

IA Générative: Générateur Naruto avec **Adversarial Diffusion Distillation**

Sebastien GASIOR, Marc-Antoine KALMUK, Titouan VETIER et Dean BAH



Etapes clefs projet

1

Chargement des données

Images et légendes Naruto.

2

Finetuning du teacher

Avec des prompts personnalisés.

3

Construction du student

À partir d'un UNet simplifié.

4

Entraînement

Distillation loss et adversarial loss.

5

Évaluation

Scores FID et LPIPS.

6

Génération

Visuels de comparaison.

7

Export

Modèle student pour usage rapide.



Le projet est développé à l'aide de PyTorch et de la bibliothèque Diffusers de Hugging Face. L'évaluation de la correspondance perceptuelle entre images générées et attentes visuelles est effectuée à l'aide de LPIPS, tandis que FID permet d'évaluer la qualité globale des images générées par rapport aux vraies images.

Overview of the Present Project



Background of Generative Models

Les modèles génératifs ont considérablement évolué pour créer des images réalistes à partir de descriptions textuelles.



Problem Statement

Les modèles actuels, comme la diffusion stable, sont coûteux en termes de calcul, ce qui les rend difficiles à utiliser en temps réel sur des appareils à faible consommation.



Project Aim

Ce projet vise à appliquer la distillation par diffusion contradictoire (ADD) pour créer un modèle plus léger tout en conservant la qualité visuelle, spécifiquement pour l'univers Naruto.

Méthodologie: Adversarial Diffusion Distillation

Modèle Teacher

Un modèle Stable Diffusion finetuné sur un corpus d'images de Naruto, servant de référence en matière de qualité d'image.

Modèle Student

Une version plus compacte avec une architecture UNet simplifiée, entraînée à imiter les sorties du modèle teacher.

Discriminateur

Un réseau qui distingue les images du teacher de celles du student, incitant le student à générer des images plus convaincantes.

La méthode ADD repose sur ces trois éléments principaux. L'entraînement combine une loss de distillation (alignement sur les sorties du teacher) et une loss adversariale (tromper le discriminateur).

Dataset: Images de l'univers Naruto



A man with a mohawk in his hair



A man wears a blue forehead protector



A man with white hair and a sword

Sources

Captures d'écran d'épisodes animés, extraits de mangas et fan arts.

Descriptions

Légendes textuelles générées ou annotées à la main pour capturer les éléments clés de chaque image.

Prétraitement

Uniformisation des dimensions, du format de couleur et alignement avec les légendes.

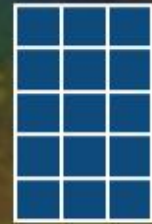
Le dataset utilisé pour l'entraînement est constitué d'images issues de l'univers de Naruto. Chaque description vise à capturer les éléments clés de l'image : personnage, action, environnement, émotions, etc.

Stable Diffusion

« Naruto Uzumaki roule en suzuki »

77 tokens

Text Encoder
(CLIPText)



Token embeddings



Random image information tensor

Image Information Creator
(UNet + Scheduler)

UNet Step
1

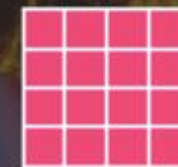


UNet Step
2



...

UNet Step
50



Processed image information tensor

Image Decoder
(Autoencoder decoder)

Generated image



Diffusion

INTEGRATION STABLE DIFFUSION

Version : runwayml/stable-diffusion-v1-5

Précision: 16-float

Code:

```
pipe = load_pipeline(model_id="runwayml/stable-diffusion-v1-5")  
tokenizer = pipe.tokenizer
```

