

Rapport Projet IMA206



Édition d'images de visages par manipulation de codes latents

Enguerrand Paquin
Chloé Court
Alexandre Heymann
Ivan Khodakov

Gwilherm Lesné
Yann Gousseau
Juin 2024

Table des matières

1	Exploration et application de StyleGAN	2
1.	1. Introduction.....	2
1.	1. générateur StyleGAN	2
1.	1. qualité des images générées	3
2.	2. Stylemixing.....	4
1.	1. Décorrelation.....	5
3.	3. Déplacement dans W.....	6
1.	1.	6
2.	4. Implémentation de GANSpace	8
1.	1. Introduction	8
1.	5. Notre implémentation	10
3.	2. Implémentation de InterFaceGAN	10
3.	6. Introduction.....	10
1.	1. implémentation	11
1.	3. corrélation des caractéristiques	15
4.	2. Manipulations supplémentaires	15
4.	4. Incorporation d'un Visage et Altération de sa Nature (IVAN)	16
3.	3. Création d'une image contenant des attributs choisis	16
4.	4.	
2		

1. Exploration et application de StyleGAN

1.1 Introduction

L'article "A Style-Based Generator Architecture for Generative Adversarial Networks" de Tero Kar- ras, Samuli Laine et Timo Aila de NVIDIA propose une nouvelle architecture de générateur pour les réseaux antagonistes génératifs (GAN). Cette architecture permet une séparation non supervisée des attributs de haut niveau comme la pose et l'identité. Elle permet aussi des variations stochastiques dans les images générées et offre un contrôle intuitif et spécifique à l'échelle de la synthèse des images. Cela permet d'améliorer la qualité de distribution et les propriétés d'interpolation, ainsi que la décorrélation des facteurs latents de variation.

1.2 Architecture du générateur StyleGAN

StyleGAN apporte des améliorations significatives à l'architecture traditionnelle des GAN. L'une des principales différences réside dans la manière dont le code latent est introduit dans le générateur. Dans l'architecture traditionnelle, le code latent est simplement injecté via une couche d'entrée. En revanche, StyleGAN utilise un réseau de mapping non linéaire pour transformer le code latent z avant de l'introduire dans le générateur. Ce réseau mappe le code latent dans un espace latent intermédiaire, produisant un vecteur w . Ce vecteur w est ensuite décomposé en styles $y = (ys, yb)$, où ys et yb contrôlent les opérations de normalisation d'instance adaptative (AdaIN). AdaIN est appliqué à chaque couche de convolution, permettant de moduler les caractéristiques de l'image en fonction des styles spécifiés. De plus, StyleGAN intègre du bruit gaussien directement dans le réseau, ce qui permet une séparation automatique et non supervisée des attributs de haut niveau et des variations stochastiques.

$$\text{AdaIN}(x_i; y) = ys, i \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (1)$$

