Hi-Res Stable Diffusion

Deep Neural Network

Anthony Bernard
David Frécon
Junyi Li
Louis Pagnier
Léandre Perrot
Léo Sron

Février 2024

Table des matières

1 Introduction

Dans le cadre du projet final de Deep Neural Network, nous avons décider d'implémenter un modèle de diffusion pour la synthèse d'images à haute résolution en se basant sur le papier High-Resolution Image Synthesis with Latent Diffusion Models. Nous avons choisi d'entraîner notre modèle sur un dataset contenant des images de vêtements que l'on a redimensionnées en taille 512 x 512 ainsi qu'un CSV avec les descriptions des articles.

2 Latent Diffusion Model

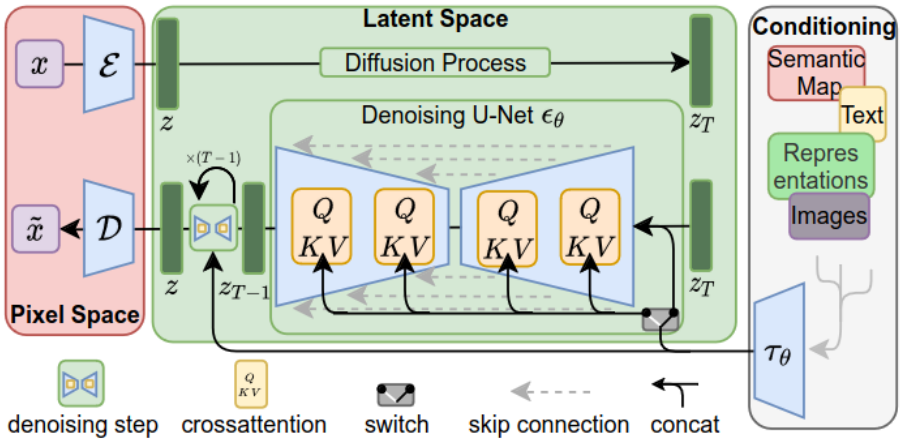

FIGURE 1 – Architecture du modèle de génération

2.1 Denoising Diffusion Probabilistic Model

Dans un premier temps, nous nous sommes concentrés sur la partie principale du modèle qui est la chaîne de Markov se réalisant dans l'espace latent. Il se trouve que toute la partie de génération du bruit Gaussien puis de débruitage avec la succession de U-Net est en fait la réalisation modifiée d'un autre papier de recherche : Denoising Diffusion Probabilistic Models. Le concept présenté dans le papier est de générer progressivement des bruits Gaussien sur des images, puis de faire apprendre au modèle de diffusion le bruit présent sur celles-ci. Ainsi, il est possible de générer des images à partir de bruits. Nos premiers essais ce sont fait sur MNIST, puis sur CIFAR10 avant de passer au dataset des vêtements.

2.1.1 Le Noise Scheduler

L'idée principale est de définir une progression pour le niveau de bruit qui sera ajoutée à l'image au fil du temps. Ce processus de diffusion de bruit est contrôlé par

un paramètre appelé T, qui représente le nombre d'étapes de diffusion. Au début, une image est initialement altérée par un bruit Gaussien. Il est ensuite progressivement atténué au fil du temps et le Scheduler décide comment ajuster la distribution du bruit à chaque étape. Après T étapes, la diffusion du bruit s'arrête et on se retrouve avec une image bruitée que le modèle doit maintenant apprendre à restaurer.

2.1.2 La chaîne de Markov : U-Net

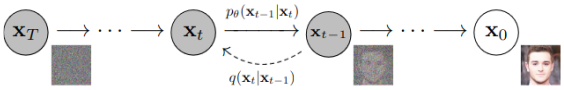

FIGURE 2 – Schéma de la chaîne de Markov

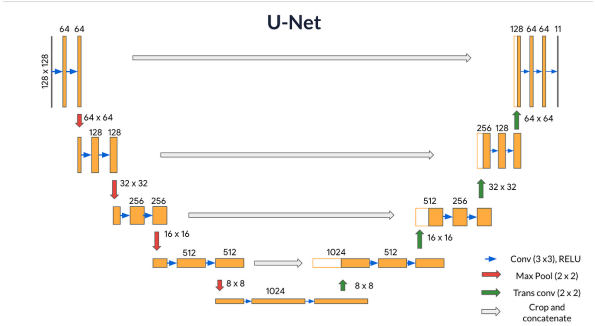

FIGURE 3 – Exemple d'architecture d'un U-Net

La chaîne de Markov est une architecture de réseaux de neurones qui permet de modéliser la distribution d'image en haute résolution à partir d'un espace latent de faible dimension. Ici, les représentations latentes utilisées pour entraîner le réseau U-Net sont apprises par un VAE réduisant la complexité de calcul et facilitant l'interpolation entre différents concepts.

2.2 Le VAE

On a fait le choix de garder un VAE classique au lieu de faire un VQ-VAE par contrainte de temps. La dimension de l'espace latent est de 16^2 que l'on a pensée convenable après avoir fait les tests sur CIFAR10, puis regardé les résultats sur le dataset final.

2.3 L'embedding de texte avec FashionCLIP

Maintenant qu'une représentation de nos images dans un espace latent a été réalisée, nous allons nous pencher sur la fonction Text-To-Image. Pour cela, on veut pouvoir

conditionner notre modèle avec un embedding du texte. Pour l'embedding du texte, nous avons opté pour l'encodeur CLIP associé a notre dataset afin d'avoir de meilleurs embeddings. L'objectif est que le modèle puisse apprendre la corrélation entre le texte et le débruitage. La finalité est que le U-Net comprenne comment manipuler l'espace latent du VAE durant le débruitage.

2.4 La Cross-Attention

Le LDM (Latent Diffusion Model) peut être conditionné par différents types d'entrées. Le mécanisme de cross attention permet d'apprendre des interactions entre les représentations latentes des images et celles des entrées conditionnelles.

$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$ où, Q(Query) est un vecteur représentant l'information que l'on cherche à obtenir, il est défini comme projection linéaire des représentations latentes. K(Key) agit comme un filtre pour déterminer les informations importantes pour la requête et V(Value) contient l'information que l'on veut extraire. K et V sont des projections linéaires des représentations conditionnelles. Le mécanisme de cross attention est intégré dans l'architecture U-Net du LDM, comme présenté dans la figure 1.

Cependant, pour le moment on utilise l'embedding de texte de la même manière que l'embedding du temps dans DDPM, c'est-à-dire avec une simple addition dans la forward pass de chaque bloc résiduel du U-Net.

3 Résultats

Voici les résultats de notre modèle de diffusion pour les requêtes suivantes : a red dress, blue T-Shirt, pink tshirt, a yellow jean et black dress.

FIGURE 4 – Résultat de notre modèle de diffusion

On peut voir les résultats de la synthèse des images tout à droite, les autres images représentes l'évolution de la génération pendant l'inférence du modèle. On peut voir qu'on a réussi à diriger le modèle à générer les vêtements avec les bonnes couleurs et les bonnes formes générales.

On pourrait améliorer nos résultats en entraînant avec la cross-attention. De plus, le papier indique qu'il faudrait avoir un VQ-VAE qui individuellement devrait donner de très bons résultats, ce qui n'est pas notre cas.