Zurich University
of Applied Sciences

School of
Management and Law

Machine Learning II
Generative AI

Dr. Elena Gavagnin
elena.gavagnin@zhaw.ch

FS2025

Machine Learning II – Images

Plan:

1. Artificial Neural Networks – Introduction, implementation with TF/Keras and Training

2. ”Classic” Deep Learning & Images – Convolutional Neural Networks and Transfer Learning

3. Multimodal and generative AI

«Multimodal and generative AI (W10 and W11):

1. Extracting Infos from Images: Object Detection
2. Multimodal AI: Connecting Anything and Images (CLIP, OWL-VIT)
3. Foundational Models in CV
3. Generative AI: from Autoencoder to Diffusion Models (and past that)

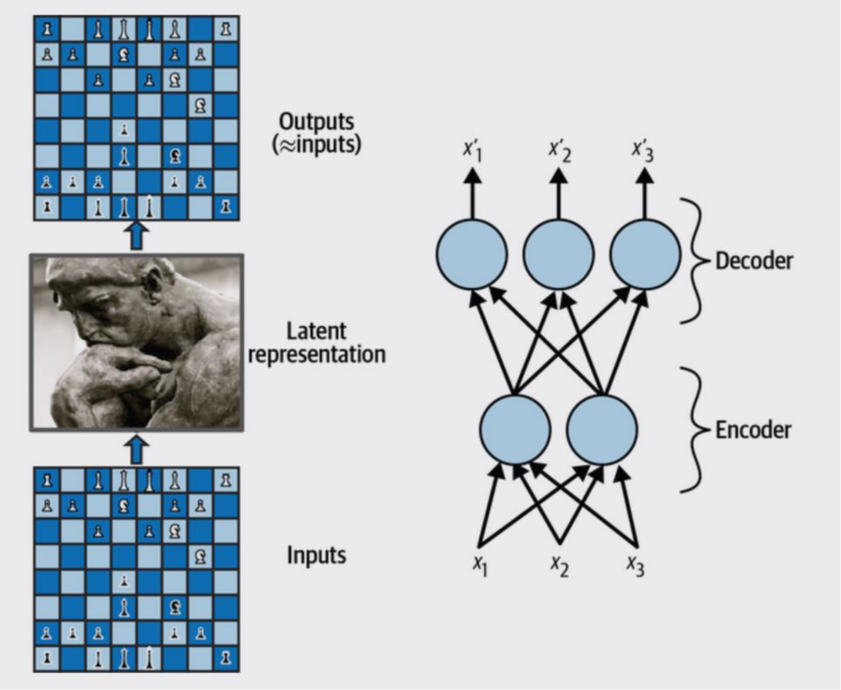
Generative AI: from Autoencoders to Diffusion Models

Efficient Data Representations

Which of the following number sequences do you find the easiest to memorize?

- 40, 27, 25, 36, 81, 57, 10, 73, 19, 68

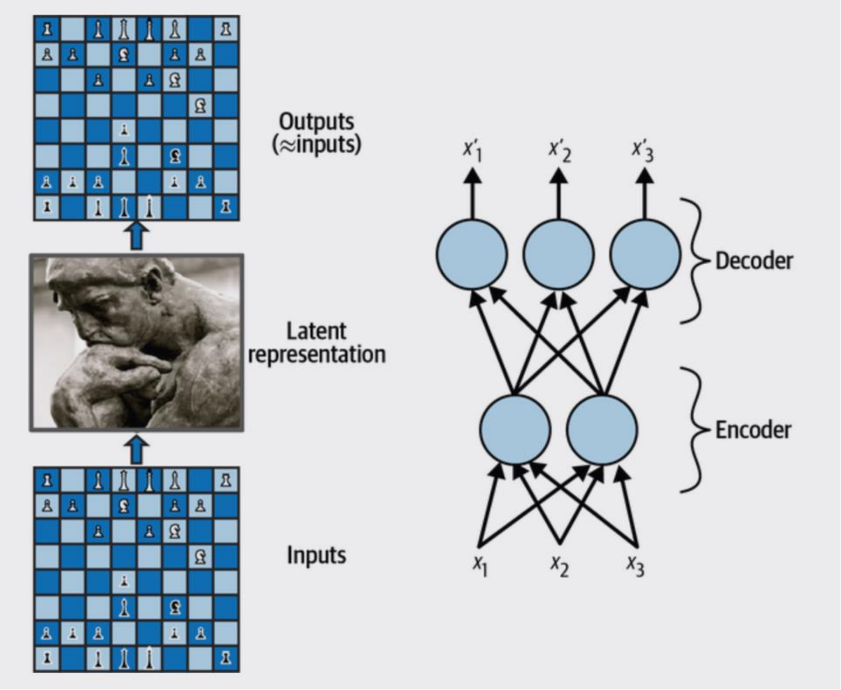
- 50, 48, 46, 44, 42, 40, 38, 36, 34, 32, 30, 28, 26, 24, 22, 20, 18, 16, 14



