Université de Gafsa

Institut Supérieur des Sciences Appliquées et de Technologie de GAFSA

Département Informatique et télécommunication

Titre

Simulation d'un Smile design basée sur les GANs

Présenté et soutenu par :

Sliman Firas

En vue de l'obtention d'un

Master Professionnel en sciences des données

Sous la Direction de :

Maitre-assistant Rekik Ahmed

Soutenu le 00/00/2023

Devant le jury composé de :

Président :

Rapporteur :

Membres :

2022/2023

Dédicaces :

A mes chers parents

Vous êtes mes plus grands héros et mes sources d'inspiration infinies. Votre soutien inconditionnel, vos sacrifices et votre confiance en moi ont été les fondations solides sur lesquelles j'ai pu construire mes aspirations académiques. Votre amour incommensurable m'a donné la force de persévérer lorsque les défis semblaient insurmontables. Je vous suis éternellement reconnaissant pour tout ce que vous avez fait pour moi.

A mes très Chères sœur et frère.

Vous avez été mes complices, mes confidents et mes partenaires de vie. Vos encouragements constants, vos conseils avisés et votre soutien indéfectible ont été d'une importance capitale pour moi.

Notre lien fraternel est une force qui m'a permis de persévérer dans les moments difficiles et de célébrer ensemble les victoires.

A mes amis proches qui m'ont soutenu tout au long de ce travail.

À tous mes professeurs qui m'ont enseigné.

Puisse ce modeste travail vous exprimer ma profonde reconnaissance,

Mon Respect et mon Admiration sans limites à votre égard.

Remerciement :

Tout d’abord, ce travail ne serait pas aussi riche et n’aurait pas pu voir le jour sans l’aide et l’encadrement de Mr **Ahmed REKIK**.

Par le biais de ces mots, je tiens à exprimer ma profonde gratitude et mes sincères remerciements pour votre inestimable soutien et votre encadrement exceptionnel tout au long de mon parcours de recherche dans le cadre de mon mémoire de Master.

Votre expertise et vos connaissances approfondies ont été une source inépuisable d'inspiration pour moi. Votre dévouement à la recherche, votre passion pour l'apprentissage et votre patience infinie ont joué un rôle crucial dans le succès de ce projet. Grâce à vos conseils avisés, j'ai pu acquérir une meilleure compréhension du sujet, affiner ma méthodologie de recherche et approfondir mes connaissances dans le domaine.

Je tiens également à souligner votre disponibilité constante pour répondre à mes questions, discuter de mes idées et fournir des commentaires constructifs. Vos remarques perspicaces et votre capacité à me guider avec tact ont grandement amélioré la qualité de mon travail. Vous avez été un mentor exemplaire, m'encourageant à repousser mes limites et à viser l'excellence.

Je tiens à exprimer ma profonde gratitude envers ma professeur Mme **Fatma HRIZI** ainsi qu'envers mes professeurs.

Votre enseignement de qualité, votre passion pour la recherche et votre dévouement envers nos études ont joué un rôle essentiel dans ma formation académique et professionnelle. Votre soutien constant et vos précieux conseils m'ont permis de repousser mes limites et de réaliser mon plein potentiel.

Enfin, Je tiens à adresser mes remerciements à toutes les personnes qui ont

Contribué de près ou de loin à la réalisation de ce projet.

Glossaire

Smile Design : Le Smile Design (ou Design du sourire) est un processus utilisé en dentisterie esthétique pour planifier et créer un sourire harmonieux et attrayant. Il s'agit d'une approche personnalisée qui prend en compte les caractéristiques faciales, la forme des dents, la couleur, l'alignement et d'autres aspects esthétiques pour concevoir un sourire qui correspond aux préférences et aux besoins du patient. Le Smile Design peut impliquer l'utilisation de techniques telles que la correction orthodontique, la pose de facettes dentaires, la blanchiment des dents, etc., afin d'obtenir un résultat esthétique optimal pour le sourire du patient.