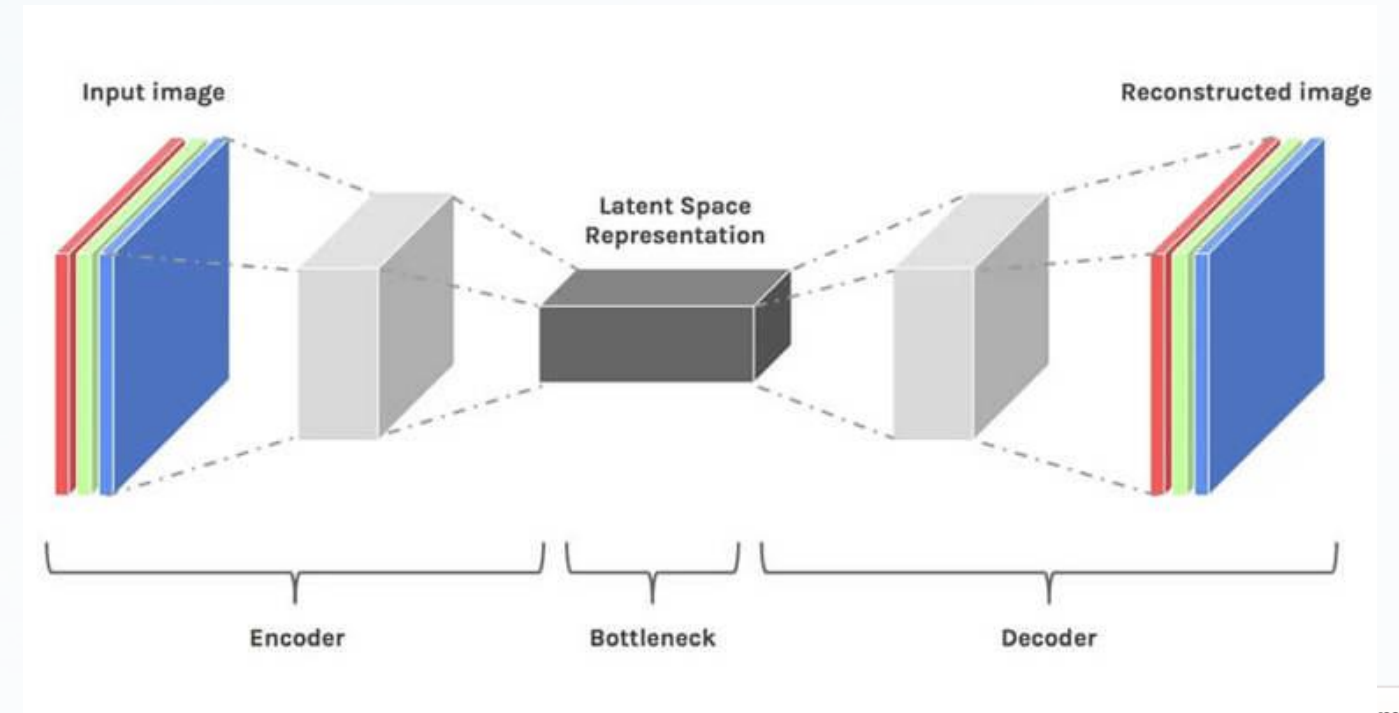
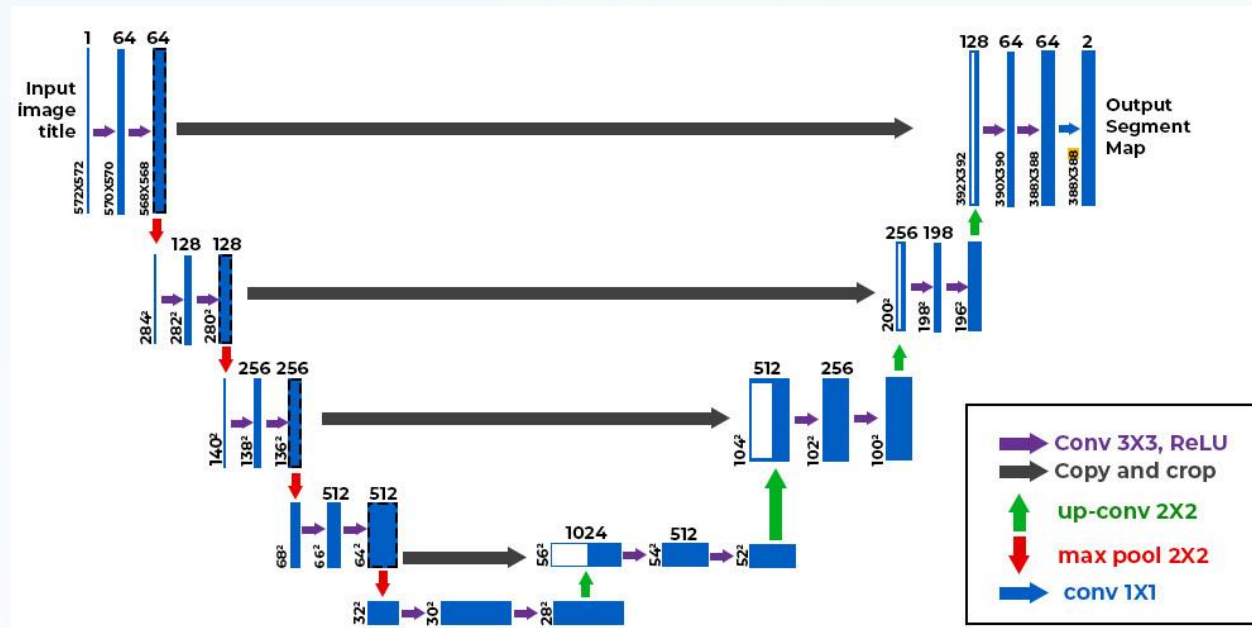
Autre notion:

