

Medium

 Search

# Stable Diffusion Model在 latent space 裡 Interpolating 和 adding noise

VAE and Latent Space



WZX

6 min read · Just now



Share



More

## VAE (變分自編碼器)

VAE (Variational Autoencoder)是一種生成模型，專注於圖像生成。通過將輸入圖像編碼成一個潛在空間，再從這個潛在空間中重新生成圖像。VAE 的目標是學習到數據的潛在結構，並能夠基於這些結構生成新的樣本。

## Latent Space(潛在空間)

Latent Space 是一個壓縮的數學空間，能夠以較低維度的向量表示高維數據（如圖像）。在 Stable Diffusion 中，模型不直接在像素空間（Pixel Space）上運作，而是在 Latent Space 進行運算，提高效率並保留圖像的關鍵特徵。

## 設定裝置與載入模型

載入 Stable Diffusion 的 VAE（變分自編碼器）模型，用於將圖像編碼為潛在空間（latent space）以及從潛在空間解碼回圖像，

這裡使用 "runwayml/stable-diffusion-v1-5"

### stable-diffusion-v1-5/stable-diffusion-v1-5 · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

[huggingface.co](https://huggingface.co)

## 圖像預處理與編碼

將輸入圖像轉換為 latent space 表示

```
def image_to_latent(img: Image.Image):  
    img = img.resize((512, 512))  
    img = transforms.ToTensor()(img).unsqueeze(0) * 2 - 1  
    img = img.to(device, dtype=torch.float16)  
    with torch.no_grad():  
        latent = vae.encode(img).latent_dist.mean  
    return latent
```

- `img.resize((512, 512))` : 將圖像調整為 VAE 模型所需的輸入尺寸 ( 512x512 ) 。
- `transforms.ToTensor()(img)` : 將 PIL 圖像轉換為 PyTorch 張量，範圍為  $[0, 1]$  。
- `unsqueeze(0)` 在張量的第 0 維增加一個維度，PyTorch 模型通常需要批次輸入，形狀為  $(1, C, H, W)$  的張量
- `* 2 - 1` : 將張量範圍從  $[0, 1]$  映射到  $[-1, 1]$ ，這是 VAE 模型的輸入範圍。
- `vae.encode(img)` : 將圖像編碼為 latent space 表示。
- `latent_dist.mean` : 取 latent space 分佈的均值作為潛在表示。

