

OpenVINO™ を使用してインテル® GPU で Stable Diffusion を実行する方法

この記事は、Medium に公開されている「[How to run Stable Diffusion on Intel GPUs with OpenVINO](#)」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

注: この方法は CPU 上でも実行できます。

OpenVINO™ ノートブックには、いくつかの AI サンプルが用意されています。では、Stable Diffusion を実行してモデルを OpenVINO™ 中間表現 (IR) 形式に変換し、CPU や GPU で効率的に実行できることはご存知ですか? FP32 モデルを FP16 に圧縮することにより、モデルのサイズがほぼ半分に減り、実行に必要な RAM/VRAM の量も大幅に減ります。最も重要なのは、インテル® Xe® マトリクス・エクステンション (インテル® [XMV - シストリック・アレイ](#) (英語)) が有効になると、GPU 処理が大幅に高速化されることです。

以下に、ノートブックを実行して得られた結果をいくつか示します。インテル® Arc™ A770m では、(デバッグモードなしで) 1 秒あたり約 6.0 回の反復を実行できます。以下のような高品質のイメージを生成するのに、通常は約 10 秒もかかりません。

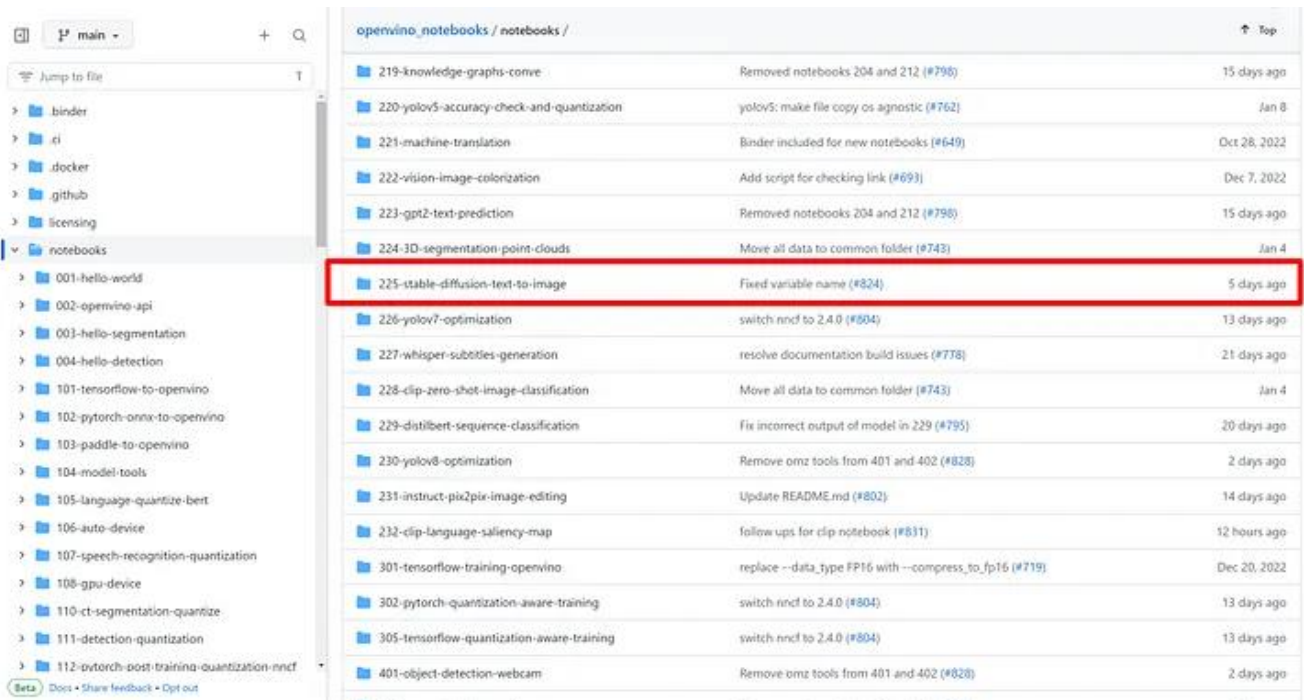


OpenVINO™ ノートブックを使用してインテル® Arc™ A770m で Stable Diffusion の Text-to-image パイプラインを実行した結果。



