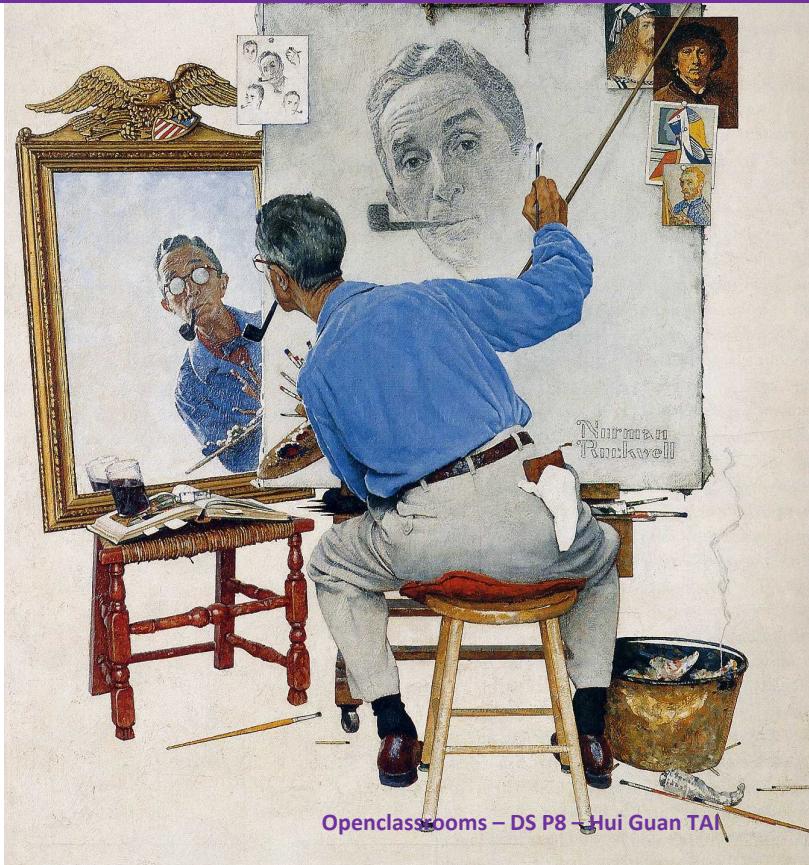


DS P8 – Développez une preuve de concept: StyleGAN



Sommaire

1. Introduction
2. Principes
3. Algorithme
4. Tests
5. Conclusion



Introduction - Contexte

- Article: <https://arxiv.org/pdf/2112.11641.pdf>
- Repository: <https://github.com/mchong6/JoJoGAN>
- Dataset:
 - Références



carey_mulligan.jpg



IMG_8457.jpg



jean-pierre-darroussin.jpg



Jonathan-Cohen.jpg



jprouve.jpg



margot-robbie.jpeg

- Styles (bd, films d'animation, peinture...)



actarus.jpg



blueberry.jpg



caricature.jpg



cpeyravernay-03.jpg



giraud-blueberry.jpg



giraud1.jpg



jimenez.jpg



kaneda.jpg



lfreud.jpg



lucaboni.jpg



raiponce.jpg



surfeurdargent.jpg

- Objectif:

Mettre en œuvre l'algorithme de styling jojoGAN pour définir un style détaillé de portrait et pouvoir l'appliquer sur n'importe quel portrait

