearrangement.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/2410.03021

[6] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/2010.11929

[7] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded Diffusion Models for High Fidelity Image Generation.” Accessed: Dec. 14, 2025. [Online]. Available: http://arxiv.org/abs/2106.15282