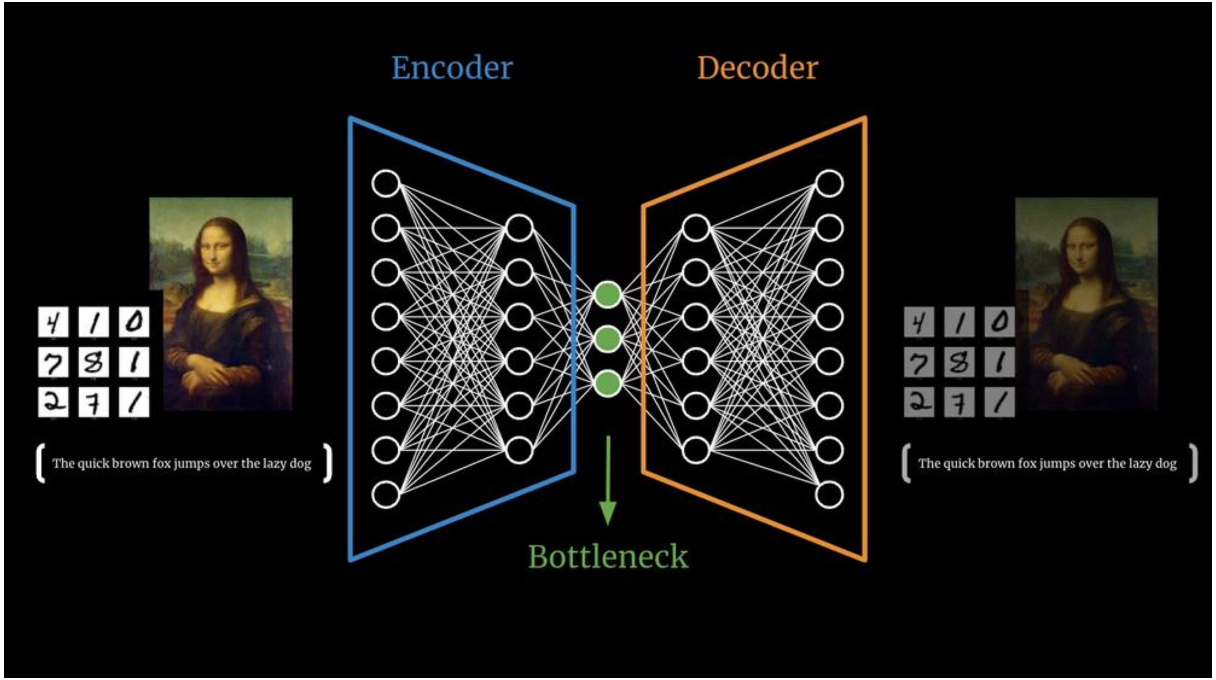
Autoencoders

Autoencoders = artificial neural networks capable of learning dense representations of the input data, called *latent representations or codings*, without any supervision (i.e., the training set is unlabeled).

much lower dimensionality! Useful for dimensionality reduction, feature detectors (unsupervised pretraining of NN) or as generative models:

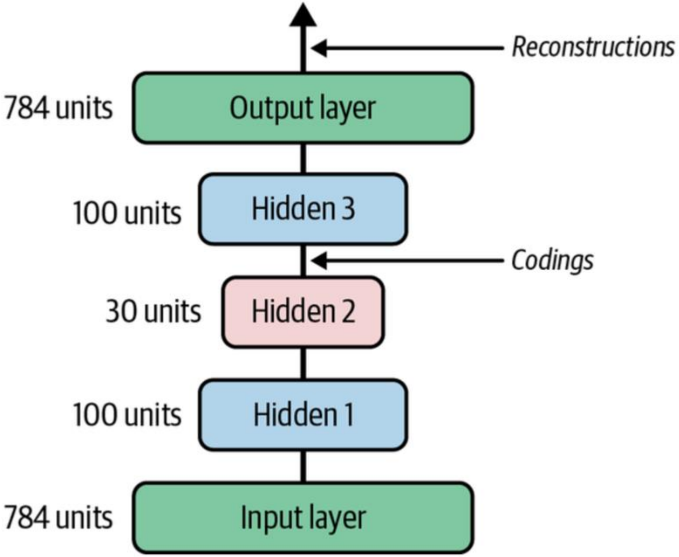


Autoencoders


Encoder
Decoder
4 1 0
7 8 1
2 7 1
The quick brown fox jumps over the lazy dog
Bottleneck
4 1 0
7 8 1
2 7 1
The quick brown fox jumps over the lazy dog