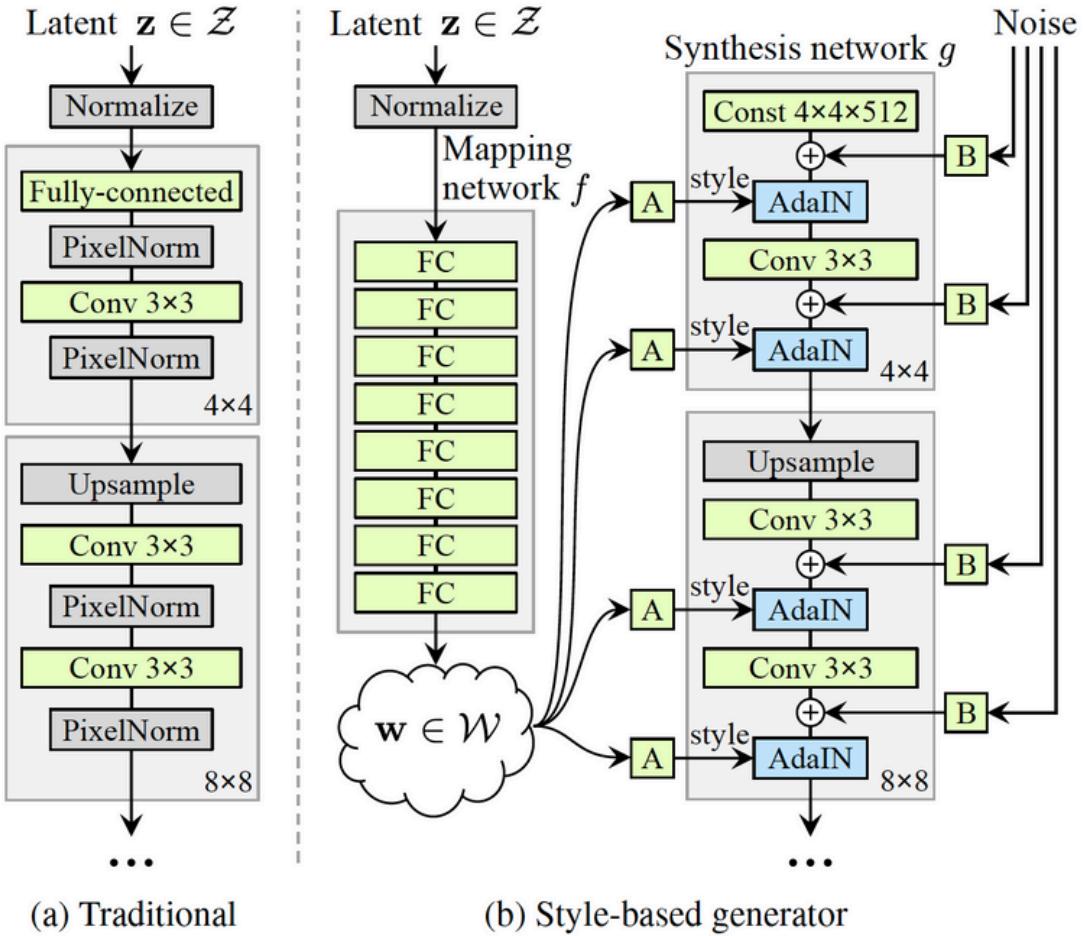


Figure 1 – Architecture de StyleGAN

1.3 Amélioration de la qualité des images générées

Cette nouvelle architecture améliore significativement la qualité des images, mesurée par la Fréchet Inception Distance (FID). Par rapport à un générateur traditionnel, l'architecture StyleGAN réduit les FID sur les datasets CELEBA-HQ et FFHQ (de près de 20%), témoignant d'une meilleure qualité de distribution.

Le dataset FFHQ représente une nette amélioration par rapport à CELEBA-HQ. En effet, il comprend beaucoup plus de variations en termes d'âge, d'appartenance ethnique et de fond. Il couvre également beaucoup mieux les accessoires tels que les lunettes de vue, les lunettes de soleil ou les chapeaux. Le dataset FFHQ a joué un rôle crucial dans l'entraînement de StyleGAN. Il comprend 70 000 images de visages humains de haute qualité et de haute résolution (1024x1024 pixels), sélectionnées à partir de Flickr. Afin de simplifier l'entraînement du modèle, les images ont été filtrées et prétraitées en alignant et en recadrant les visages. L'utilisation de ce dataset a contribué de manière significative à la qualité et à la fidélité visuelle des images générées par StyleGAN. Cela le rend capable de produire des visages synthétiques qui sont presque indiscernables de ceux des vraies personnes.



Figure 2 – Images produites par le générateur StyleGAN avec le dataset FFHQ

1.4 Style mixing

Nous avons utilisé le Style mixing afin de créer des images visuellement intéressantes et variées. Cette méthode consiste à combiner les styles de différentes images pour générer de nouvelles images avec des caractéristiques hybrides. Pour cela, on génère des images à l'aide de deux codes latents aléatoires plutôt qu'un seul lors de l'apprentissage. Nous générerons deux codes latents z_1 et z_2 et les vecteurs w_1 et w_2 correspondants. Nous appliquons w_1 sur les 6 premières couches de style et w_2 sur les autres. En entrant des vecteurs w différents dans les couches de StyleGAN, on constate que le réseau peut toujours générer des images visuellement cohérentes, même si les styles sont mélangés ou interpolés. Ainsi, le Style mixing montre la robustesse et la flexibilité de l'architecture de StyleGAN.



Figure 3 – 4 images générées à partir d'un premier code latent (première ligne) et 5 images générées à partir d'un second code latent (première colonne); le reste des images a été généré en copiant les 6 premiers styles de la première source sur les images de la colonne.

