Image Generation with Foundation Models

GANs (Generative Adversarial Networks)

Besteht aus zwei Neuralen Netzwerken

- Der Generator
- Der Diskriminator

Erzeugen Daten durch Konkurrenz

Schwer zu trainieren da:

- Mode Collapse -> produziert immer ähnliche Varianten
- Divergenz -> produziert schlechte Bilder da ein Netz zu stark ist

Das U-Net

Architektur, um Bilder zu verstehen und wieder aufzubauen
Encoder:

- Bild wird immer verkleinert
- Wichtigsten Strukturen werden beibehalten

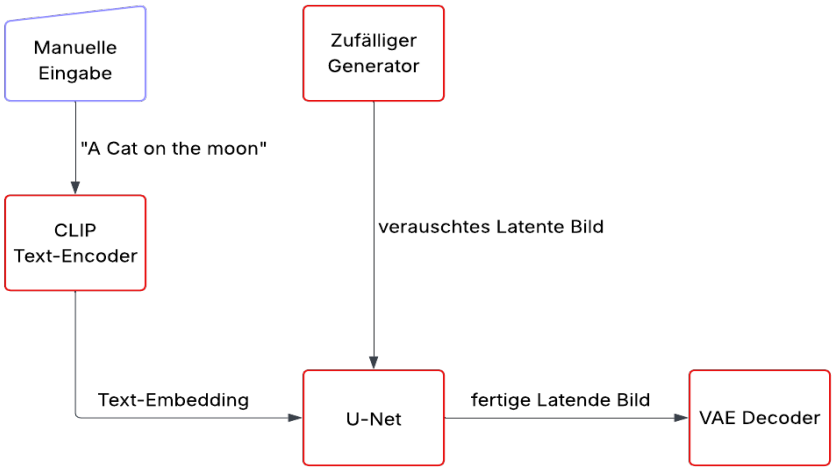
Decoder:

- Netz baut das Bild wieder auf
- Entrauscht

Skip Connection:

- Sorgen dafür des feinen Details nicht aus früheren Schichten verloren gehen

Diffusionsmodel – Ablauf



1. Manuelle Eingabe

- Text wird in Token zerlegt

2. CLIP-Text-Encoder

- Wandelt den Text in Semantischen Vektor um

3. Zufälliger Generator

- Ein rein zufälliges, verrauschtes Latente Bild wird generiert

4. U-Net

- **Encoder** reduziert die Auflösung und extrahiert Kontext.
- **Skip Connections** bringen Details zurück.
- **Decoder** baut die Auflösung wieder hoch.
- **Output:** Schätzung des Rauschens, das entfernt werden soll.
- **Sampler:** Berechnet aus dieser Schätzung das nächstweniger verrauschte Bild.

Dieser Schritt wiederholt sich mehrmals (20-50 Iterationen).

5. VAE-Decoder

- Der VAE-Decoder wandelt das Latente Bild in ein Pixelbild um

Diffusion vs Flow

Diffusionsmodell:

Versucht **vorherzusagen**, wie ein verrauschtes Bild wieder zu einem realistischen Bild wird.

Training: lernt, Rauschen Schritt für Schritt zu entfernen (Schritt-für-Schritt-Rauschvorhersage).
Generation: beginnt mit vollem Rauschen → mehrere Iterationen, in jeder wird Rauschen reduziert → Bild entsteht stufenweise.

Flow-Matching:

Versucht **vorherzusagen**, wie ein zufälliges Rauschen **direkt** in ein realistisches Bild transformiert wird.

Training: lernt direkt eine Transformation vom Rauschen zum Datensatz.
Generation: wendet diese einmalig deterministische Transformation an → Bild entsteht direkt.