IA : L'Intelligence Artificielle est un domaine de l'informatique qui vise à développer des systèmes capables de réaliser des tâches nécessitant normalement l'intelligence humaine, telles que la reconnaissance vocale, la vision par ordinateur, la prise de décision, etc.

GANs : Generative Adversarial Networks, Les Réseaux Adversaires Génératifs sont une architecture de réseau de neurones artificiels utilisée dans l'apprentissage automatique. Les GANs sont composés de deux parties, un générateur et un discriminateur, qui s'affrontent pour améliorer la génération de données réalistes.

CGAN : Le Conditional Generative Adversarial Networks, CGAN est une variation des GANs où le générateur et le discriminateur sont conditionnés à des informations supplémentaires, telles que des étiquettes de classe, pour générer des données spécifiques.

CNN : Le Convolutional Neural Network est un type de réseau de neurones artificiels couramment utilisé pour l'analyse d'images. Les CNN utilisent des opérations de convolution pour extraire des caractéristiques des images, ce qui les rend efficaces pour des tâches telles que la classification d'images, la détection d'objets, etc.

RNN : Le Recurrent Neural Network est un type de réseau de neurones qui est conçu pour traiter des données séquentielles, telles que des séquences de mots ou des séries temporelles. Les RNN utilisent des boucles récurrentes pour conserver et exploiter l'information du passé lors de la prise de décision.

DNN : Le Deep Neural Network est un réseau de neurones artificiels avec plusieurs couches intermédiaires (appelées couches cachées). Les DNN sont utilisés pour des tâches complexes, telles que la reconnaissance d'images, la traduction automatique, etc.

KL : La divergence de Kullback-Leibler (Kullback-Leibler divergence en anglais) est une mesure utilisée en théorie de l'information pour quantifier la différence entre deux distributions de probabilité.

BCE : Binary Cross-Entropy est une fonction de perte couramment utilisée dans l'apprentissage automatique pour les tâches de classification binaire. Elle mesure la différence entre les prédictions d'un modèle et les étiquettes réelles en utilisant la fonction d'entropie croisée.

UNet : U-shaped Network est une architecture de réseau de neurones convolutifs utilisée pour la segmentation sémantique des images médicales. Il est célèbre pour sa capacité à effectuer une segmentation précise en utilisant une architecture en forme de U.

Faster R-CNN : Faster Region-CNN est un modèle de détection d'objets utilisant des réseaux neuronaux convolutifs. Il combine une région de propositions pour localiser les objets dans une image et une classification pour identifier les classes des objets détectés.

Mask R-CNN : Mask Region-CNN est une extension du modèle Faster R-CNN qui permet également de segmenter les objets détectés par des masques précis.

API : Une Interface de Programmation Applicative est un ensemble de définitions et de protocoles qui permettent à à des logiciels différents de communiquer et d'interagir entre eux.

Résumé

Ce mémoire de projet de fin d'études se concentre sur l'utilisation des réseaux antagonistes génératifs (GAN) de type Pix2Pix pour la génération automatique du Smile Design dans le domaine de la dentisterie esthétique. Le travail de recherche vise à explorer comment les GAN Pix2Pix peuvent être utilisés pour générer des images de sourire réalistes à partir de données d'entrée, telles que des images de références et des paramètres de conception souhaités. L'étude présente une méthodologie détaillée pour l'entraînement du modèle Pix2Pix spécifique au Smile Design, ainsi que des expérimentations pour évaluer les performances et la qualité des résultats générés. Les résultats obtenus démontrent que les GAN Pix2Pix peuvent être une approche prometteuse pour la génération automatisée du Smile Design, offrant un potentiel d'amélioration des processus de conception et de prise de décision en dentisterie esthétique.

Abstract

This final year project thesis focuses on the use of Pix2Pix generative adversarial networks (GANs) for automatic Smile Design generation in the field of aesthetic dentistry. The research work aims to explore how Pix2Pix GANs can be utilized to generate realistic Smile images from input data, such as reference dataset images and desired Design parameters. The study presents a detailed methodology for training the specific Smile Design Pix2Pix modèle , along with experimentation to evaluate the performance and quality of the generated results. The obtained results demonstrate that Pix2Pix GANs can be a promising approach for automated Smile Design generation, offering potential improvements in the design and decision-making processes in aesthetic dentistry.