1.5 Variation stochastique

De nombreuses caractéristiques d'un visage humain peuvent être considérées comme stochastiques, telles que l'emplacement exact des cheveux ou des poils. Ces éléments peuvent être générés aléatoirement sans affecter le réalisme de l'image.

En injectant du bruit gaussien à différentes étapes du réseau, StyleGAN introduit une variabilité stochastique dans le processus de génération d'images. Cette variabilité imite les imperfections et les variations naturelles présentes dans les images réelles, rendant ainsi les images générées plus réalistes et moins artificielles.



Figure 4 – Effet du bruit sur une image générée; à gauche avec du bruit et à droite sans.

Ici, on constate que le bruit n'affecte que les aspects stochastiques, donc les hautes fréquences, laissant intacts la composition globale et les aspects de haut niveau. L'image bruitée est nettement plus réaliste.

1.6 Décorrélation

L'objectif de la décorrélation est d'obtenir un espace latent composé de sous-espaces linéaires, chacun d'eux contrôlant un facteur de variation.

Un des avantages principaux de l'architecture de StyleGAN réside dans le mapping effectué dans W . En effet, dans l'espace latent Z les facteurs de variations sont entrelacés. Cela est dû au fait que cet espace doit respecter la proportion de ces facteurs dans la donnée d'entraînement. Ceci crée des zones qualifiées de "repliées" dans Z . Or, le mapping permet de déformer Z en W afin que les facteurs de variation deviennent plus linéaires. L'hypothèse avancée par l'article est que lors de l'entraînement, le générateur induit cette déformation car elle aide à générer des images plus réalistes.

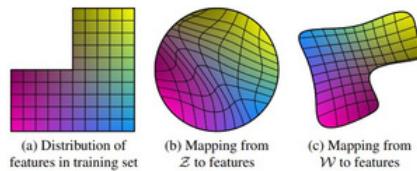


Figure 5 – Interprétation de la déformation induite par le mapping

1.7 Déplacement dans W

Nous avons exploré l'espace latent W et visualisé l'impact de la modification des vecteurs w sur les images générées. Dans la Fig 6, nous nous déplaçons selon les 5 premières composantes de l'espace W avec un pas de 2. Nous observons des modifications sémantiques progressives.



Figure 6 – Effet du déplacement selon les 5 premières composantes de W sur une image générée

Nous pouvons voir que dans l'espace W , modifier une dimension de l'espace revient à modifier des caractéristiques sémantiques de l'image distinctes même s'il n'est pas possible de prédire quelle caractéristique sera modifiée à l'avance. C'est une preuve de la décorrélation globale de cet espace.

2. Implémentation de GANSpace

2.1 Introduction

Nous avons commencé par l'implémentation de l'article GANSpace pour avoir une première idée de la géométrie de l'espace de style. Après avoir compris l'idée derrière chaque article, l'utilisation d'une PCA sur l'espace de style semblait être l'approche la plus simple.

2.2 Notre implémentation

Nous avons décidé d'appliquer une PCA sur les 5 premiers composantes de l'espace de style en l'appliquant sur les 18 canaux de l'architecture.



Figure 7 – Etude des 5 premières composantes de la PCA

A chaque ligne i, Nous sommes à $w(\text{initial})$ la ième composante de la PCA multiplié par un pas et un scalaire allant de -2 à 2 (Fig 7). Nous observons que les principales modifications sémantiques des 5 premières directions de la PCA concernent la position de la tête, le genre et l'âge.

D'après l'article, seules les 20 premières composantes de la PCA ont une sémantique apparente sur l'image. Pour le vérifier, nous nous sommes déplacés selon les composantes 200 à 205.



Figure 8 – Etudes des composantes 200 à 205 de la PCA

Effectivement nous n'observons pas de fortes modifications sur les images quelque soit la taille du pas choisi (Fig 8).

3. Implémentation de InterFaceGAN

3.1 Introduction

Nous avons ensuite implémenté InterFaceGAN. L'article propose de trouver les vecteurs sémantiques des espaces latents des GANs pour éditer des images. La méthode consiste en pratique à trouver le vecteur directeur de l'hyperplan séparant les sémantiques grâce à un SVM.