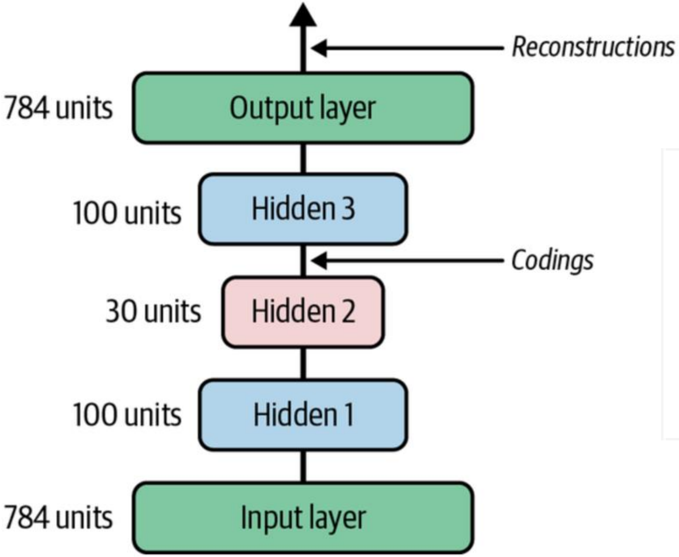
number of neurons in the output layer must be equal to the number of inputs.

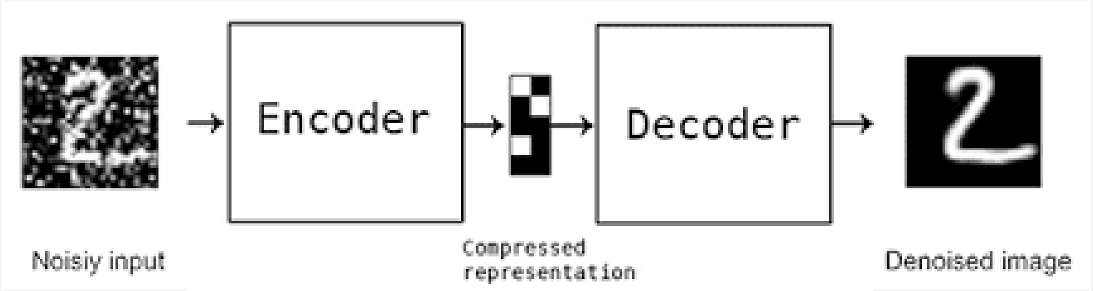
Autoencoder



```
stacked_encoder = tf.keras.Sequential([
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(100, activation="relu")
tf.keras.layers.Dense(30, activation="relu"), ])

stacked_decoder = tf.keras.Sequential([
tf.keras.layers.Dense(100, activation="relu"),
tf.keras.layers.Dense(28 * 28),
tf.keras.layers.Reshape([28, 28])
])

stacked_ae = tf.keras.Sequential([stacked_encoder,
                         stacked_decoder])

stacked_ae.compile(loss="mse", optimizer="nadam")

history = stacked_ae.fit(X_train, X_train, epochs=20,
                 validation_data=(X_valid, X_valid))
```

Autoencoders





- Learned features can be used in pre-training
- Can combine with Convolutions
- Force Learning: Denoising Autoencoders

VAE: Variational Autoencoders

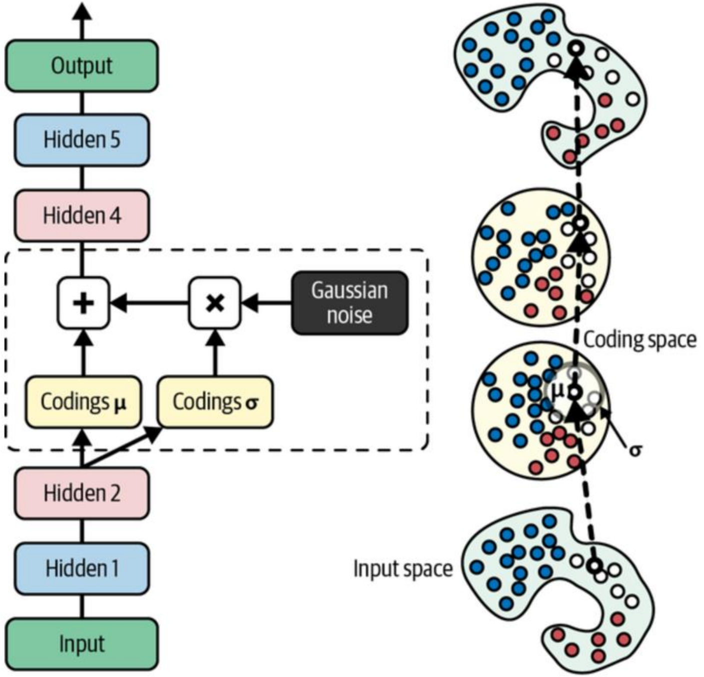
https://arxiv.org/abs/1312.6114

Specialty:

- *probabilistic* models
-> outputs are partly determined by chance, even after training

- *generative*:
-> generate new instances that look like they were sampled from the training set

How to generate <u>new images</u>?
Sample random encodings from a Gaussian distribution and decode them!