Listes des figues :

Table des matières

Introduction générale

Le sourire est un élément essentiel de notre apparence et de notre expression personnelle. Il joue un rôle crucial dans notre communication quotidienne, tant sur le plan personnel que professionnel. Le "Smile Design" est un domaine de la dentisterie esthétique qui se concentre sur l'amélioration de l'apparence du sourire en prenant en compte plusieurs éléments, tels que la forme, la couleur, l'alignement et la texture des dents.

Le Smile Design vise à créer un sourire harmonieux et attrayant, tout en prenant en considération les traits faciaux uniques de chaque individu. Il s'agit d'une approche personnalisée qui tient compte des souhaits et des préférences du patient, ainsi que des aspects anatomiques et esthétiques de son visage.

Le processus de Smile Design commence généralement par une évaluation complète de l'état actuel du sourire du patient. Ensuite, en collaboration avec le patient, le dentiste élaborera un plan de traitement personnalisé pour améliorer l'apparence du sourire.

L'avancée des technologies dentaires permet aux patients de visualiser les résultats potentiels du Smile Design avant de commencer le traitement. Cela permet une meilleure communication entre le dentiste et le patient, et assure une compréhension mutuelle des attentes et des résultats attendus.

Le Smile Design et l'intelligence artificielle (IA) peuvent être combinés pour créer des résultats encore plus précis et personnalisés. L'IA est de plus en plus utilisée dans le domaine dentaire pour aider les dentistes à planifier et à réaliser des traitements de Smile Design de manière plus efficace. L'intégration de l'IA dans le Smile Design offre des avantages significatifs en termes de visualisation, de planification et de personnalisation des traitements. Cela permet aux dentistes de fournir des résultats plus précis et aux patients de participer activement au processus de prise de décision. L'IA ouvre de nouvelles perspectives passionnantes pour l'avenir du Smile Design en offrant des solutions plus avancées et plus satisfaisantes pour les patients.

Dans le cadre de ce travail, nous allons explorer l’utilisation des réseaux antagonistes génératifs (GANs) pour la génération automatique d’un Smile Design réaliste. Les GANs sont une forme d'intelligence artificielle qui peut être utilisée dans le domaine du Smile Design pour générer des images réalistes de sourires améliorés et personnalisés. Les GANs sont composés

de deux parties : un générateur et un discriminateur, qui travaillent en tandem pour produire des résultats optimaux.

Dans le contexte du Smile Design, les GANs peuvent être entraînés à partir de grandes bases de données d'images de sourires pour apprendre les caractéristiques esthétiques et les variations naturelles des sourires. Le générateur du GAN peut ensuite être utilisé pour créer de nouvelles images de sourires en modifiant des paramètres spécifiques tels que la forme, la couleur ou l'alignement des dents.

Il convient de noter que les GANs pour le Smile Design sont encore une technologie en développement et que leur utilisation actuelle est limitée. Cependant, ils offrent un potentiel intéressant pour l'amélioration de la planification et de la visualisation des résultats de Smile Design, ce qui peut contribuer à des décisions plus informées et à des résultats plus satisfaisants pour les patients.

Il est important de souligner que, malgré les avancées technologiques telles que l'utilisation des GANs, l'expertise et l'expérience des dentistes restent essentielles pour évaluer et comprendre les besoins spécifiques des patients, ainsi que pour réaliser les traitements de Smile Design de manière sûre et efficace.

Pour cela nous avons structuré notre travail en trois grands chapitres qui sont les suivants :

- Dans le premier chapitre, nous aborderons le contexte général de notre étude, en expliquant le contexte dans lequel notre travail s'inscrit ainsi que les enjeux et les objectifs visés.
- Dans le deuxième chapitre, nous nous concentrerons sur la génération automatique de sourires. Nous explorerons en détail les techniques de GAN, CGAN et Pix2Pix qui ont été utilisées dans notre travail, en expliquant leurs fonctionnements respectifs.
- Enfin, dans le troisième et dernier chapitre, nous présenterons les résultats de nos expérimentations et les conclusions qui ont été tirées de nos observations.