3.2 Notre implémentation

L'espace de style étant plus décorrélé que l'espace latent, nous avons commencé par appliquer la méthode sur W avant de l'appliquer sur Z.

Afin de séparer l'espace selon une sémantique recherchée, il est nécessaire de classifier les images générées par StyleGAN pour déterminer si elles possèdent ou non un attribut donné. Pour cela, nous avons utilisé un classifieur pour chaque attribut du dataset CelebA. Ces attributs sont par exemple : le sexe, la présence de lunettes, la présence de maquillage, etc...

Nous avons alors généré 5000 points dans l'espace W, et pour chaque point, l'image associée et son score de classification selon chaque attribut. Ce score est une valeur entre 0 et 1 indiquant la confiance du classifieur sur la présence d'un attribut dans l'image. Nous avons gardé 5 % des points avec le plus haut score et 5% avec le plus bas score pour un attribut donné. Puis, nous avons appliqué un SVR sur ces 250 points, et récupérer le vecteur directeur de l'hyperplan que l'on appelle "vecteur sémantique". En se déplaçant dans la direction du vecteur sémantique sur de nouvelles images, nous sommes parvenus à ajouter des lunettes aux visages (Fig 9).

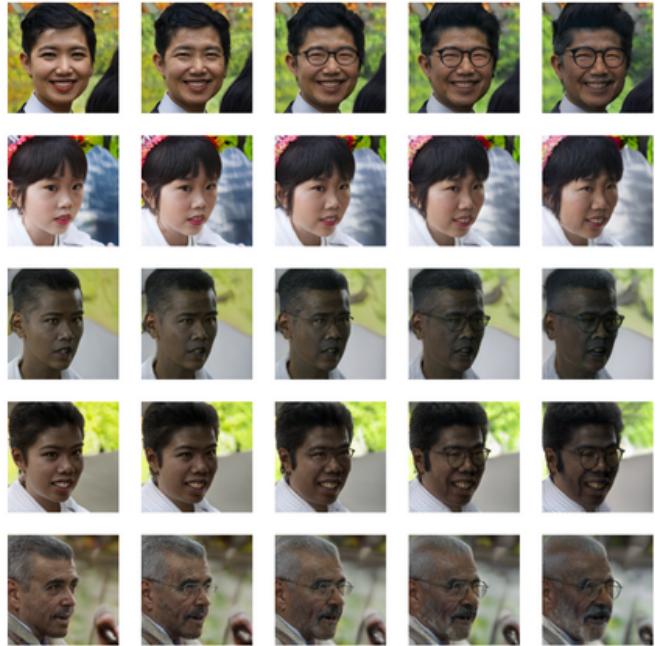


Figure 9 – Ajout de lunettes

3.3 Décorrélation des caractéristiques

Nous avons ensuite appliqué la même méthode à l'espace latent. L'espace latent étant beaucoup plus corrélé que l'espace de style, nous nous attendions à l'apparition d'autres attributs en même temps que l'ajout de lunettes.

