diff_01_aufgaben

November 6, 2025

1 Einführung in Bildgenerierung mittels stable diffusion

Fiona Lublow, Tjark Pfeiffer

Aufgabe: Erstelle eine DiffusionPipeline mit folgendem Code und analysiere die pipeline mittels `help(pipeline)`.

[1]:

```python
# Prüfen, ob PyTorch installiert ist und Kontrollieren der Version (mindestens␣
 ↪2.5.0 empfohlen)
import torch
print("PyTorch Version:", torch.__version__)
```

```
PyTorch Version: 2.8.0+cu126
```

[2]:

```python
# Prüfen, ob GPU verfügbar ist
if torch.cuda.is_available():
    print("GPU is available.")
    print("GPU Name:", torch.cuda.get_device_name(0))
else:
    print("GPU is not available. Using CPU instead.")
```

```
GPU is available.
GPU Name: Tesla T4
```

[3]:

```
# Downgrade diffusers and transformers to compatible versions
!pip install diffusers==0.32.2 transformers==4.49
```

```
Collecting diffusers==0.32.2
  Downloading diffusers-0.32.2-py3-none-any.whl.metadata (18 kB)
Collecting transformers==4.49
  Downloading transformers-4.49.0-py3-none-any.whl.metadata (44 kB)
                           44.0/44.0 kB
1.9 MB/s eta 0:00:00
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.12/dist-packages (from diffusers==0.32.2) (8.7.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-
packages (from diffusers==0.32.2) (3.20.0)
Requirement already satisfied: huggingface-hub>=0.23.2 in
/usr/local/lib/python3.12/dist-packages (from diffusers==0.32.2) (0.36.0)
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages
(from diffusers==0.32.2) (2.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.12/dist-packages (from diffusers==0.32.2) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-
packages (from diffusers==0.32.2) (2.32.4)
Requirement already satisfied: safetensors>=0.3.1 in
/usr/local/lib/python3.12/dist-packages (from diffusers==0.32.2) (0.6.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages
(from diffusers==0.32.2) (11.3.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.12/dist-packages (from transformers==4.49) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-
packages (from transformers==4.49) (6.0.3)
Collecting tokenizers<0.22,>=0.21 (from transformers==4.49)
  Downloading tokenizers-0.21.4-cp39-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-
packages (from transformers==4.49) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.12/dist-packages (from huggingface-
hub>=0.23.2->diffusers==0.32.2) (2025.3.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.12/dist-packages (from huggingface-
hub>=0.23.2->diffusers==0.32.2) (4.15.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in
/usr/local/lib/python3.12/dist-packages (from huggingface-
hub>=0.23.2->diffusers==0.32.2) (1.2.0)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-
packages (from importlib-metadata->diffusers==0.32.2) (3.23.0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests->diffusers==0.32.2)
(3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-
packages (from requests->diffusers==0.32.2) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests->diffusers==0.32.2)
(2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests->diffusers==0.32.2)
(2025.10.5)
Downloading diffusers-0.32.2-py3-none-any.whl (3.2 MB)
                         3.2/3.2 MB
53.8 MB/s eta 0:00:00
Downloading transformers-4.49.0-py3-none-any.whl (10.0 MB)
                         10.0/10.0 MB
117.5 MB/s eta 0:00:00
Downloading
```

```
tokenizers-0.21.4-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1
MB)
                         3.1/3.1 MB
89.8 MB/s eta 0:00:00
Installing collected packages: tokenizers, diffusers, transformers
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.22.1
    Uninstalling tokenizers-0.22.1:
      Successfully uninstalled tokenizers-0.22.1
  Attempting uninstall: diffusers
    Found existing installation: diffusers 0.35.2
    Uninstalling diffusers-0.35.2:
      Successfully uninstalled diffusers-0.35.2
  Attempting uninstall: transformers
    Found existing installation: transformers 4.57.1
    Uninstalling transformers-4.57.1:
      Successfully uninstalled transformers-4.57.1
Successfully installed diffusers-0.32.2 tokenizers-0.21.4 transformers-4.49.0
```