1. Chapitre 1 Contexte et Problématique

1.1. Introduction

Un sourire sain et éclatant peut avoir un impact significatif sur notre bien-être émotionnel et notre qualité de vie. Cependant, pour certaines personnes, le sourire peut être une source de complexe et de gêne. Les défauts dentaires tels que les dents manquantes, la dentition mal alignée ou les dents tachées peuvent affecter la confiance en soi et l'estime de soi. C'est pourquoi la dentisterie esthétique est devenue une discipline importante qui vise à aider les patients à atteindre leur sourire idéal en prenant en compte leurs préférences et leurs besoins individuels. Cependant, la conception de sourires personnalisés reste un processus complexe qui nécessite une grande expertise et une compréhension approfondie des caractéristiques faciales et dentaires individuelles. C'est pourquoi l'utilisation de modèles génératifs GANs peut être un outil précieux pour aider les professionnels du Smile Design à créer des sourires attrayants et personnalisés pour chaque patient. En combinant les connaissances et l'expertise des professionnels du sourire avec les capacités des modèles génératifs, nous pouvons améliorer les résultats pour les patients et les aider à atteindre leur sourire de rêve.

1.2. Contexte général

Au début, les dentistes dessinaient à la main des croquis des dents pour simuler un "Smile Design" en utilisant une variété de techniques artistiques. Tout d'abord, ils peuvent prendre des photographies de la bouche du patient et les dessiner à l'échelle pour créer une représentation précise de l'alignement et de la forme des dents. Ensuite, ils peuvent utiliser des outils tels que des marqueurs de couleur pour dessiner des lignes qui montrent où les dents devraient être allongées, raccourcies ou repositionnées pour créer un sourire plus harmonieux. Enfin, ils peuvent utiliser leur expérience et leur expertise pour ajuster les dessins et créer une simulation visuelle proche du réel et du résultat final attendu.

Avec le progrès technologique, les dentistes peuvent utiliser la technologie des images en deux dimensions pour simuler un sourire en utilisant des logiciels de retouche d'images tels que Photoshop comme l’illustre la figure 1. Ils peuvent prendre une photo du sourire du patient et l'importer dans le logiciel pour modifier numériquement l'apparence des dents. Les dentistes peuvent utiliser des outils tels que la correction de couleur, le lissage des imperfections, le redimensionnement des dents et le réalignement des dents pour simuler le résultat final. Cette

approche est particulièrement utile pour les patients qui veulent avoir une idée rapide et facile du résultat possible de leur traitement. Bien que cette approche ne soit pas aussi précise, elle n'est pas rapide mais elle peut fournir une idée générale du résultat possible. Cependant, elle nécessite la maîtrise des logiciels de traitement d'images tels que Photoshop. Les dentistes doivent être prudents lors de l'utilisation de cette technologie, car elle peut parfois être trompeuse et donner des attentes irréalistes au patient