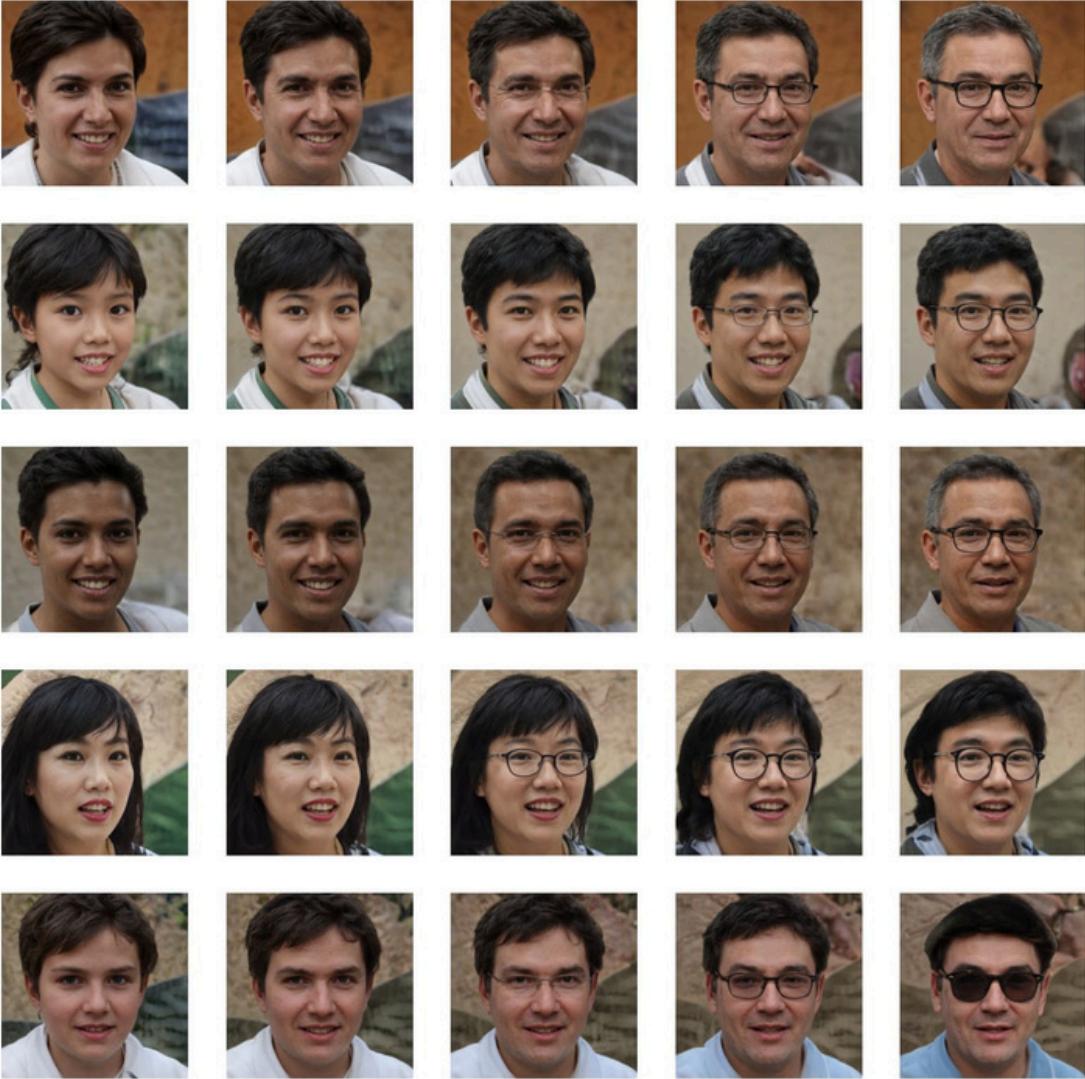


Figure 10 – Ajout de lunettes dans l'espace Z

Nous constatons en effet dans la Fig 10 que l'ajout de lunette va de pair avec le vieillissement et la masculinisation des visages. Nous en déduisons que les attributs 'Eyeglasses', 'Male' et 'Gray_Hair' du classificateur de CelebA sont corrélés. Nous avons donc décorrélés le vecteur sémantique de 'Eyeglasses' en lui soustrayant ses projections sur 'Male' et 'Gray_Hair'.