[4]:

```python
# Pfad zum Speichern der 3 ... 5 GB grossen Modelle sinnvoll anpassen
#
# Hinweis Stand 8.10.25: Falls ein Fehler mit "...offset..." auftritt, bitte␣
 ↪die diffusers und transformers Bibliothek downgraden:
# pip install diffusers==0.32.2 transformers==4.49
# dann jupyter notebook neu starten und den Code erneut ausführen.

from diffusers import DiffusionPipeline

cache_dir = "c:\\Users\\<USERNAME>\\Models"

pipeline = DiffusionPipeline.from_pretrained(
    "stable-diffusion-v1-5/stable-diffusion-v1-5",
    use_safetensors=True,
    cache_dir=cache_dir
    )
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(
```

```
model_index.json:   0%|          | 0.00/541 [00:00<?, ?B/s]
```

```
Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]

config.json:   0%|          | 0.00/617 [00:00<?, ?B/s]

config.json: 0.00B [00:00, ?B/s]

special_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]

preprocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]

scheduler_config.json:   0%|          | 0.00/308 [00:00<?, ?B/s]

merges.txt: 0.00B [00:00, ?B/s]

text_encoder/model.safetensors:   0%|          | 0.00/492M [00:00<?, ?B/s]

safety_checker/model.safetensors:   0%|          | 0.00/1.22G [00:00<?, ?B/s]

config.json:   0%|          | 0.00/743 [00:00<?, ?B/s]

vocab.json: 0.00B [00:00, ?B/s]

tokenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/547 [00:00<?, ?B/s]

unet/diffusion_pytorch_model.safetensors:   0%|          | 0.00/3.44G [00:00<?, ?
 ↪B/s]

vae/diffusion_pytorch_model.safetensors:   0%|          | 0.00/335M [00:00<?, ?B/
 ↪s]

Loading pipeline components...:   0%|          | 0/7 [00:00<?, ?it/s]
```

```
[ ]: help(pipeline)
```

```
[ ]: print(pipeline)
```

1.1 TODO

Hierauf aufbauend bitte mit Konfigurationsmöglichkeiten und der Funktionsweise der Pipeline vertraut machen.

Die Kernfrage lautet: Wie kann die Bildgenerierung beschleunigt werden?

2 Vorgehen

Die Zeiten für die Bildgenerierung mit einer GPU werden für 10,25,50,75,100 interfence-steps gemessen und zum Vergleich nochmal für 25 steps nur auf der CPU.

```
[7]: import time
     import numpy as np
     import pandas as pd

     #prompt und neg prompt und Nutzung der Grafikkarte
```

```
#prompt="A photo of a tall green imposing alien, with vey long limbs, with big␣
 ↪black eyes, at night in a forest, full body shot, focus on the alien,␣
 ↪realistic, 4k, photo, real, dark"
prompt="A photo of a cat with black and white fur, with green eyes and wearing␣
 ↪a purple party hat on its head, laying on a brown wooden table, full body␣
 ↪shot, focus on the cat, realistic, 4k, photo, real, cozy"
negative_prompt="drawing, Artwork, Painting, Morphing"
pipeline.to("cuda")

#Funktion für Zeitmessung mit verschiedener Anzahl von steps
def get_time_and_pic(steps):
  start_time = time.perf_counter()
  image =␣
 ↪pipeline(prompt,negative_prompt=negative_prompt,num_inference_steps=steps).
 ↪images[0]
  end_time = time.perf_counter()
  print(f"Image generation with {steps} took {end_time - start_time:.3f}␣
 ↪seconds")
  display(image)
  return (end_time-start_time)

#Arrays für step und time values
step_vals=np.array([10,25,50,75,100])
#time_vals=np.array()
time_vals=[]

#testing
for steps in step_vals:
  temp=get_time_and_pic(steps)
  time_vals.append(temp)

#pandas table
table=pd.DataFrame({"steps_gpu":step_vals,"time_gpu":time_vals})
display(table)
```

```
Output hidden; open in https://colab.research.google.com to view.
```

[8]:

```
#Zeitmessung mit cpu, aufbauend auf die vorherige code zelle

pipeline.to("cpu")
table_cpu=table

#Arrays für step und time values
step_vals_cpu=np.array([25])
time_vals_cpu={}

#testing
```

```
for steps in step_vals_cpu:
  temp=get_time_and_pic(steps)
  time_vals_cpu.update({steps:temp})

#pandas table cpu werte eintragen, nur vergleichbare werte werden verwendet
for key, val  in time_vals_cpu.items():
  table_cpu.loc[table_cpu["steps_gpu"] == key, "time_cpu"] = val
display(table_cpu)
```

```
  0%|          | 0/25 [00:00<?, ?it/s]
```

```
Image generation with 25 took 500.816 seconds
```

```
   steps_gpu   time_gpu    time_cpu
```

```
0         10   6.926133         NaN
1         25  11.875367  500.815889
2         50  23.326165         NaN
3         75  36.113137         NaN
4        100  49.794656         NaN
```

3 Beobachtungen

3.1 Zeitvergleich

Die Bildgenerierung mit der CPU dauert sehr viel länger als mit Nutzung einer GPU, erkennbar in der vorherigen Tabelle.

3.2 Qualität

Ab 50 steps gibt es keine große Verbesserung mehr in der Bildqualität. Prompts werden nur teilweise mit einbezogen. Je detaillierter der Prompt geschrieben ist, desto häufiger werden Details rausgelassen oder miteinander vermischt, z.B.: der Hut der Katze fehlt oder ist grün, anstatt lila, wie die Augen.

Zu viele beschreibende Adjektive führen auch zu Verzerrungen bei den erstellten Bildern.

3.3 Fazit

Zur Verbesserung der Laufzeit reicht eine Begrenzung der interference-steps auf 50 völlig aus. Da es mit höherer Schrittezahl nicht mehr zu ausschlagebenden Verbesserungen kommt.