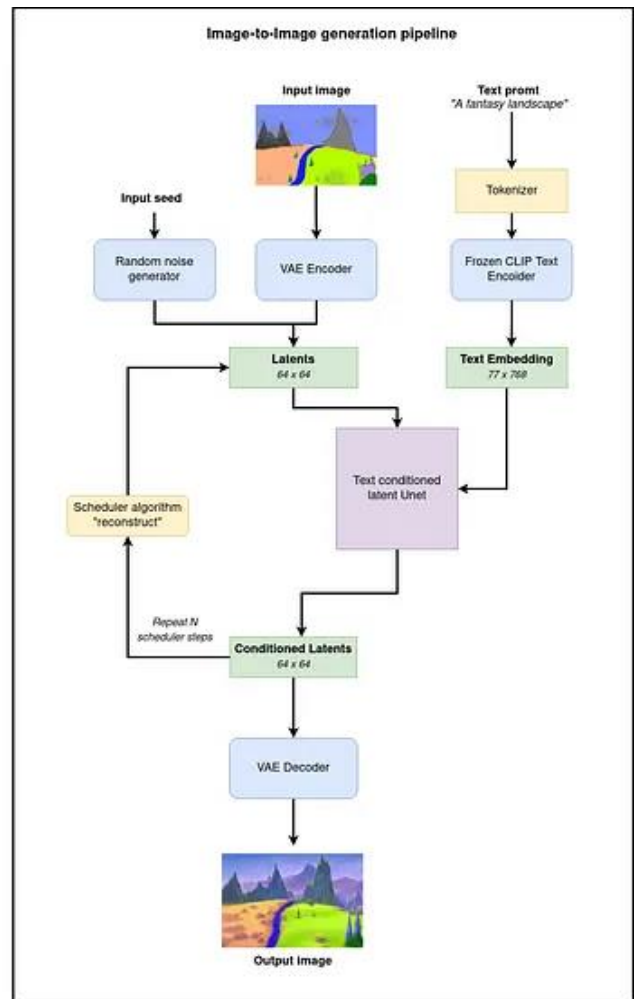
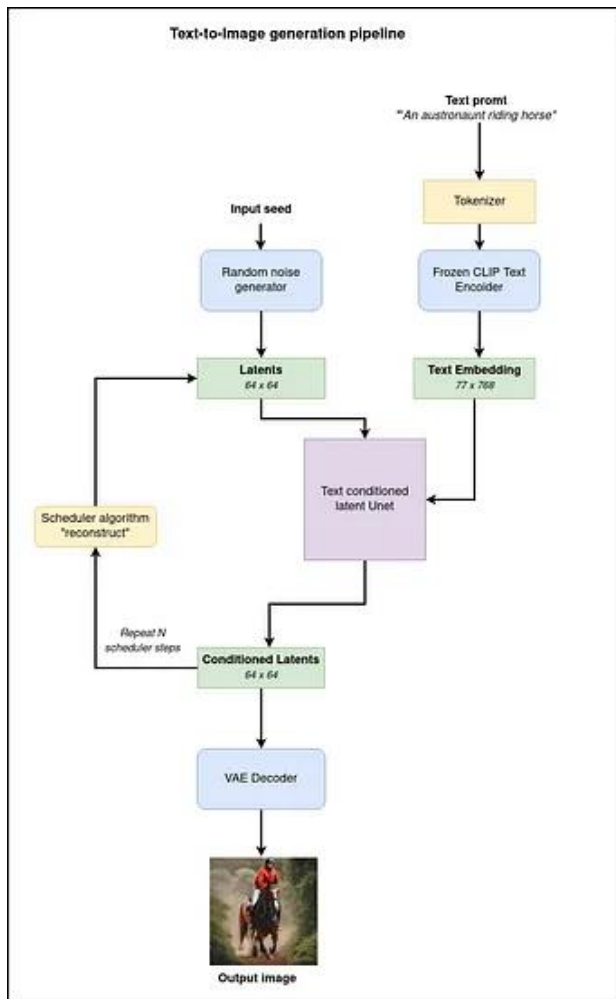
Image-to-Image パイプラインで写真を水彩画に変換した例。

