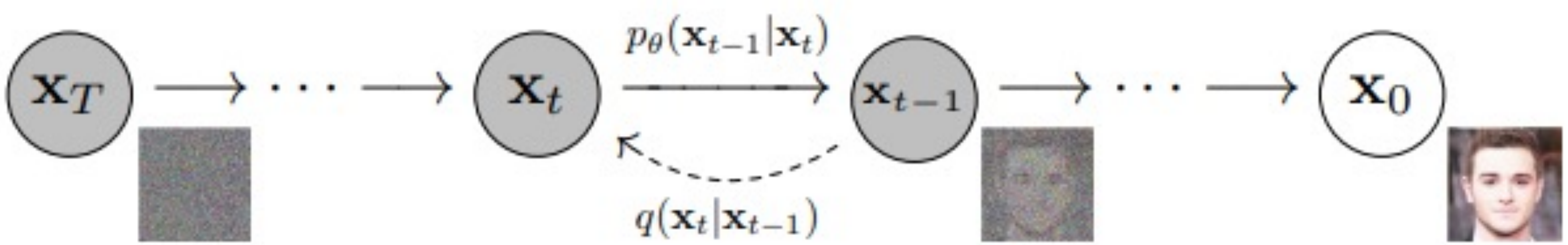
Image Generation with Diffusion Models

Lambda

Márk Somorjai *Zsombor Szommer* *Csanád Telbisz*

Denoising Diffusion Probabilistic Model

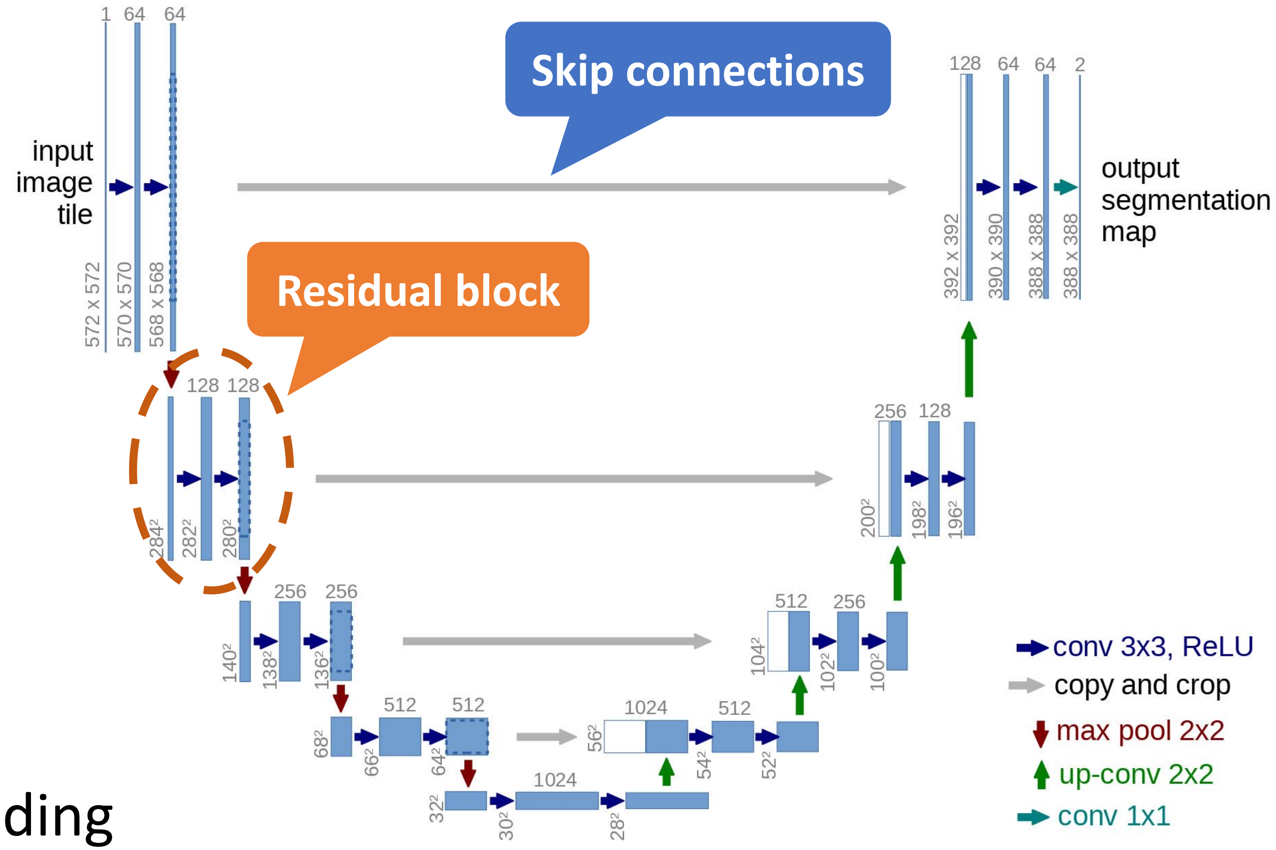
- Adding (Gaussian) noise to image
- Neural network: learns to predict noise at each small step
- Image generation: start from pure noise -> denoise step-by-step