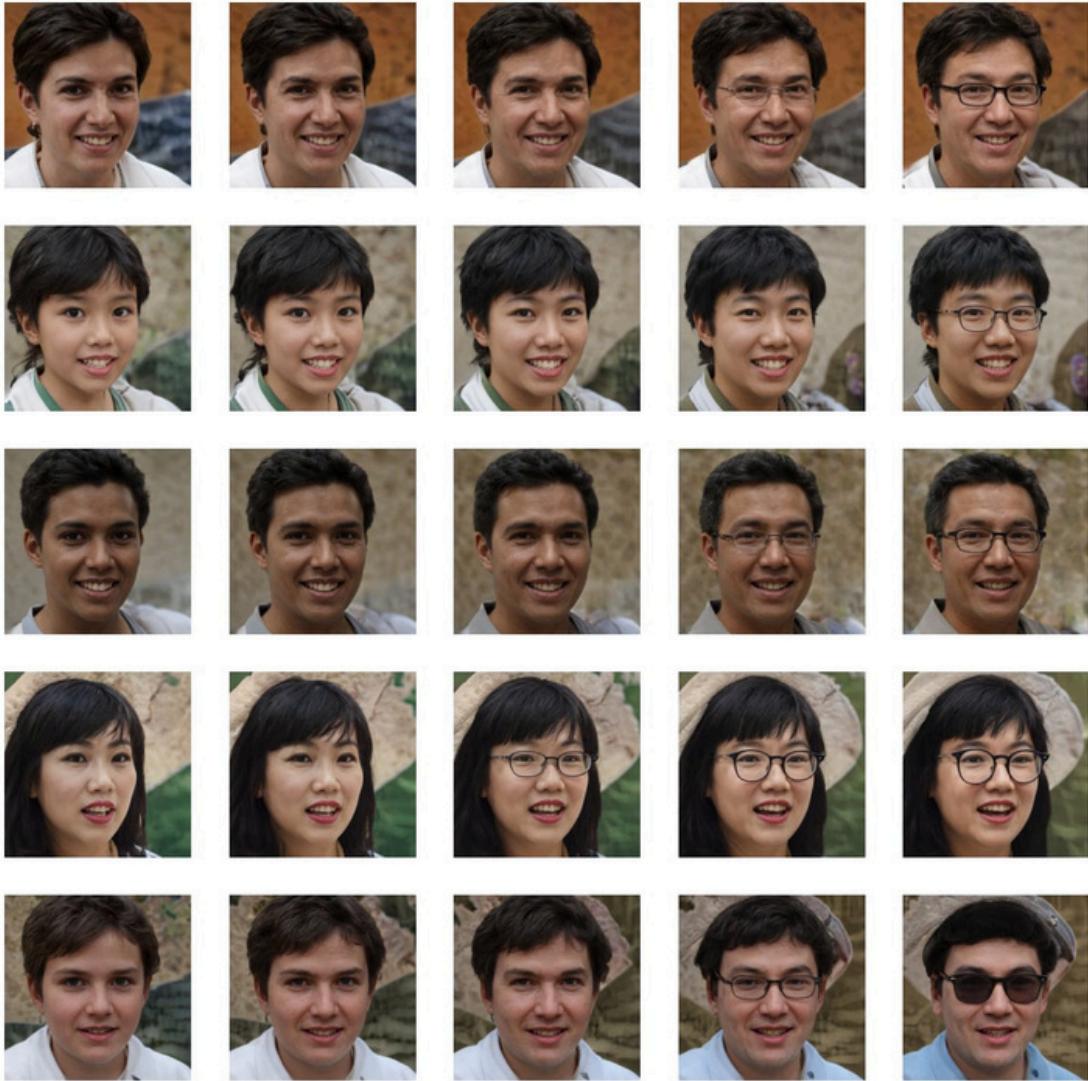


Figure 11 – Ajout de lunettes selon le vecteur décorrélé dans l'espace Z

Les résultats sont concluants. Même si la composante 'Male' de 'Eyeglasses' n'a pas tout à fait disparu, on constate une nette amélioration (Fig 11).

Nous avons enfin pensé à calculer la matrice de corrélation de chaque vecteur sémantique pour observer la différence de corrélation entre l'espace latent et celui de style.

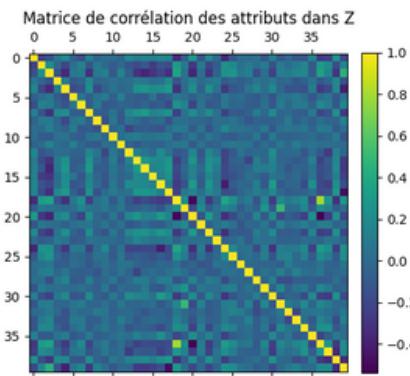


Figure 12 – Matrice de décorrélation de l'espace latent

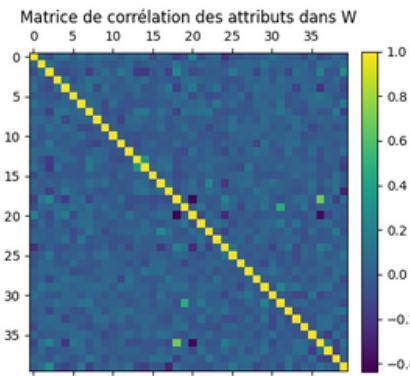


Figure 13 – Matrice de décorrélation de l'espace de style

Ces matrices sont obtenues en calculant la cosine similarity entre les vecteurs directeurs de chaque attribut, respectivement dans l'espace Z (Fig 12) et dans l'espace de style (Fig 13)

4. Manipulations supplémentaires

4.1 Incorporation d'un Visage et Altération de sa Nature (IVAN)

Notre objectif dans cette partie est de pouvoir modifier une image d'un visage réel. La première étape est de trouver le vecteur de w correspondant à une image donnée. Il s'agit ici d'effectuer une projection dans l'espace W . Pour faire cela, nous avons utilisé le fichier `projector.py` du GitHub `stylegan2-ada-pytorch`. Afin d'améliorer la projection, nous avons aligner le visage d'Ivan à la manière des visages dans FFHQ.