GANs: Generative Adversial Networks

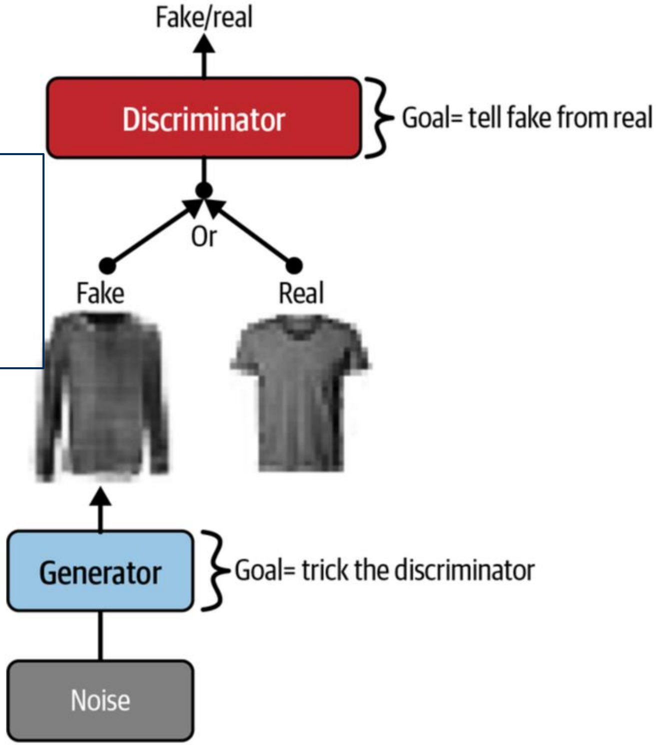
https://arxiv.org/abs/1406.2661

2 Neural Networks: Discriminator & Generator

Input: random (Gaussian) distribution
Output: an image

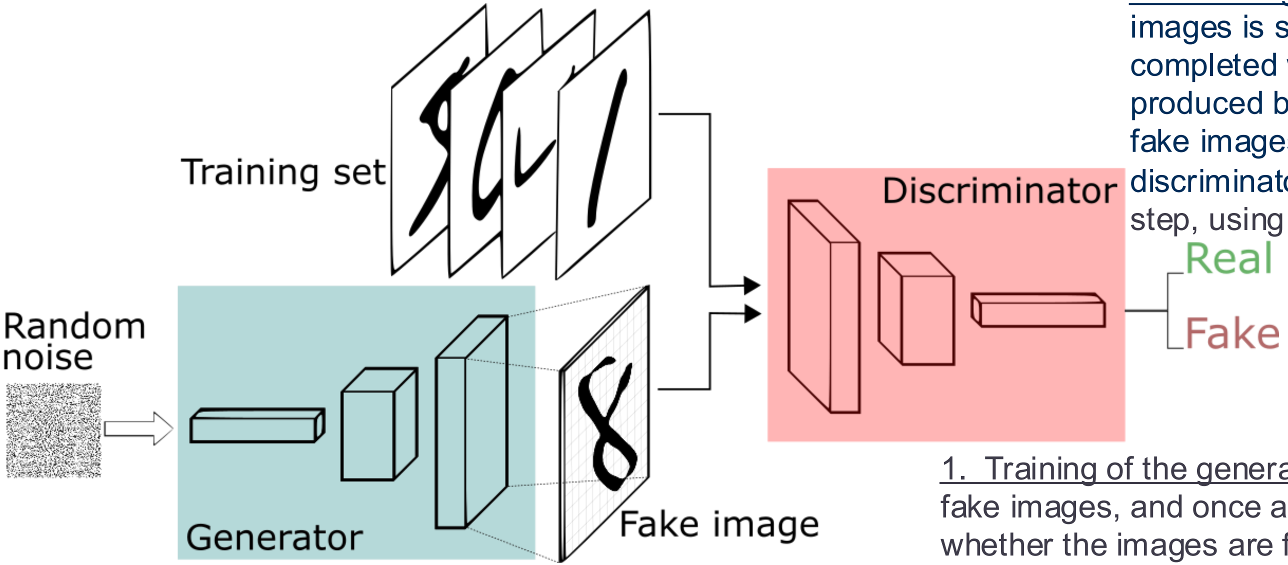
Random inputs are like latent representation, Generator is like a decoder in VAE

Input: Either a fake image from the generator or a real image from the training set, and must guess whether the input image is fake or real.