まず、OpenVINO™ ノートブックのリポジトリ (英語) にアクセスします。このリポジトリに、このデモを完了するために必要なものがすべて含まれています。



Stable Diffusion は 225-stable-diffusion-text-to-image フォルダに含まれています。

このフォルダには、有名な Text-to-image パイプラインだけでなく、Image-to-Image 生成パイプラインも含まれています。では、実際にパイプラインを実行してみましょう。



パイプライン

インストール方法

最初に、OpenVINO™ ノートブックをインストールします。Windows* を使用している場合は、以下の手順に従ってください。

https://github.com/openvinotoolkit/openvino_notebooks/wiki/Windows (英語)

Linux* を使用している場合は、以下の手順に従ってください。

https://github.com/openvinotoolkit/openvino_notebooks/wiki/Ubuntu (英語)

具体的に説明します。

Python* 3.10.x (またはそれ以前) をインストールして仮想環境を作成します。

```
python3 -m venv openvino_env
source openvino_env/bin/activate #for linux
```

git clone コマンドを実行してリポジトリのクローンを作成します。

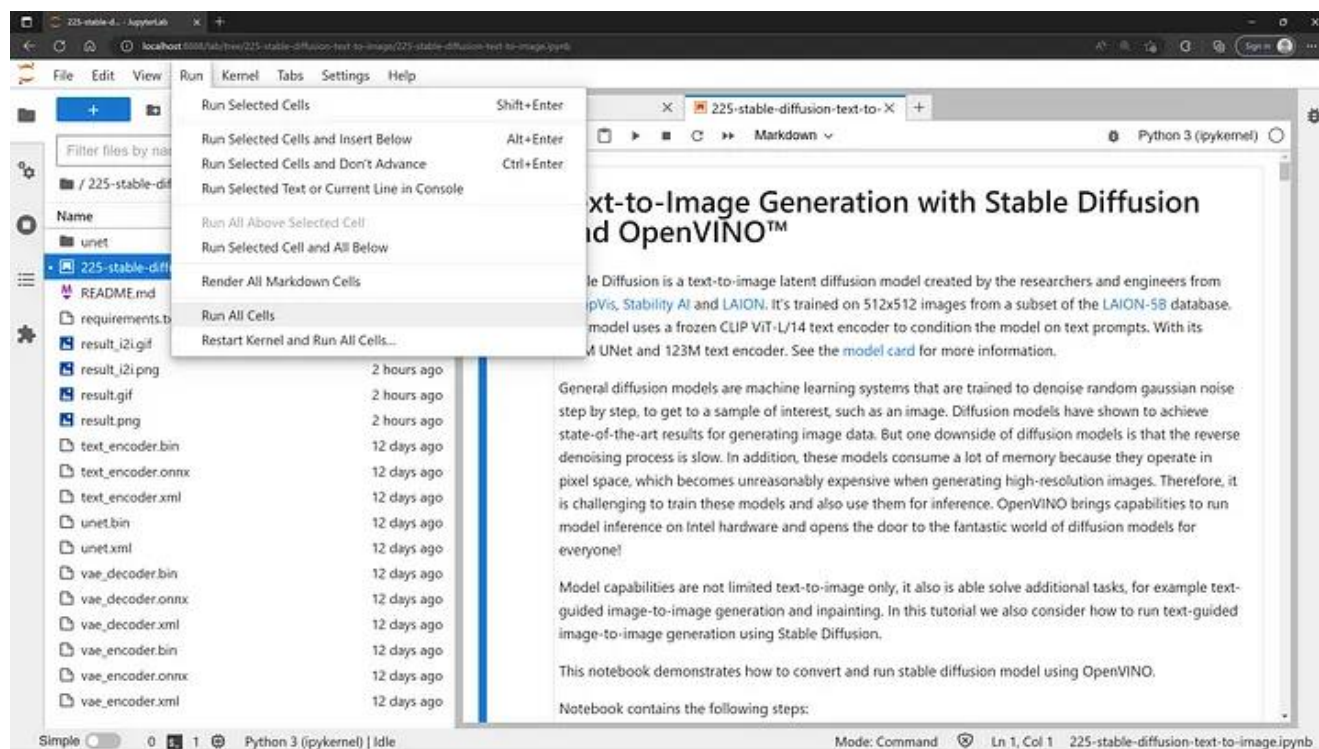
```
git clone --depth=1 https://github.com/openvinotoolkit/openvino_notebooks.git
cd openvino_notebooks
```