Le code utilise un processus d'optimisation pour ajuster cette représentation jusqu'à ce que l'image générée ressemble le plus possible à l'image cible. Dans un premier temps, on calcule la moyenne et l'écart-type de w grâce à un tirage. Ensuite, un détecteur de caractéristiques VGG16 est configuré pour extraire les caractéristiques de l'image cible (Ivan). Enfin, un optimiseur Adam est utilisé pour minimiser la distance euclidienne entre les caractéristiques de l'image synthétisée et celles de l'image cible, tout en régularisant le bruit.



Figure 14 – Image obtenue après projection dans W

Une fois le vecteur w obtenu, nous pouvons nous amuser à changer ses composantes pour modifier les attributs de l'image. Ainsi, en utilisant les directions données par InterFaceGAN, on obtient ces images :



Figure 15 – Déplacements suivant des vecteurs obtenus via InterFaceGAN

Nous constatons que la méthode permet en effet d'ajouter ou d'enlever des attributs. Cependant, les images obtenues s'éloignent un peu de l'image d'origine : on ne reconnaît plus aussi bien Ivan. Cette manipulation ne permet pas de modifier efficacement une image.

4.2 Crédation d'une image contenant des attributs choisis

Notre second but est de créer une image contenant des attributs choisis au préalable. Pour faire cela, nous voulons premièrement nous placer dans une zone qui réagirait bien aux différents déplacement que l'on souhaiterait faire. Il nous a paru pratique de se placer au centre de l'espace W , afin de pouvoir se déplacer dans toutes les directions sans trop s'éloigner de visages réalistes. Nous prenons donc comme vecteur de départ le visage moyen dans W . Il s'agit d'un visage androgyne, sans caractéristique particulière.



Figure 16 – Visage moyen de W , calculé avec 5000 tirages

En partant de cette image, nous ajoutons une des directions obtenue par InterFaceGAN, puis nous tirons un point aléatoire (suivant une distribution normale) dans cette région (Fig 17).