GANs: Generative Adversial Networks

2. Training of the discriminator. A batch of real images is sampled from the training set and is completed with an equal number of fake images produced by the generator. The labels are set to 0 for fake images and 1 for real images, and the discriminator is trained on this labeled batch for one step, using the binary cross-entropy loss.

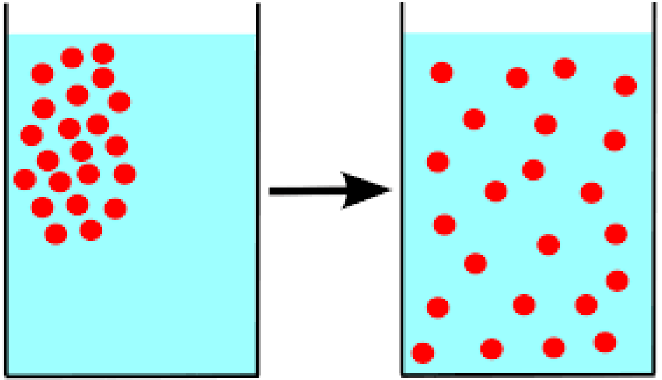


1. Training of the generator: It is used to produce another batch of fake images, and once again the discriminator is used to tell whether the images are fake or real. This time we do not add real images in the batch, and all the labels are set to 1 (real): in other words, we want the generator to produce images that the discriminator will (wrongly) believe to be real! Crucially, the weights of the discriminator are frozen during this step, so backpropagation only affects the weights of the generator.

The generator never actually sees any real images! ,But it gradually learns to produce convincing fake images guided by the gradients flowing back through the discriminator.

(Latent) Diffusion Models

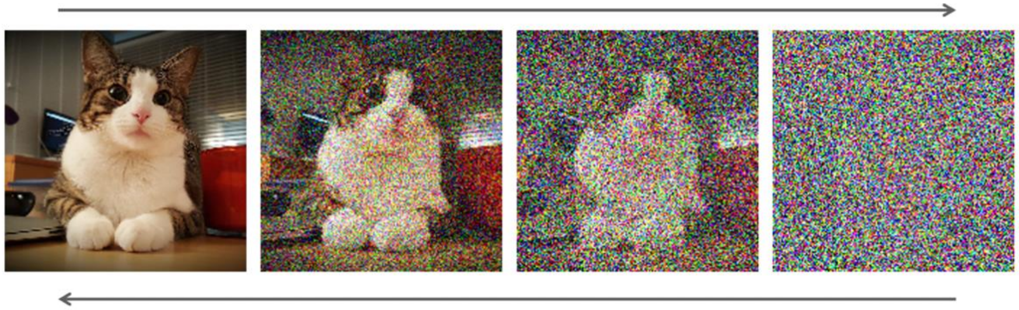
https://arxiv.org/abs/2112.10752
https://arxiv.org/abs/2102.09672

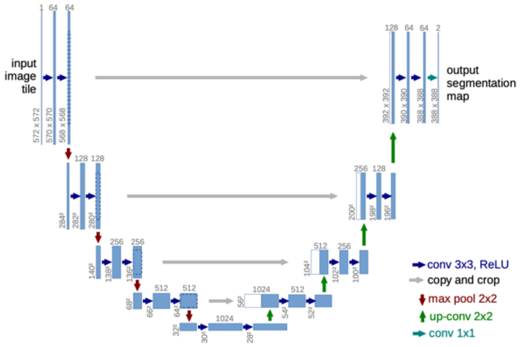


(Latent) Diffusion Models

https://arxiv.org/abs/2112.10752
https://arxiv.org/abs/2102.09672

- Diffusion models transform an image of random noise into a target image
- To gradually denoise the image we use a U-Net
- To guide the denoising process we leverage a text encoder (e.g. CLIP)



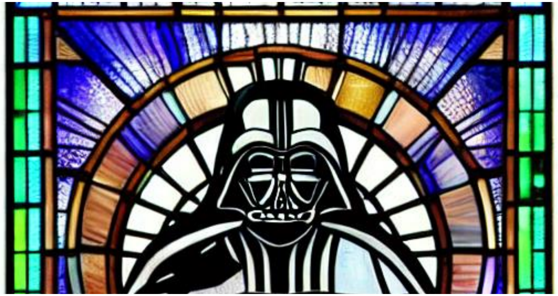


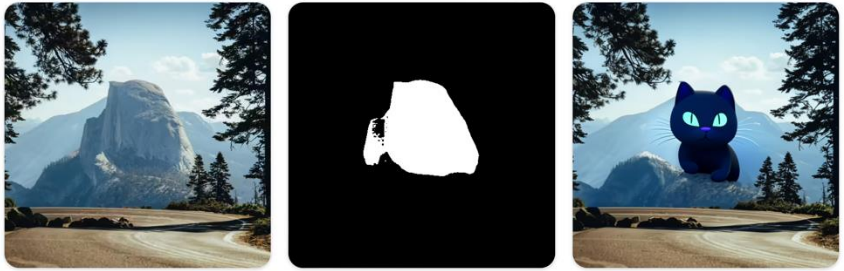
Credits: NVIDIA

Text-to-Image Diffusion Models (but also Img-2-Img, Inpainting etc)