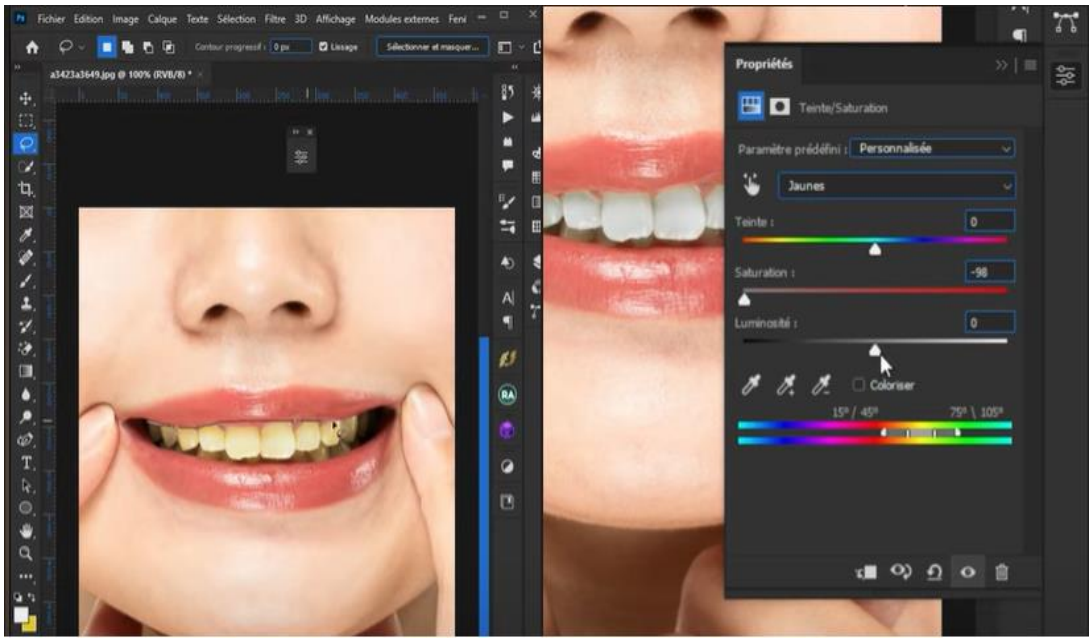

Figure 1simulation d'un sourire avec photoshop

Les dentistes peuvent simuler un sourire en superposant deux images, une image des dents superposée sur le sourire du patient (figure 2 et 3). Pour ce faire, ils peuvent prendre une photographie de la bouche du patient et une autre photographie des dents de référence, qui peut être une photo de dents de célébrités ou d'un sourire idéal. Ensuite, ils peuvent importer ces deux images dans un logiciel de conception assistée par ordinateur et les superposer. Les dentistes peuvent ensuite utiliser des outils de manipulation d'images pour ajuster la taille, la forme et la position des dents de référence afin de les faire correspondre à la bouche du patient. Cela permet aux dentistes et aux patients de visualiser de manière plus précise les changements proposés et de discuter des options de traitement avec une compréhension plus claire du résultat final. Bien que cette méthode ne soit pas aussi précise, elle peut fournir une idée générale du résultat possible et aider les patients à mieux comprendre leur traitement.

Figure 2: sourire simulé par superposition d'images

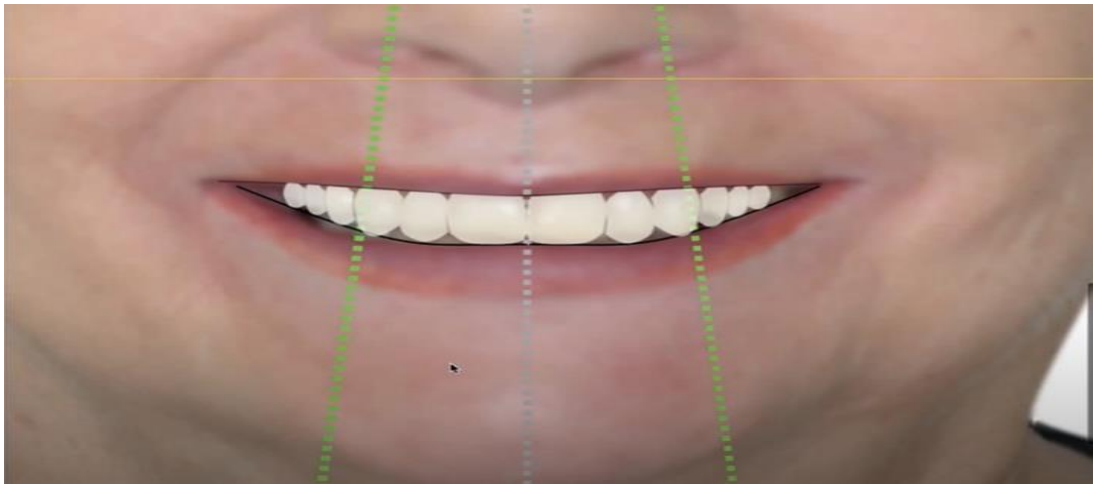

Figure 3 sourire simulé par superposition de facette dentaire

Les dentistes utilisent aussi une technique appelée Mock-up dentaires pour simuler un Smile Design comme le montre la figure 4. Le mock-up dentaire est une technique utilisée par les dentistes pour créer une maquette en cire ou en résine représentant le sourire souhaité. Cette maquette est temporairement placée sur les dents du patient, lui permettant ainsi de visualiser et d'évaluer l'apparence potentielle du résultat final avant de procéder à des traitements dentaires esthétiques. Cela aide à la communication entre le dentiste et le patient, permettant de définir les attentes et d'apporter d'éventuelles modifications avant de passer aux traitements permanents.