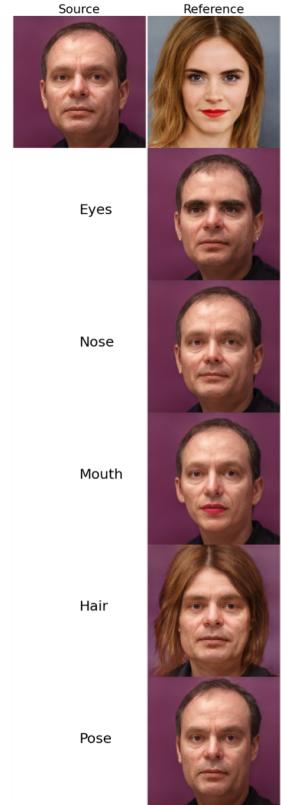
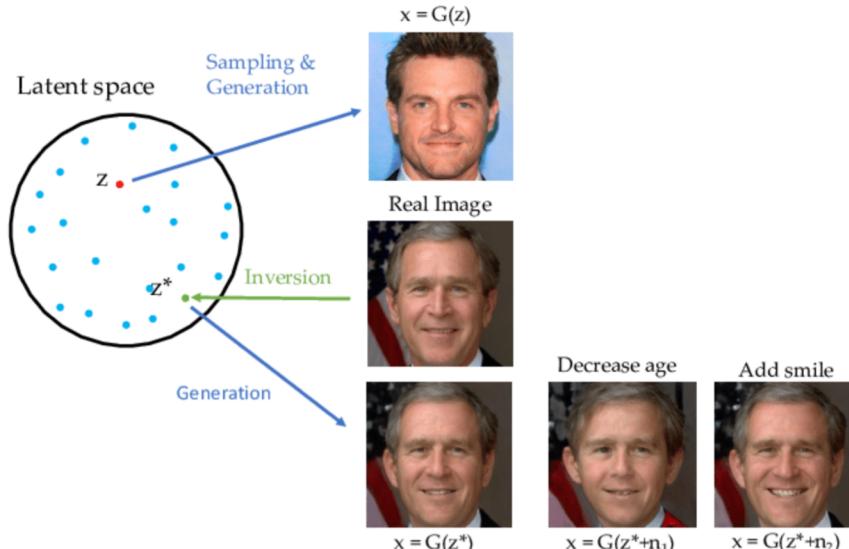
Introduction - Problématique

- Capturer le style à partir d'une seule image avec peu de données?
- Appliquer fidèlement le style détaillé sur toute photo de référence?