- E.g. DALL-E3, Stable Diffusion, Midjourney, Imagen

"stained glass of darth vader, backlight, centered composition, masterpiece, photorealistic, 8k"



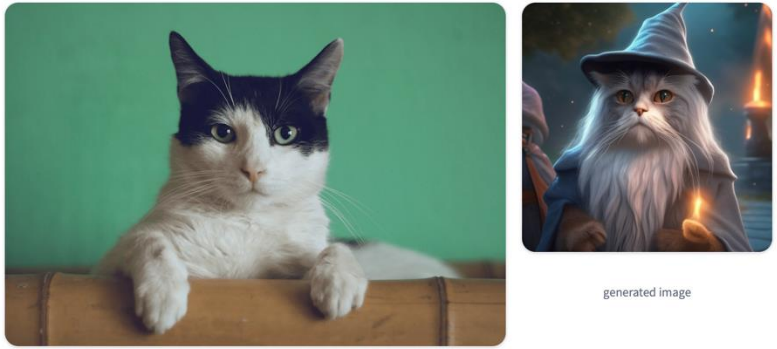


base image

mask image

generated image

“cat wizard, gandalf, lord of the rings, detailed, fantasy, cute, adorable, Pixar, Disney, 8k"

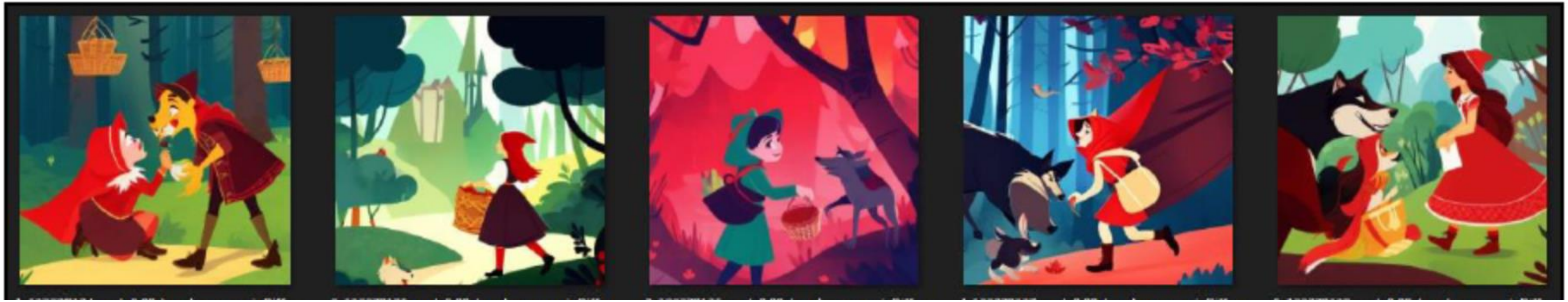


generated image

initial image

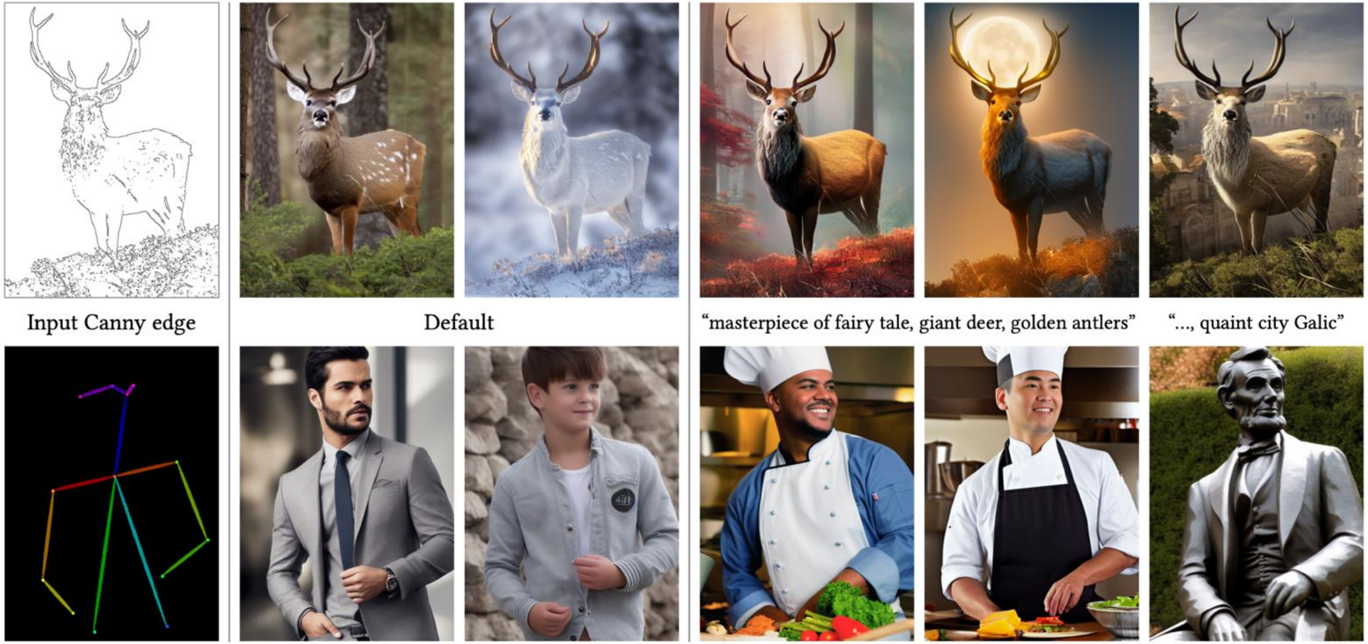
Trying to fix the limitations…

Panel 3: The wolf asking Little Red Riding Hood about the contents of the basket and where she is going.



Trying to fix the limitations…

ControlNets



Input Canny edge

Default

“masterpiece of fairy tale, giant deer, golden antlers”

“..., quaint city Galic”

Input human pose

Default

“chef in kitchen”

“Lincoln statue”