Figure 4 simulation d'un sourire par Mock-up

On trouve également la techniques des facettes dentaires temporaires qui sont des revêtements temporaires en résine ou en composite qui sont appliqués sur les dents du patient pour simuler un "Smile Design" ou un Design de sourire souhaité. Cette technique permet au dentiste de créer une transformation esthétique provisoire et de montrer au patient à quoi ressemblerait son sourire final avant de procéder à des facettes dentaires permanentes.

Les facettes temporaires sont réalisées en prenant des empreintes des dents du patient, puis en utilisant ces empreintes pour fabriquer des facettes en résine ou en composite qui sont ensuite fixées sur les dents préparées. Les facettes temporaires permettent de modifier la taille, la forme, la couleur et l'alignement des dents, offrant ainsi une vision concrète du résultat final attendu.

Ces facettes temporaires illustré par la figure 5 restent en place pendant une période déterminée, généralement quelques semaines, afin que le patient puisse s'habituer à son nouveau sourire et donner son avis sur l'apparence et le confort des facettes. Si le patient est satisfait du résultat, le dentiste procédera ensuite à la fabrication et à la pose des facettes dentaires permanentes en utilisant des matériaux plus durables tels que la porcelaine.

Figure 5 Avant et après application des facettes dentaires temporaire

Pendant ce stage au sein d'Udini, on travaille dans un laboratoire de recherche et développement de pointe spécialisé dans l'application de l'IA en dentisterie. On bénéficie des capacités avancées en IA, en clinique dentaire et en cloud computing, ce qui en fait le partenaire idéal pour les entreprises cherchant à développer des solutions logicielles médicales dentaires innovantes et automatisées.

Au cours de ce stage, nous allons travailler sur la simulation de sourires, également appelée "Smile Design", en utilisant les technologies de pointe en IA et en dentisterie d 'Udini. Le but de notre travail sera de développer une solution logicielle automatique et innovante pour aider les dentistes à créer des sourires personnalisés pour leurs patients. Nous allons explorer les différentes méthodes et techniques pour la modélisation de sourires en 2D, ainsi nous allons également développer des algorithmes de traitement d'images et d'apprentissage automatique pour automatiser le processus de conception de sourires. Ce stage nous permettra d'acquérir une expérience pratique en utilisant les technologies les plus avancées dans le domaine de la dentisterie et de l'IA pour aider à améliorer la vie des patients dentaires.

1.3. Problématique

Notre problématique concerne les limites actuelles de la technologie en matière de traitement d'images pour la simulation de sourire.

Bien que les dentistes aient accès à des logiciels avancés qui leur permettent de superposer des images de dents sur le sourire d'un patient, le résultat final peut ne pas être satisfaisant en raison de la texture des dents qui n'est pas réaliste. En effet, les dents ont une texture complexe et variée, avec des nuances subtiles et des irrégularités qui reflètent la lumière de manière différente selon l'angle et l'intensité. La technologie actuelle de traitement d'images ne peut pas toujours reproduire cette texture avec la même précision et la même variété, ce qui peut entraîner des résultats insatisfaisants.

Les dentistes doivent donc travailler en étroite collaboration avec leurs patients pour comprendre leurs attentes et leurs préférences en matière de texture et de réalisme.

En fin de compte, pour créer un sourire harmonieux et réaliste, les dentistes doivent continuer à chercher des solutions pour améliorer la précision et la variété de la simulation, en tenant compte des limites actuelles de la technologie de traitement d'images.