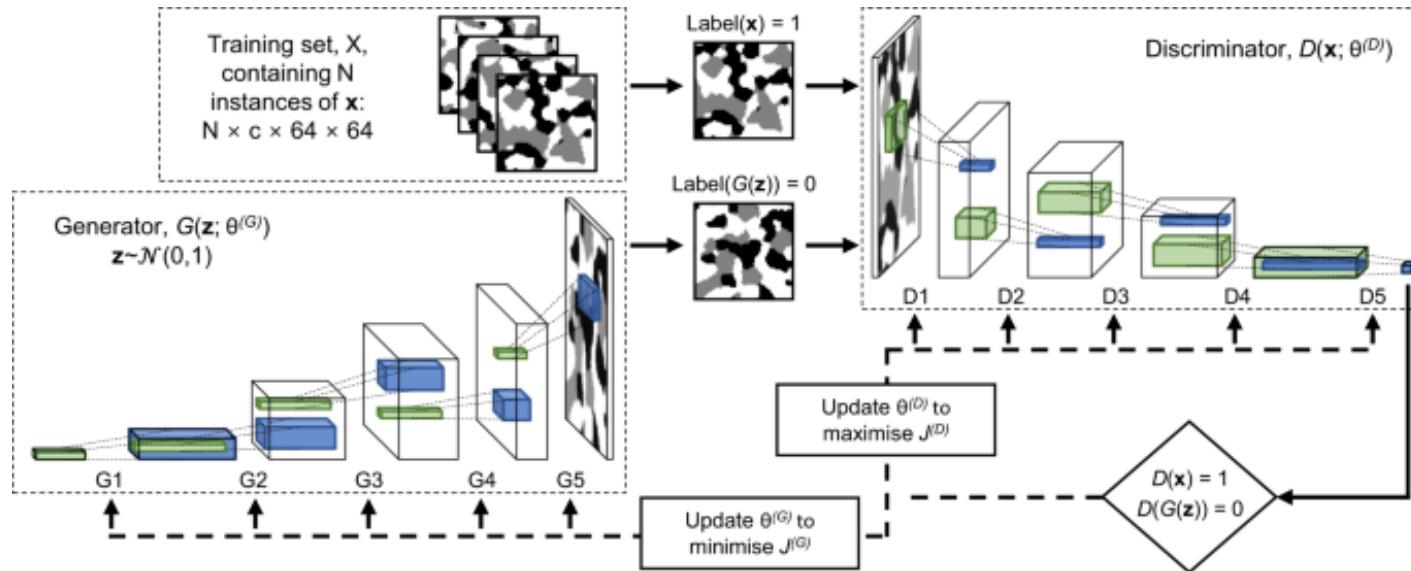


Principe – GAN Inversion

- Permet de
 - Inverser une image dans l'espace latent d'un modèle GAN pré-entraîné
 - Reconstruire fidèlement l'image à partir du code inversé par le générateur
 - Manipuler l'image et générer une nouvelle image



Principe – Generative Adversarial Networks



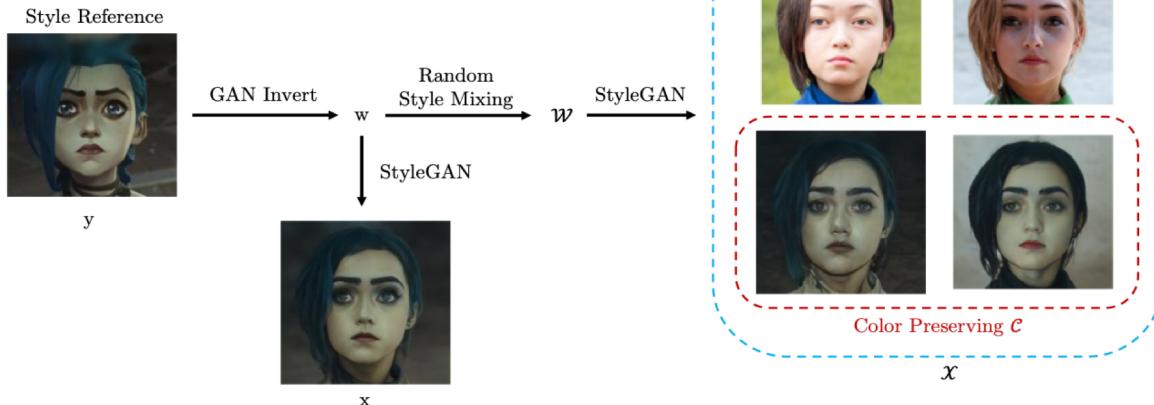
- Générateur

- CNN qui génère une image

- Discriminateur

- CNN qui apprend à partir d'images réelles si ce que le générateur a produit est vrai ou faux

Algorithme



Inversion -> code de style w

- Inversion de l'image de référence
- Inversion de l'image de style
- Génération d'échantillons de style W sur certaines couches du réseau



famille de style W -> Entrainement StyleGAN

- W génère une famille d'images de visages réels correspondant à y
- Affinage de paramétrage du styleGAN (itération, alpha, preservation couleur)
- Itération sur une/plusieurs image(s) de style



Génération

- Génération à partir du StyleGAN affiné

Test - Résultats



- Environnement: Colab pro (GPU)
- Temps moyen: 45s
 - Inversion Référence : 8s
 - Inversion Style : 7s
 - Entrainement Style : 27s
 - Génération : 3s



Conclusion

- Les +
 - Exécution très rapide (hors temps de chargement de modèle)
 - Très bonne transposition des différents styles
 - Mise en œuvre concrète d'un GAN
- Les -
 - Réutilisation de code
 - Métriques baseline
 - Peu familier avec PyTorch
- Pour la suite
 - Passer le code sous TensorFlow