すべてのライブラリーと依存ファイルをインストールします。

```
pip install -r requirements.txt
```

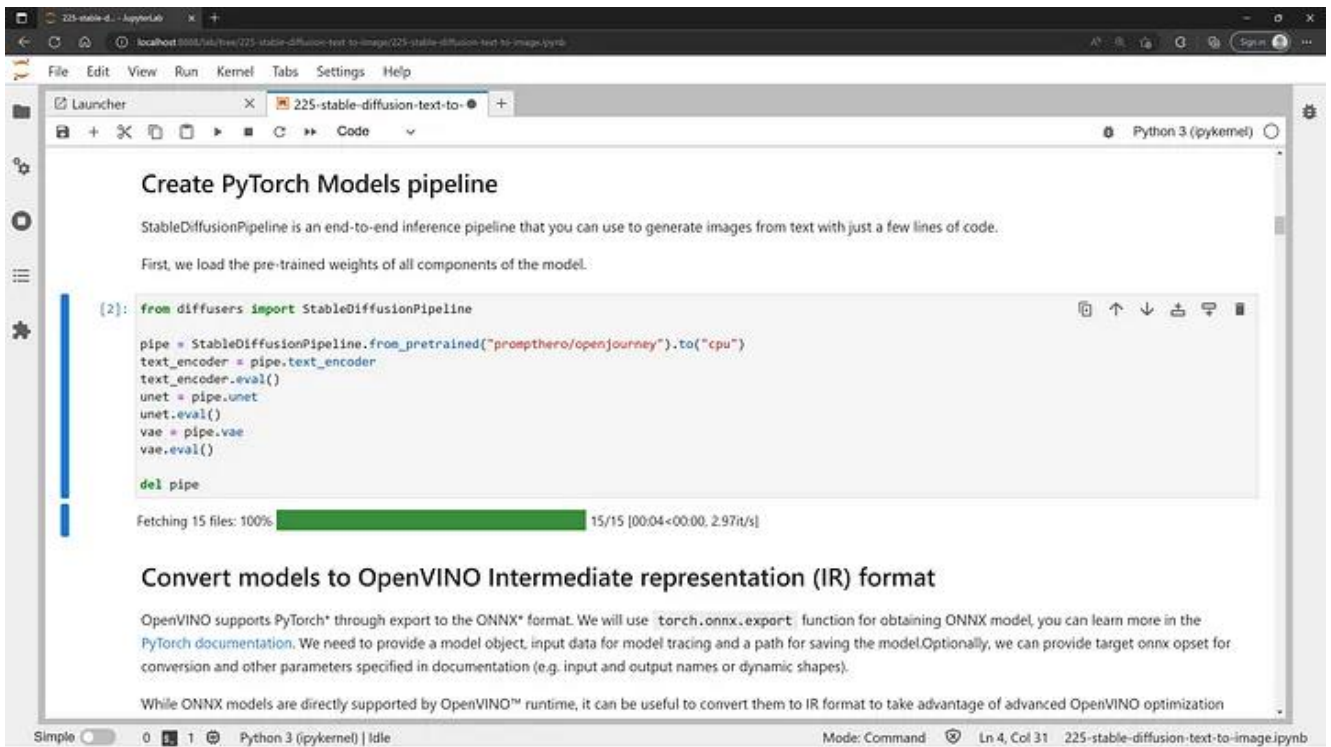
Jupyter* Notebook を実行します。

```
jupyter lab notebooks
```