Tokenizer : CLIPTokenizer

Scheduler : PNDMScheduler

Unet: UNet2DConditionModel

CLIP Text



Diffusion

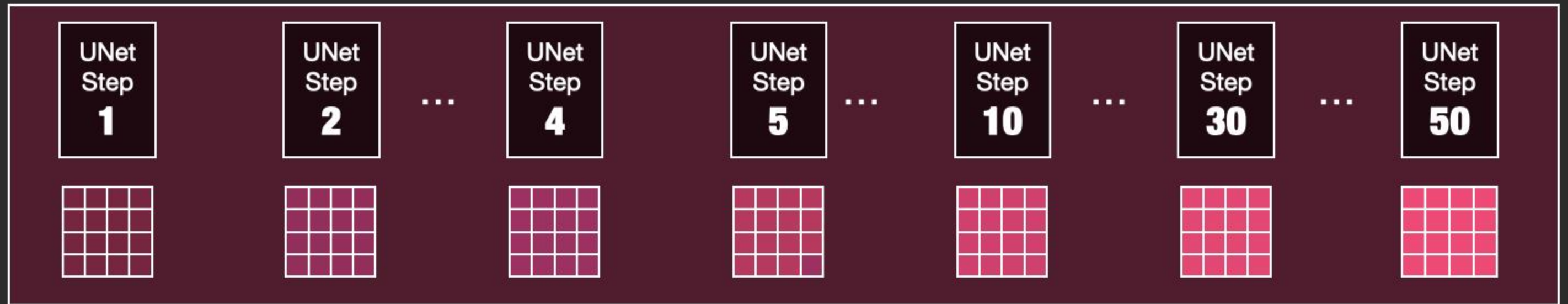


Image Information Creator

Image
Decoder
(Autoencoder
decoder)



Implémentation de PyTorch

Qu'est ce que PyTorch ?

PyTorch est une bibliothèque Python développée par Facebook AI.
Elle permet de construire, entraîner et déployer des modèles de deep learning, notamment en gérant :

Les tenseurs (équivalents de tableaux multidimensionnels, comme des matrices)

Les opérations mathématiques optimisées (CPU & GPU)

Le calcul automatique des gradients pour l'entraînement (backpropagation)

L'interopérabilité avec CUDA pour utiliser la puissance du GPU

Pourquoi on en a besoin?

Notre projet consiste à adapter Stable Diffusion à l'univers de Naruto. Pour cela, on utilise une méthode de fine-tuning appelée LoRA.
PyTorch intervient à chaque étape critique de ce processus :

Étape	Rôle de PyTorch
Prétraitement d'images/textes	Conversion en tenseurs pour traitement GPU
Encodage VAE	Propagation des images dans le modèle
Ajout de bruit	Génération de bruit via <code>torch.randn_like()</code>
Propagation avant du modèle UNet	Calcul des prédictions
Calcul de la perte	Utilisation de <code>torch.nn.MSELoss()</code>
Rétropropagation	<code>loss.backward()</code> pour ajuster les poids LoRA
Optimisation	<code>optimizer.step()</code>
Gestion GPU	<code>.to("cuda")</code> , <code>torch.cuda.empty_cache()</code>

Le projet est développé à l'aide de PyTorch et de la bibliothèque Diffusers de Hugging Face. L'évaluation de la correspondance perceptuelle entre images générées et attentes visuelles est effectuée à l'aide de LPIPS, tandis que FID permet d'évaluer la qualité globale des images générées par rapport aux vraies images.

Qu'est-ce que LoRa

Pourquoi on réentraîne le modèle ?

Stable diffusion a plus d'un milliard de paramètres :

- Prend trop de mémoire GPU
- Nécessite des semaines d'entraînement
- Peut détruire ses compétences générales d'origine

Dans notre code

```
attn_processors[name] = LoRAAttnProcessor(...)
```

Injecte des mini-couches entraînables dans le modèle

Seules ces mini-couches seront modifiées pendant l'entraînement

Tout le reste du modèle reste figé et stable

Low-Rank Adaptation :

Technique de fine-tuning ultra légère:

Permet d'adapter notre modèle aux images Naruto

On ajuste de tout petits modules (matrices A et B) dans les couches d'attention

Résultats: Vitesse vs. Qualité

Rapidité

Le modèle student est plus rapide que le teacher, permettant une utilisation plus fluide sur des machines aux ressources limitées.