Figure 1: Controlling Stable Diffusion with learned conditions. ControlNet allows users to add conditions like Canny edges (top), human pose (bottom), *etc*., to control the image generation of large pretrained diffusion models. The default results use the prompt “a high-quality, detailed, and professional image”. Users can optionally give prompts like the “chef in kitchen”.

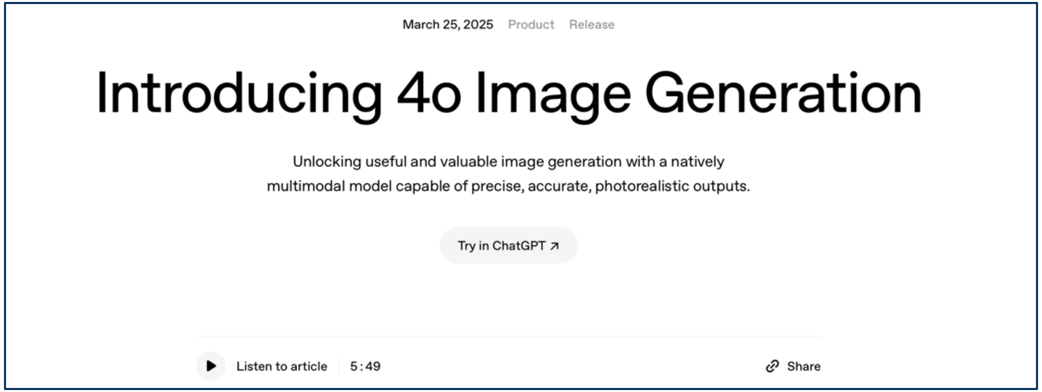
https://arxiv.org/abs/2302.05543

Life After Diffusion Models

Autoregressive Image Generation

Probably:

- https://arxiv.org/abs/2404.02905
- https://github.com/FoundationVision/VAR



March 25, 2025 Product Release

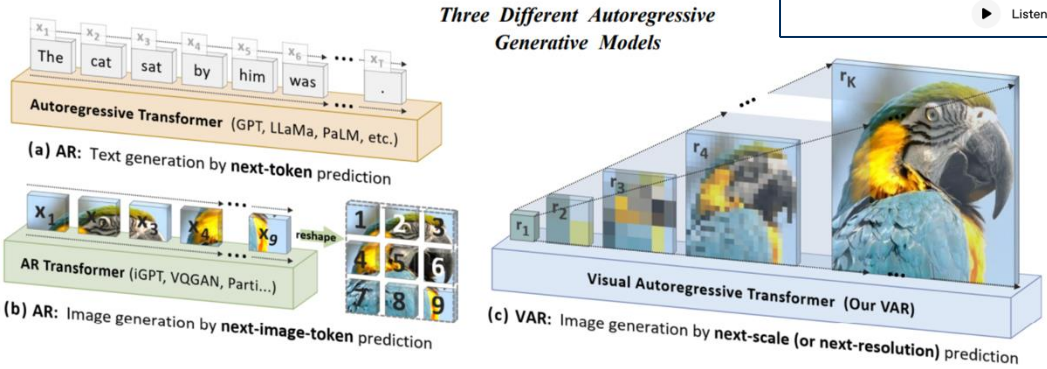
Introducing 4o Image Generation

Unlocking useful and valuable image generation with a natively multimodal model capable of precise, accurate, photorealistic outputs.