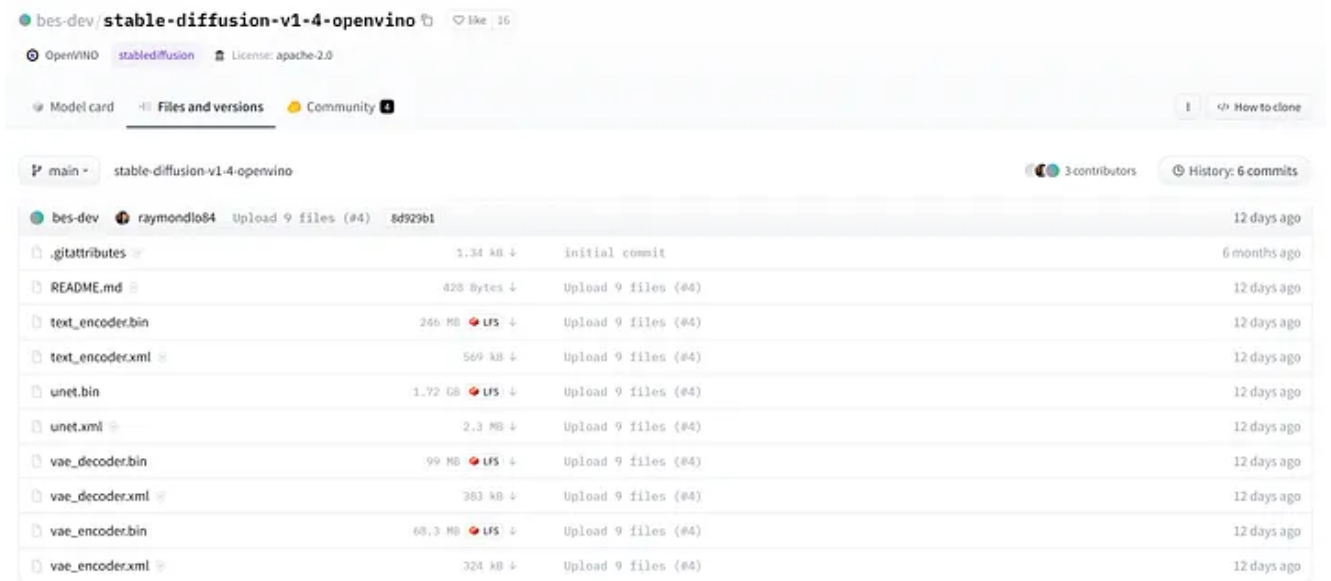


[Run All Cells] を選択して待ちます。

コードを見てみましょう。ここでは、PyTorch* パイプラインを実際に最適化し、OpenVINO™ を使用してコードを実行します。



初回のダウンロードと変換には少し時間がかかります。完了すると、IR ファイルのセットが得られます。事前トレーニング済みの、最適化されたモデルを [huggingface](#) (英語) から入手することもできます。



インテル® Arc™ GPU を利用可能な場合は、コードを「GPU」に変更できます。デフォルトは「AUTO」で、GPU を検出すると自動的に GPU に切り替わります。

The screenshot shows a JupyterLab window with a code editor. The code is as follows:

```
return timesteps, num_inference_steps - t_start
```

Configure Inference Pipeline

First, we should create instances of OpenVINO Model.

```
[49]: from openvino.runtime import Core
      core = Core()
      text_enc = core.compile_model(TEXT_ENCODER_OV_PATH, 'GPU')
```

```
[50]: unet_model = core.compile_model(UNET_OV_PATH, 'GPU')
```

```
[51]: vae_decoder = core.compile_model(VAE_DECODER_OV_PATH, 'GPU')
      vae_encoder = core.compile_model(VAE_ENCODER_OV_PATH, 'GPU')
```