Related Work & Sources:
[1] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. URL https://arxiv.org/abs/2006.11239
[2] N. Rogge and K. Rasul. The annotated diffusion model, 2022. URL https://huggingface.co/blog/annotated-diffusion
[3] A. Béres. Denoising diffusion implicit models, 2022. URL https://keras.io/examples/generative/ddim/
[4] A. K. Nain. Denoising diffusion probabilistic model, 2022. URL https://keras.io/examples/generative/ddpm/

Neural Network

- Input:
  - noisy image
  - noise level
- Output:
  predicted noise at the level
- Architecture: U-NET
- Optimizations:
  - Multiple residual blocks per level
  - Group normalization
  - Sinusoidal time (noise level) embedding
  - Exponential moving average for weight update



Source: O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015. URL http://arxiv.org/abs/1505.04597

Datasets

- Oxford 102 Flower Dataset[1]
  - 8189 images of various kinds of flowers

- Oxford-IIIT Pet Dataset[2]
  - 7349 images of cats and dogs

- Scaled down and cropped to 64*64 pixels
- 80%/20% split used for training/validation sets

[1] https://www.robots.ox.ac.uk/~vgg/data/flowers/102/
[2] https://www.robots.ox.ac.uk/~vgg/data/pets/

Evaluation

- Solution written in Jupyter Notebook, wrapped in Docker container
- Training
  - GTX 1070
  - 1000 epochs

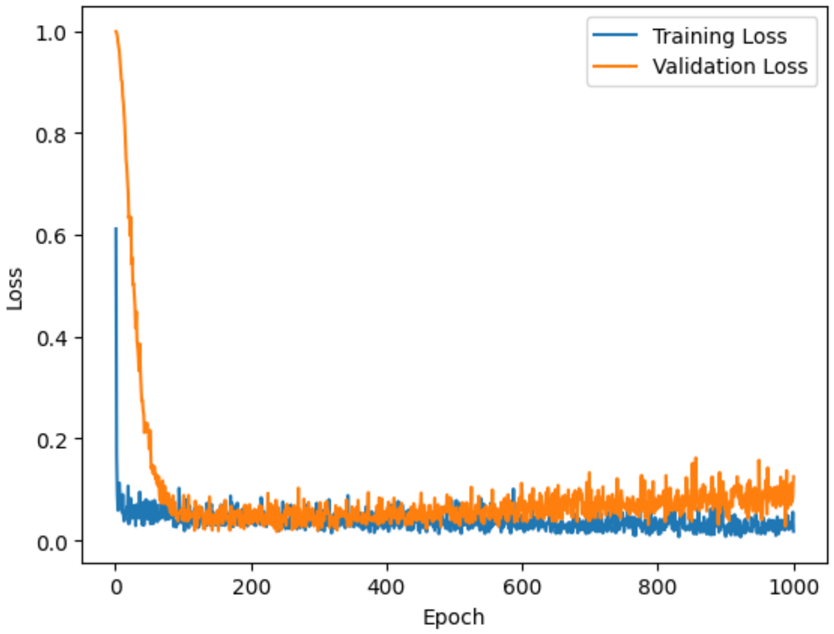
→ Smaller, overfit training: 6-8 hours
Training on full dataset: 36 hours