Try in ChatGPT ↗

Listen to article 5:49

Share



Three Different Autoregressive Generative Models
x1 x2 x3 x4 x5 x6 ... xT
The cat sat by him was .
Autoregressive Transformer (GPT, LLaMa, PaLM, etc.)
x1 x2 x3 x4 ... x9
reshape
1 2 3 4 5 6 7 8 9
AR Transformer (iGPT, VQGAN, Parti...)
r1 r2 r3 r4 ... rK
Visual Autoregressive Transformer (Our VAR)

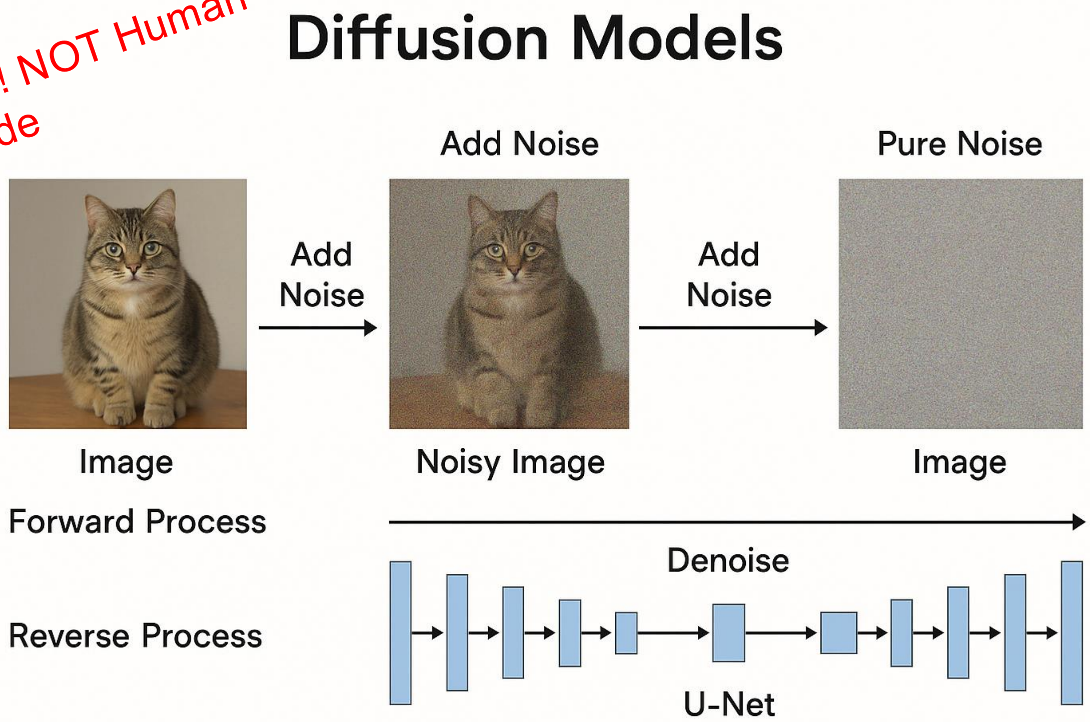
(a) AR: Text generation by **next-token** prediction

(b) AR: Image generation by **next-image-token** prediction

(c) VAR: Image generation by **next-scale (or next-resolution)** prediction

Diffusion Models explained by VisualAR (e.g GOT4o Image)

CAREFUL!!! NOT Human created Slide



Diffusion Models

Add Noise

Pure Noise

Add Noise

Add Noise

Image

Noisy Image

Image

Forward Process

Denoise

Reverse Process

U-Net

❖ Useful Reads:

- A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, Tensorflow*, O’Reilly
- https://www.kdnuggets.com/2021/03/beginners-guide-clip-model.html
- https://huggingface.co/docs/transformers/main/en/tasks/zero_shot_object_detection
- https://huggingface.co/docs/transformers/main/en/tasks/object_detection
- https://huggingface.co/docs/diffusers/index (and following)
- https://huggingface.co/docs/transformers/model_doc/owlvit
- https://towardsdatascience.com/stable-diffusion-using-hugging-face-501d8dbdd8#:~:text=Stable%20diffusion%20simply%20put%20is,image%20given%20a%20textual%20prompt.&text=As%20we%20can%20see%20from,image%20representative%20of%20the%20text.
- https://gregrobison.medium.com/tokens-not-noise-how-gpt-4os-approach-changes-everything-about-ai-art-99ab8ef5195d
- And references within the presentation

Thank you.