Model tokenizer and scheduler are also important parts of the pipeline. Let's define them and put all components together

```
[52]: from transformers import CLIPTokenizer
      from diffusers.schedulers import LMSDiscreteScheduler

      lms = LMSDiscreteScheduler(
          beta_start=0.00085,
          beta_end=0.012,
          beta_schedule="scaled_linear"
      )
      tokenizer = CLIPTokenizer.from_pretrained('openai/clip-vit-large-patch14')
```

```
ov_pipe = OVStableDiffusionPipeline(
    tokenizer=tokenizer
```

Mode: Command Ln 4, Col 31 225-stable-diffusion-text-to-image.ipynb

GPUで実行。

The screenshot shows the same JupyterLab window, but the code is updated to use 'AUTO' for acceleration:

```
[40]: from openvino.runtime import Core
      core = Core()
      text_enc = core.compile_model(TEXT_ENCODER_OV_PATH, 'AUTO')
```

```
[50]: unet_model = core.compile_model(UNET_OV_PATH, 'AUTO')
```

```
[51]: vae_decoder = core.compile_model(VAE_DECODER_OV_PATH, 'AUTO')
      vae_encoder = core.compile_model(VAE_ENCODER_OV_PATH, 'AUTO')
```

Model tokenizer and scheduler are also important parts of the pipeline. Let's define them and put all components together

```
[52]: from transformers import CLIPTokenizer
      from diffusers.schedulers import LMSDiscreteScheduler

      lms = LMSDiscreteScheduler(
          beta_start=0.00085,
          beta_end=0.012,
          beta_schedule="scaled_linear"
      )
      tokenizer = CLIPTokenizer.from_pretrained('openai/clip-vit-large-patch14')
```

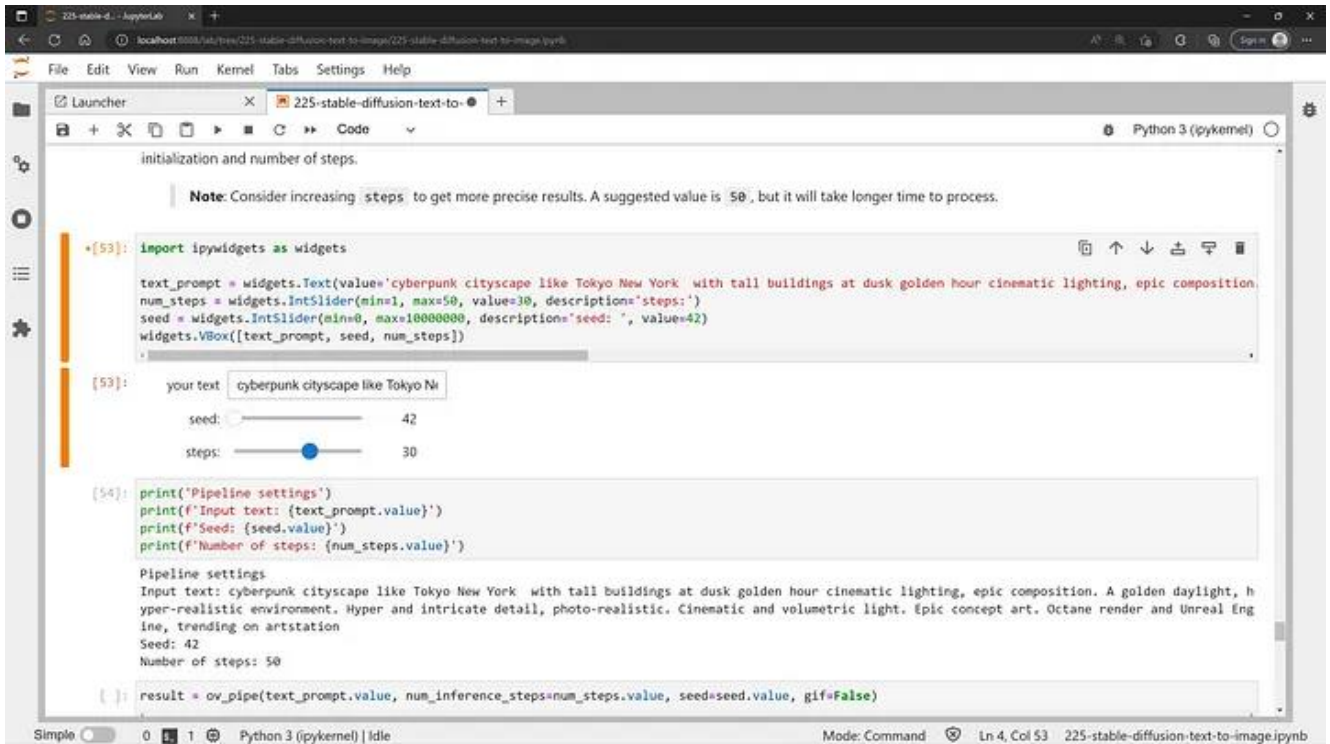
```
ov_pipe = OVStableDiffusionPipeline(
    tokenizer=tokenizer
```