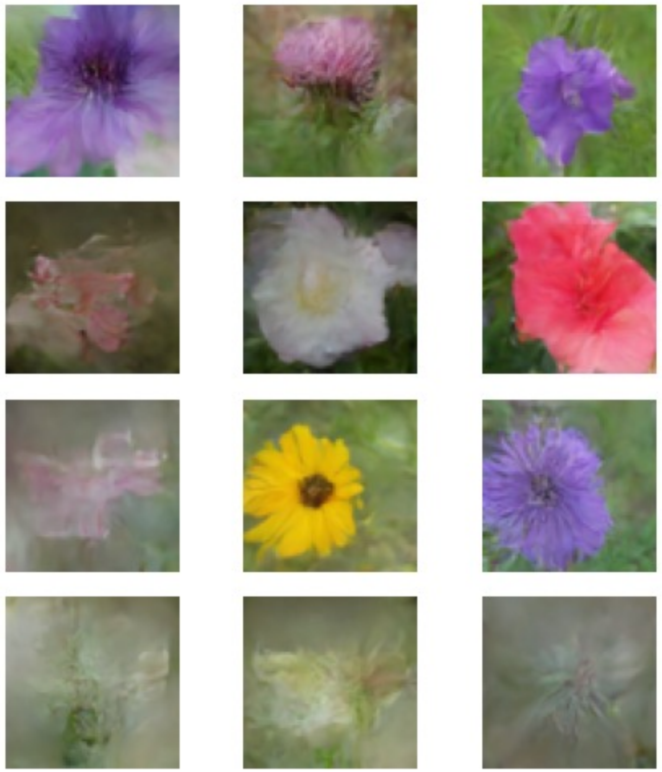
- Measured metrics:
  - training loss
  - validation loss
  - Kernel Inception Distance[1] (KID)
- Subjective evaluation: how *flower-like* are the generated images?

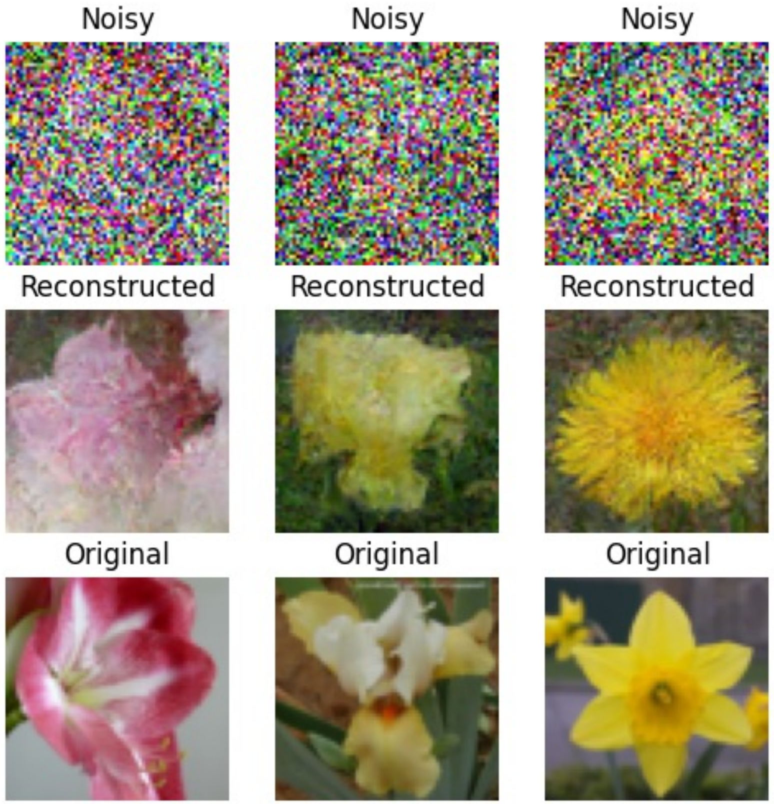
[1] M. Bińkowski, D. J. Sutherland, M. Arbel, A. Gretton: Demistifying MMD GANs (https://arxiv.org/abs/1801.01401)