Les performances du modèle student sont légèrement supérieures en termes de rapidité par rapport au modèle teacher. Qualitativement, les images produites par le modèle teacher présentent un haut niveau de détail et reproduisent fidèlement les éléments visuels emblématiques de l'univers Naruto.

Qualité

Les images du teacher sont plus détaillées et fidèles à l'univers Naruto, tandis que celles du student manquent de netteté et de cohérence.

Évaluation Quantitative: Scores FID et LPIPS

Teacher - Naruto meditating on a mountain under the sunset



Student - Naruto meditating on a mountain under the sunset




100

Étapes de diffusion pour le modèle teacher.

4

Étapes de diffusion pour le modèle student.

 Comparaison Student vs Teacher:
FID Score : 286.3204
LPIPS Score : 0.7306 (plus proche de 0 = plus proche visuellement)

Le score FID du student indique une différence notable par rapport au dataset réel, reflétant les limites de la compression. Le score LPIPS suggère une similarité perceptuelle en termes de correspondance texte-image.

Problèmes rencontré

Complicé de créer un environnement stable pour toutes les libraires

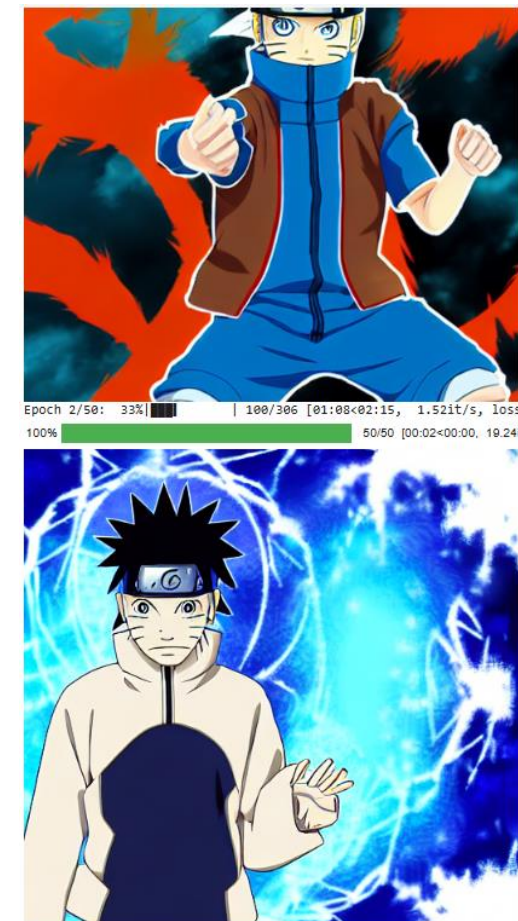
Problème d'injection des poids LoRa

Limité par les capacités de nos GPU

Paieement de google colab pro sans succès

Modèle de plus en plus nul

Utilisation de scale way instance GPU en SSH remote sur vs code





Discussion: Compromis et Limitations

Compromis

L'approche ADD offre un excellent compromis entre vitesse et qualité, permettant une utilisation dans des contextes où la latence est une contrainte.

Limitations

La stabilité de l'entraînement adversarial nécessite des ajustements précis, et la qualité visuelle peut parfois souffrir.

Améliorations

Utilisation de loss perceptuelles, entraînement multiéchelle et intégration de supervision multimodalité.

Ce projet montre qu'un modèle de diffusion allégé peut rivaliser avec un modèle complet sur un domaine aussi riche et stylisé que Naruto. Des améliorations possibles incluent l'utilisation de loss perceptuelles, l'entraînement multiéchelle, ou l'intégration de supervision multimodalité.



Conclusion: Perspectives d'Avenir

Réalisation

Mise en œuvre d'une version accélérée et compacte d'un modèle de diffusion pour la génération d'images stylisées Naruto.

Avantages

Combinaison des avantages des grands modèles génératifs avec la rapidité d'un modèle allégé.

Applications

Perspectives intéressantes pour des applications créatives, interactives ou embarquées.

En conclusion, ce projet a permis de mettre en œuvre une version accélérée et compacte d'un modèle de diffusion pour la génération d'images stylisées Naruto. À l'avenir, cette approche pourra être appliquée à d'autres univers visuels, ou utilisée dans des outils accessibles aux artistes, aux fans, ou aux développeurs de jeux vidéo.