Mode: Edit Ln 2, Col 60 225-stable-diffusion-text-to-image.ipynb

AUTO プラグイン。最初に CPU を使用した後、自動的に GPU に切り替わります。

ここでは、ステップを 30 に設定しました。見栄えの良い結果を得るには 50 に設定します。ここで入力テキストを変更すると、さまざまなシーンを生成できます。とてもクールな、素晴らしいイメージを生成したい場合は、コミュニティがまとめたプロンプトをいくつか試してみてください。

<https://mpost.io/best-100-stable-diffusion-prompts-the-most-beautiful-ai-text-to-image-prompts/>
(英語)



最後に、各ステップで何が起こったかを視覚化する GIF ファイルも生成しました。

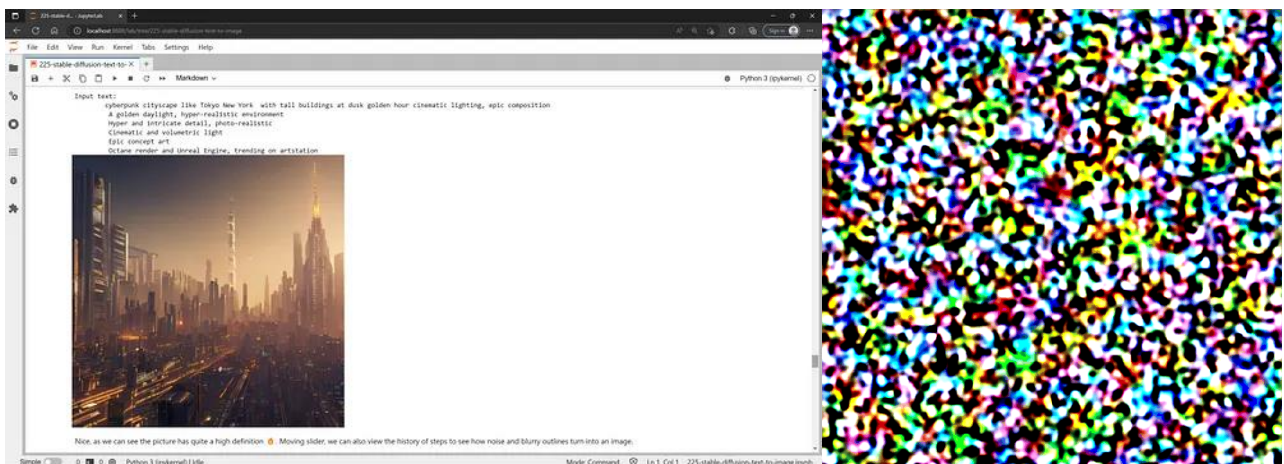
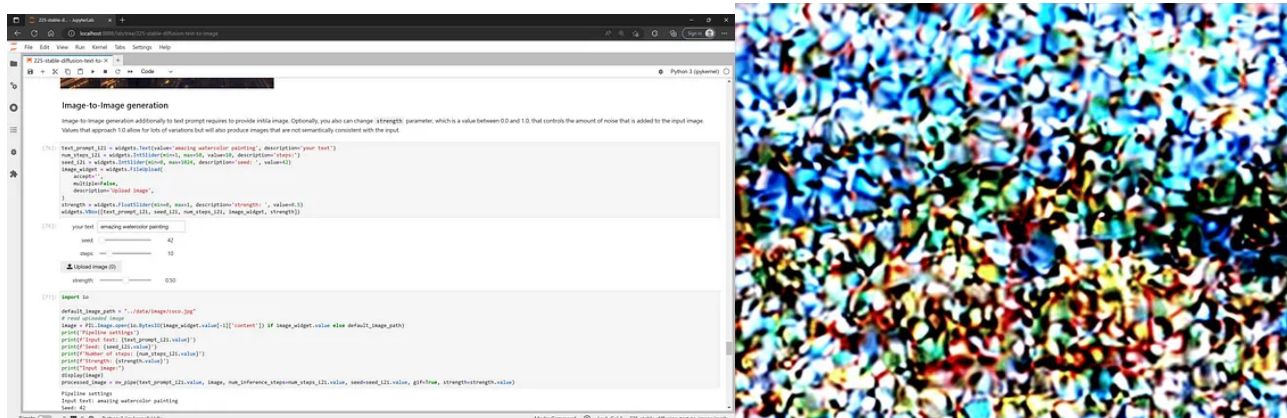


