

# 基本のパラメータ…Seed/Strength/ Steps/CFG scaleの調整

佐藤 聖

## 実験1 Seed値の調整

### ● Seed値は生成画像の原点となるもの

Stable Diffusionのような画像生成AIにおいてSeed値は、生成される画像の原点となる数値です。同じSeed値とプロンプトを用いれば、全く同じ画像を生成できます。これにより、気に入った画像を再度生成したり、微調整したりが可能になります。Seed値を少し変えるだけで、生成される画像は大きく変化します。同じプロンプトから、全く異なる雰囲気の画像を生成できます。

### ● 実験目的

Seed値が画像生成に与える影響を明確に理解するため、画像生成AIが作りやすい一般的なテーマとして、猫の画像を作成します。

異常状態の画像を作成する際にも、プロンプトを変更することで対応できます。さらに、同じPythonプログラムやSeed値を使っても、生成される画像の特徴が大きく異なることを体験できます。

### ● 実験プログラム

Python画像生成プログラムを、リスト1に示します(このプログラムの詳細は第1部の第2章で解説した)。

リスト1中の「#パイプラインを実行」について説明します。このステップで画像を生成します。画像生成に使用するSeed値として、[1, 10, 20, 30, 40]のリストを使用し、プロンプトをprompt変数

リスト1 Seed値の調整が生成画像にどのような影響を与えるかを確認するPythonプログラム(seed01.py)

```
# ライブラリを読み込み
from diffusers import StableDiffusionPipeline
import torch

# 画像生成パイプラインを設定
pipe = StableDiffusionPipeline.from_pretrained(
    'runwayml/stable-diffusion-v1-5',
    torch_dtype=torch.float16,
    use_safetensors=True,
    variant='fp16')
pipe.to("cuda")

# パイプラインを実行
num = [1, 10, 20, 30, 40] # 乱数の固定Seed値
prompt = 'Scottish Fold, full body, reading the
          magazine, couch, detailed, 8k' # プロンプト
for i in num:
    generator = torch.Generator("cpu").manual_seed(i)
    image = pipe(prompt=prompt, generator=generator)
              .images[0]

# 画像を保存
save_file = 'seed01-' + str(i) + '.png'
image.save(save_file)
```

に格納します。for文でSeed値のリスト分繰り返し画像を生成します。

```
generator = torch.Generator("cpu").
manual_seed(i)
```

で、固定のSeed値でジェネレータ・オブジェクトを作成してパイプラインに設定します。生成された画像はimageという変数に格納されます。

### ● 実験で使用したプロンプト

リスト1をベースにして、Pythonプログラムseed02.py～seed04.pyを作成しました。プロ

表1 実験1のプログラムseed01.py～seed04.pyで使用したプロンプトの内容

No.	プロンプト	Python プログラム
1	'Scottish Fold, full body, reading the magazine, couch, detailed, 8k'	seed01.py
2	'(Scottish Fold:1.2), full body, reading the magazine, couch, detailed, 8k'	seed02.py
3	'Scottish Fold, full body, (reading the magazine:1.2), couch, detailed, 8k'	seed03.py
4	'Scottish Fold, full body, reading the magazine, (couch:1.2), detailed, 8k'	seed04.py