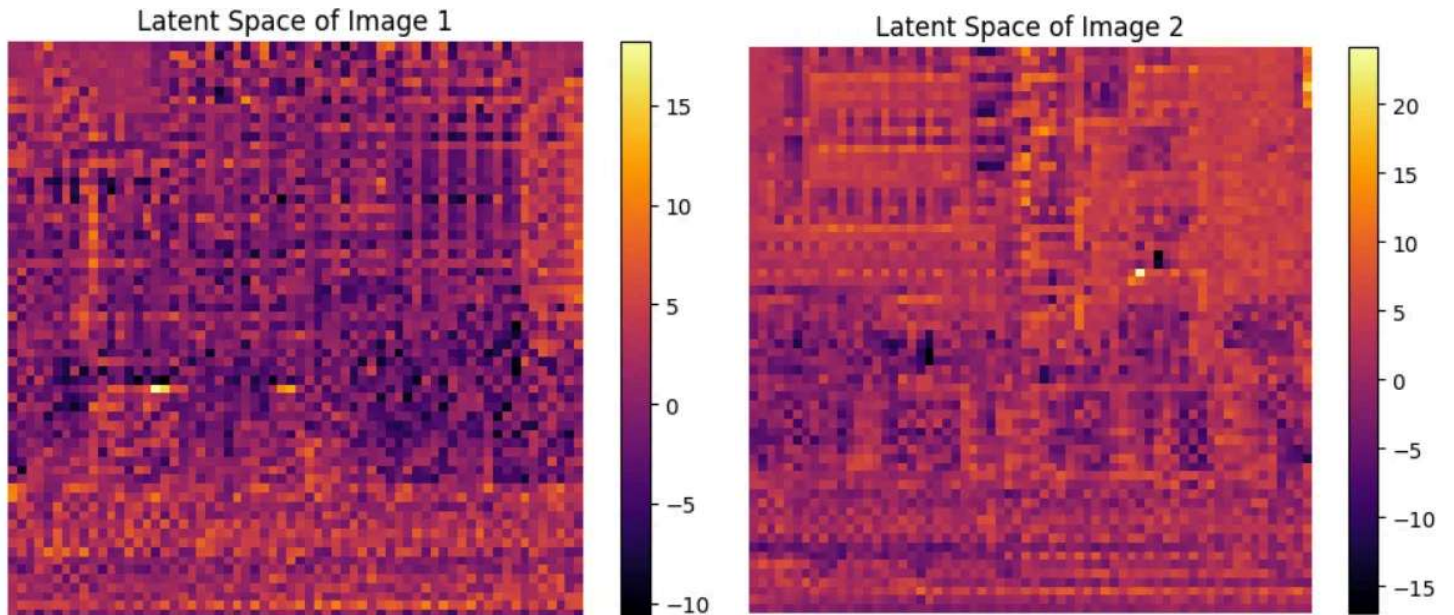


左:image1, 右:image2

```
image1 = Image.open("1.png").convert("RGB")  
image2 = Image.open("2.png").convert("RGB")  
  
latent1 = image_to_latent(image1)  
latent2 = image_to_latent(image2)
```

```
print("維度:", latent1.shape)
print("維度:", latent2.shape)
```

```
維度: torch.Size([1, 4, 64, 64])
維度: torch.Size([1, 4, 64, 64])
```



### 潛在空間插值

latent space之間進行線性插值，生成過渡影像

```
num_steps = 10 # 設定過渡影像的數量
images = []
for alpha in np.linspace(0, 1, num_steps):
    interpolated_latent = (1 - alpha) * latent1 + alpha * latent2
    interpolated_image = latent_to_image(interpolated_latent)
    images.append(interpolated_image)
```

- $(1 - \alpha) * \text{latent1} + \alpha * \text{latent2}$ ：在兩個 latent space 之間進行線性插值。

Interpolated Images



## 添加噪音

在每個步驟添加噪音可以使每個過渡圖像在細節上產生變化

```
def add_controlled_noise(latent, noise_level=0.1):  
    noise = torch.randn_like(latent) * noise_level  
    noisy_latent = latent + noise  
    return noisy_latent
```

- `torch.randn_like(latent)` : 生成與 latent space 形狀相同的隨機高斯噪音。
- `* noise_level` : 控制噪音的強度。
- `latent + noise` : 將噪音添加到潛在空間中。



Noise Level = 0.1 (Image 1)



Noise Level = 0.1 (Image 2)



Noise Level = 0.5 (Image 1)



Noise Level = 0.5 (Image 2)



Noise Level = 1.0 (Image 1)



Noise Level = 1.0 (Image 2)



## 解碼潛在空間為圖像

```
def latent_to_image(latent):  
    with torch.no_grad():  
        image = vae.decode(latent).sample  
        image = (image / 2 + 0.5).clamp(0, 1)  
        image = image.cpu().permute(0, 2, 3, 1).float().numpy()[0]  
        return Image.fromarray((image * 255).astype(np.uint8))
```

- `vae.decode(latent)`：將潛在空間解碼為圖像。
- `(image / 2 + 0.5).clamp(0, 1)`：將解碼後的圖像範圍從  $[-1, 1]$  映射回  $[0, 1]$ 。
- `permute(0, 2, 3, 1)`：將張量的維度從  $(\text{batch}, \text{channel}, \text{height}, \text{width})$  轉換為  $(\text{batch}, \text{height}, \text{width}, \text{channel})$ ，以便轉換為 NumPy 陣列。
- `Image.fromarray`：將 NumPy 陣列轉換為 PIL 圖像。



## 透過 latent space 進行Interpolation 形成兩張圖片間的過渡



Stable Diffusion

Interpolation

Noise

Pytorch



Edit profile

**Written by WZX**

6 Followers · 2 Following