Image-to-Image パイプライン

ノートブックで作業を続けると、プロンプトを使用して最終イメージの外観に「影響を与える」こともできることがわかります。ここでは写真を水彩画に変換する例を示しました。



初期イメージとプロンプトに基づいてイメージを生成します。これで必要な結果が得られます。



Image-to-Image パイプラインの結果。

最新情報: [Stable Diffusion v2 が OpenVINO™ ノートブックで利用できるようになりました \(英語\)](#)。

まとめ

Stable Diffusion の動作を確認したい場合、およびインテルのハードウェアにおけるハードウェア・アクセラレーションの動作を確認したい場合は、OpenVINO™ ノートブックを使用するのが最適です。質問がある場合や、得られた結果を披露したい場合は、GitHub* の[ディスカッション・ボード \(英語\)](#) にコメントしてください。コーディングを楽しみましょう!

この記事が役立つ場合は、オリジナル記事の Clap ボタンを押してください。ありがとうございます!

#iamintel

OpenVINO™ ツールキットとは

AI を加速する無償のツールである OpenVINO™ ツールキットは、インテルが無償で提供しているインテル製の CPU や GPU、VPU、FPGA などのパフォーマンスを最大限に活用して、コンピュータービジョン、画像関係をはじめ、自然言語処理や音声処理など、幅広いディープラーニング・モデルで推論を最適化し高速化する推論エンジン/ツールスイートです。

OpenVINO™ ツールキット・ページでは、ツールの概要、利用方法、導入事例、トレーニング、ツール・ダウンロードまでさまざまな情報を提供しています。ぜひ特設サイトにアクセスしてみてください。

<https://www.intel.co.jp/content/www/jp/ja/internet-of-things/openvino-toolkit.html>

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。

インテルは、サードパーティーのデータについて管理や監査を行っていません。ほかの情報も参考にして、正確かどうかを評価してください。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、および非侵害性の黙示の保証、ならびに履行の過程、取引の過程、または取引での使用から生じるあらゆる保証を含みますが、これらに限定されるわけではありません。

本資料は、(明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず) いかなる知的財産権のライセンスも許諾するものではありません。

© Intel Corporation. Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。