Overfitting the Model

- smaller dataset: 1000-1000 training and validation images





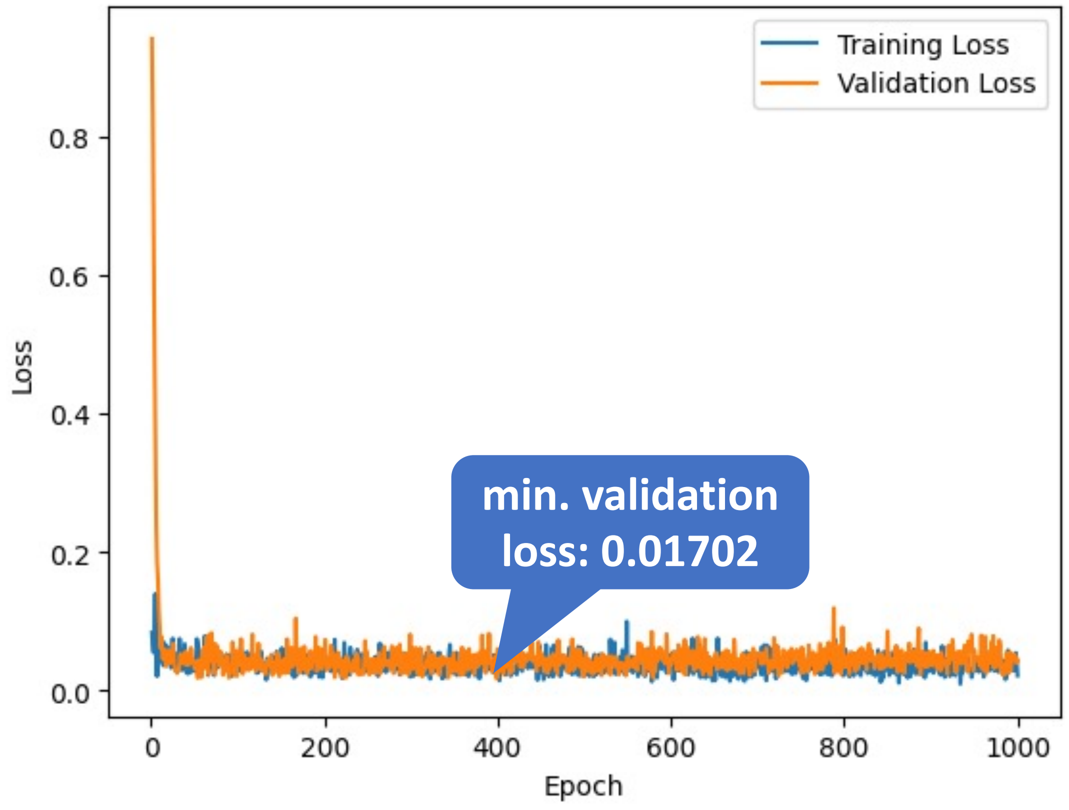


Loss **Generation** **Reconstruction**

Training on Full Dataset

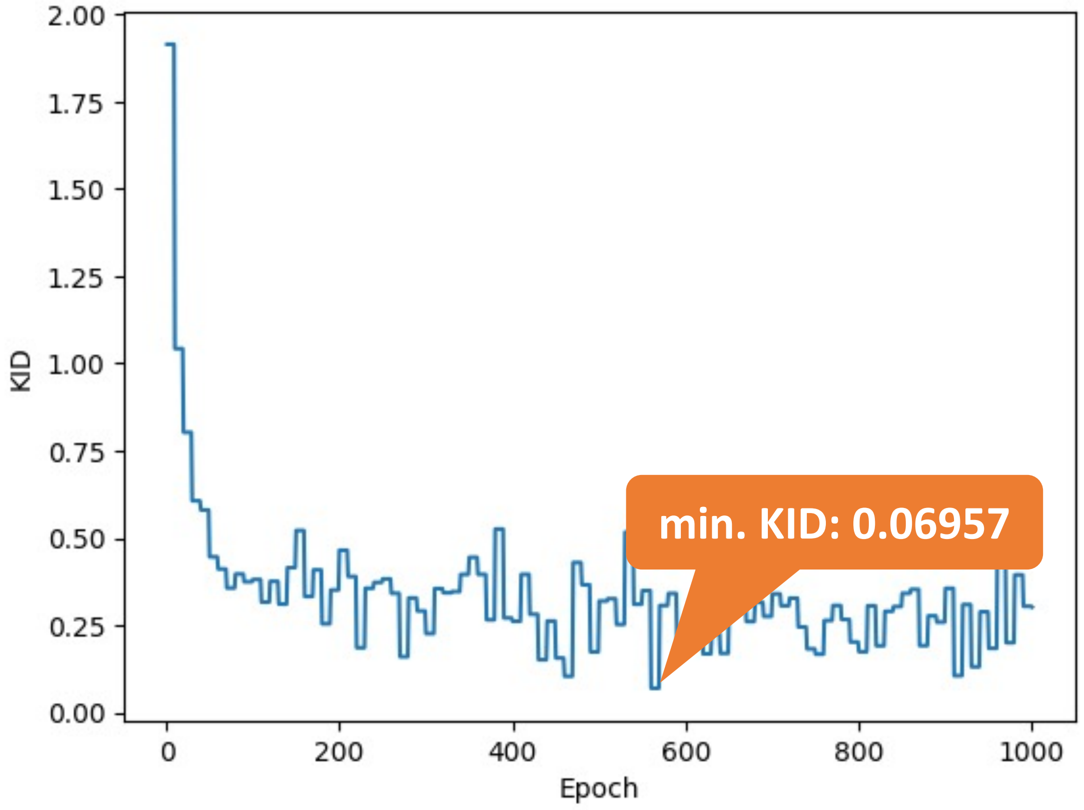
Loss

measured every epoch

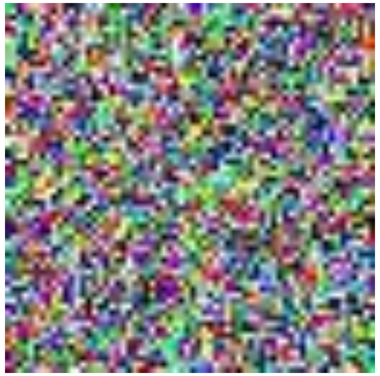


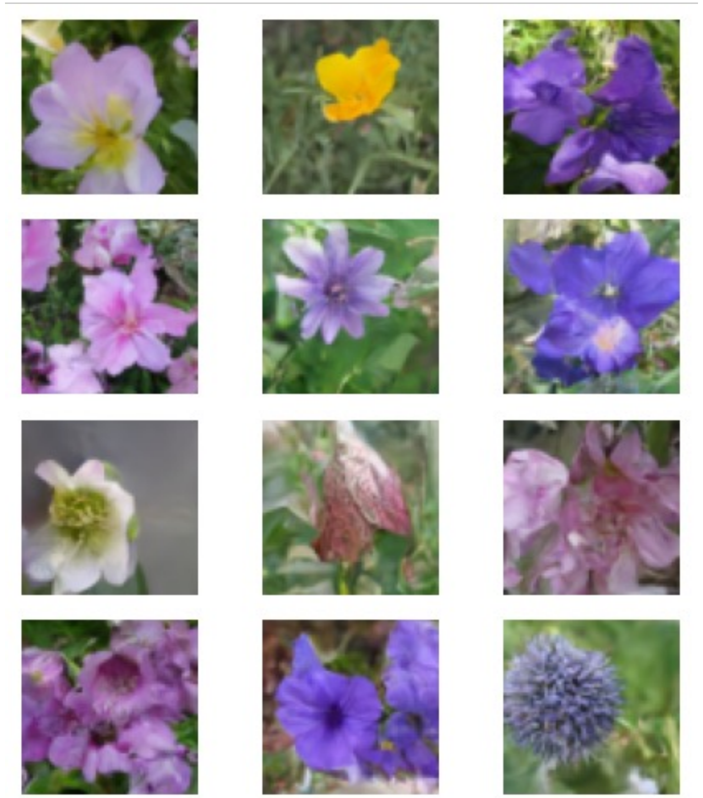
Kernel Inception Distance