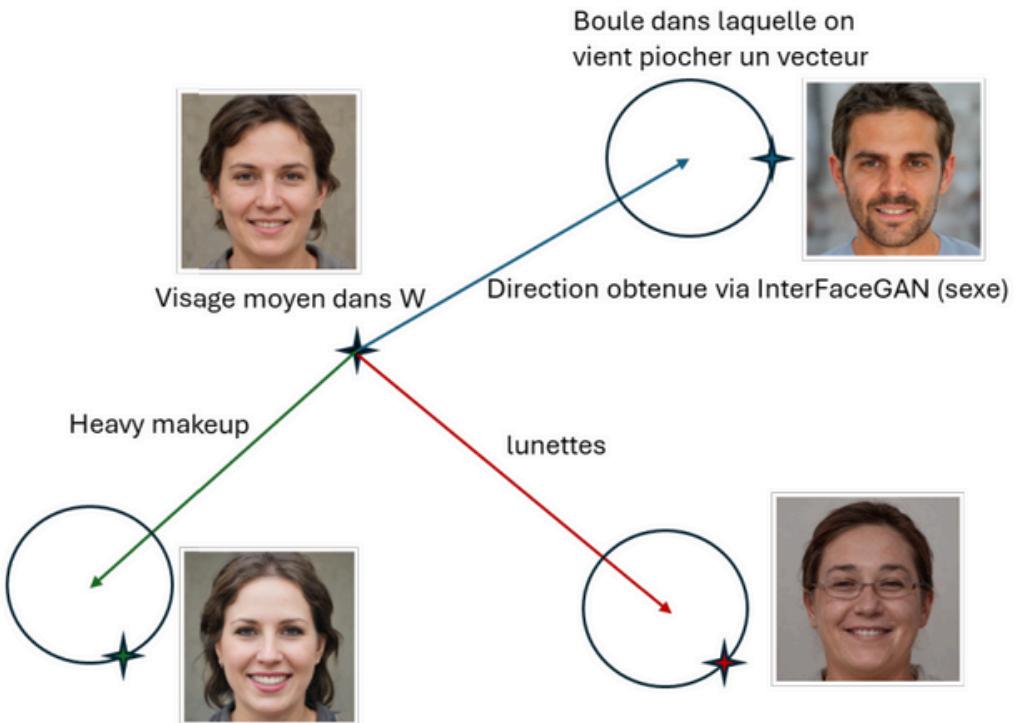


Figure 17 – 1ère méthode

Cependant, nous voyons que les images obtenues sont parfois encore semblables au visage moyen (nez, sourire...), pour éviter ce problème nous essayons une seconde méthode.

La seconde méthode que nous avons testé utilise des classificateurs. Pour une feature choisie, nous sélectionnons les vecteurs de W ayant le meilleur score pour le classifieur associé. Puis, nous calculons le barycentre de ce nuage de point. Nous faisons de même pour l'ensemble des attributs du dataset CelebA. Nous appliquons ensuite un algorithme de PCA sur l'ensemble de ces barycentres et nous projetons nos vecteurs selon les deux premières directions de la PCA. Cela nous permet de plot une "cartographie" en deux dimensions de l'espace W (Fig 18). Nous pouvons voir que l'attribut 35 correspondant au fait que le visage possède un chapeau est très éloigné du reste des points. Cela est expliquable par le fait que c'est un attribut facile à détecter et qu'il est très différent du reste des attributs de CelebA. Cela montre la cohérence de l'espace W .

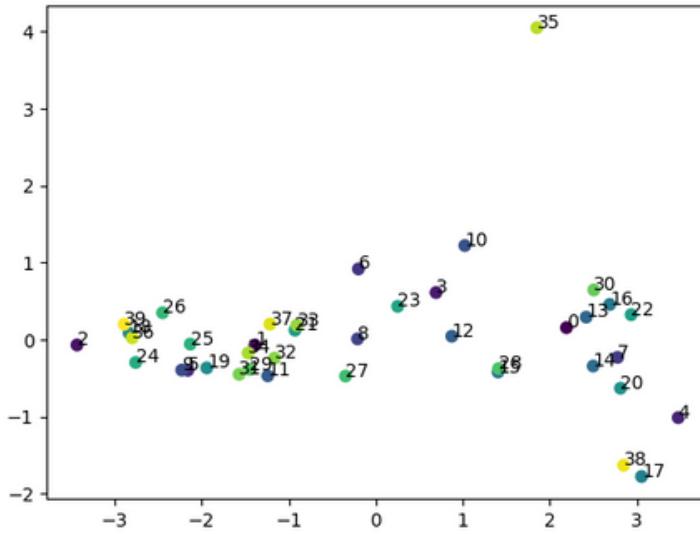


Figure 18 – 2ème méthode : Cartographie 2D de W

En sélectionnant le barycentre correspondant à un attribut donné, il est possible de lui ajouter un vecteur gaussien aléatoire de moyenne nulle et de faible variance. Le vecteur de l'espace W obtenu correspond alors à un visage contenant l'attribut voulu. Il est donc possible de tirer un point de l'espace W et de connaître à l'avance l'attribut principal que le visage final possèdera. Cette méthode permet d'ajouter un contrôle supplémentaire sur les visages générés par StyleGAN comparé à la méthode InterFaceGAN.



Figure 19 – 2ème méthode : Visages générés (visage avec des lunettes, visage d'homme)

5. Conclusion

Lors de ce projet, nous avons dans un premier temps exploré et compris le fonctionnement de StyleGAN. Puis, nous avons implémenté les méthodes GANSpace et InterFaceGAN. Nous avons pu notamment étudié la decorrélation de l'espace W comparé à l'espace Z de StyleGan. Enfin, nous avons exploré deux méthodes : la première permet de modifier le visage d'une image réelle en calculant sa projection dans l'espace W et la seconde permet de cartographier l'espace W selon les attributs du dataset CelebA et de tirer des points de l'espace W dont on est sûr que le visage qui leur est associé possède une certaine caractéristique.