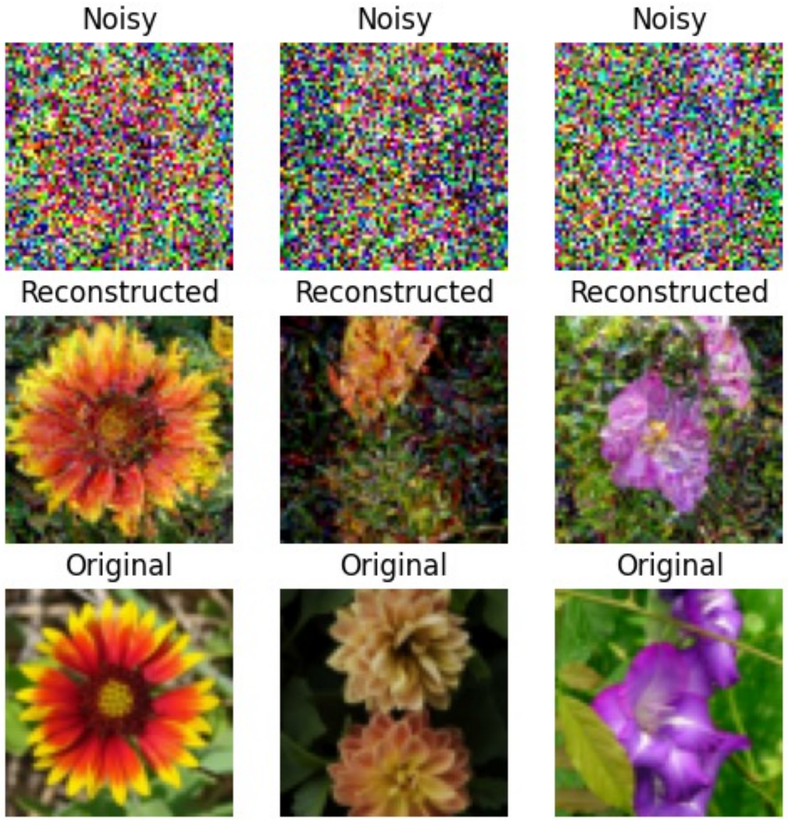
measured every 10 epochs



Best KID Model





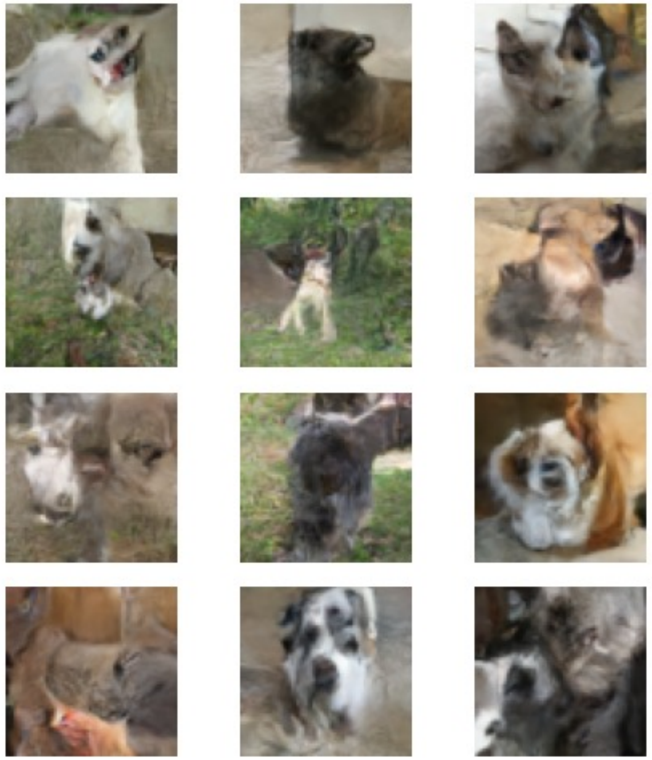


Generation

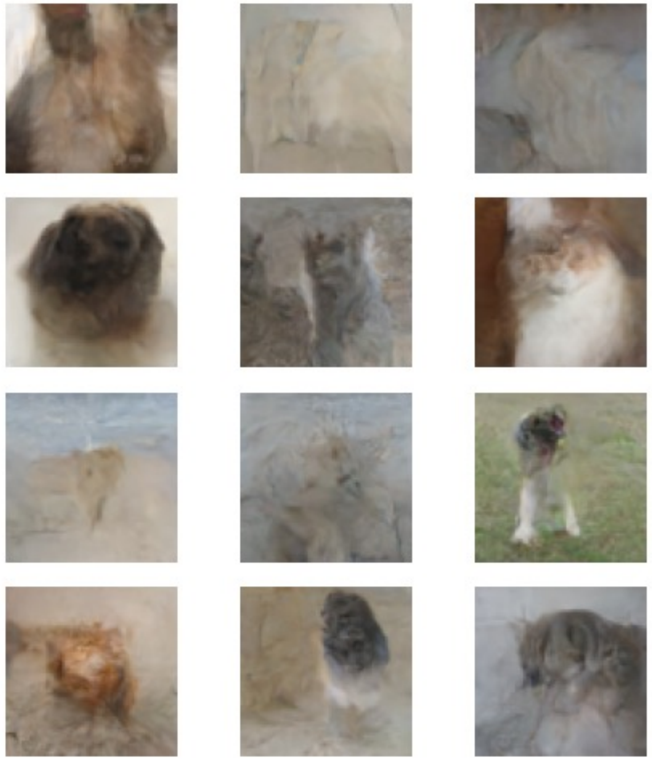
Reconstruction

Generated Pets

- smaller dataset: 1000-1000 training and validation images



Best KID Model



Overfit Model

Summary

- Familiarized ourselves with diffusion models
- Implemented denoising diffusion process and U-Net
- Trained the model on 2 datasets
- Evaluated the models based on 2 metrics: loss and KID
- Containerized the solution in Docker
- Created flower generating Gradio demo[1]

- Lessons learned: output activation function, group normalization
- Future work: further improve net (attention), complex noise schedule

[1] https://huggingface.co/spaces/Melidon/flower-generation