1.4. Objectif

Une solution potentielle à la problématique de la génération de textures réelles des dents dans la simulation de sourire est l'utilisation de l'intelligence artificielle et des modèles génératifs d'images. Ces modèles d'IA sont capables d'apprendre à partir de grandes quantités de données d'images dentaires pour créer des textures réalistes de dents. En utilisant des techniques de Machine Learning et des algorithmes de traitement d'images, on peut entraîner ces modèles à comprendre les caractéristiques de la texture des dents, telles que les nuances subtiles, les motifs et les irrégularités, pour générer des textures plus précises et variées. De plus, l'utilisation de ces modèles d'IA peut également aider à améliorer la rapidité et l'efficacité de la simulation de sourire, en réduisant le temps nécessaire pour créer des modèles réalistes et des sourires harmonieux.

1.5. Conclusion

Pour conclure, un sourire confiant peut améliorer l'apparence, la confiance en soi et la qualité de vie d'une personne. Les dentistes utilisent des technologies telles que la simulation numérique pour aider les patients à visualiser le résultat final de leur traitement dentaire, mais la texture des dents peut ne pas toujours être réaliste. Les modèles génératifs d'images basés sur l'IA offrent une solution potentielle à ce problème en utilisant des techniques de Machine Learning pour créer des textures réalistes de dents. Tout en comprenant les limites actuelles de la technologie de traitement d'images pour mieux visualiser la forme et la texture des dents de manière plus organique, réelle et harmonieuse.

2. Chapitre 2 Génération automatique d'un sourire

2.1. Introduction

Dans ce stage, nous allons nous intéresser particulièrement à l'application des GANs en dentisterie pour générer des images de dents réalistes. Nous allons explorer précisément l'architecture de CGANs et leurs applications dans la génération d'images dentaires.

Le but de ce travail est de permettre aux dentistes d'avoir accès à des images de dents réalistes et de haute qualité qui leur permettront de mieux planifier leurs traitements**.**

2.2. Les Réseaux Antagonistes Génératifs

Les réseaux antagonistes génératifs (GAN) [1] sont une avancée récente et fascinante en matière d'apprentissage automatique. Les GAN sont des modèles génératifs, c'est-à-dire qu'ils ont la capacité de créer des instances de données qui ressemblent à celles de l'ensemble d'entraînement. Par exemple, les GAN peuvent produire des images qui ressemblent à des photographies de visages humains, même si ces visages ne sont pas liés à une personne réelle. Ces images dans la figure 6 ont été générées à l'aide d'un GAN :

Figure 6: Images générées par un réseau GAN créé par NVIDIA

2.2.1. Structure des GANs

Un réseau antagoniste génératif (GAN) se compose de deux parties :

- **Le générateur**

Il s'agit d'un modèle de réseau de neurones qui apprend à générer de nouvelles données en apprenant la distribution sous-jacente des données d'entraînement. Le générateur prend en entrée un vecteur de bruit aléatoire et produit une sortie qui doit ressembler à des données réelles. Au fur et à mesure de l'entraînement, le générateur ajuste les poids et les biais de son réseau de neurones pour produire des sorties de plus en plus réalistes. Le but ultime est que le

générateur produise des données synthétiques qui sont indiscernables des données réelles pour le discriminateur.

- **Le discriminateur**

C'est un modèle qui apprend à distinguer les données générées par le générateur et des données réelles provenant de l'ensemble d'entraînement. Plus précisément, le discriminateur prend en entrée une instance de données (générée ou réelle) et prédit si elle appartient à l'ensemble d'entraînement réel ou s'il s'agit d'une instance générée par le générateur. Le but du discriminateur est d'être le plus précis possible dans sa prédiction afin de minimiser l'erreur de classification.

En même temps, le générateur est formé pour tromper le discriminateur en produisant des instances générées qui sont indiscernables des instances réelles, forçant ainsi le discriminateur à être moins précis. Finalement, le discriminateur doit être capable de distinguer les instances générées des instances réelles avec une précision d'environ 50%, ce qui signifie qu'il ne peut plus faire la distinction entre les deux types d'instances.

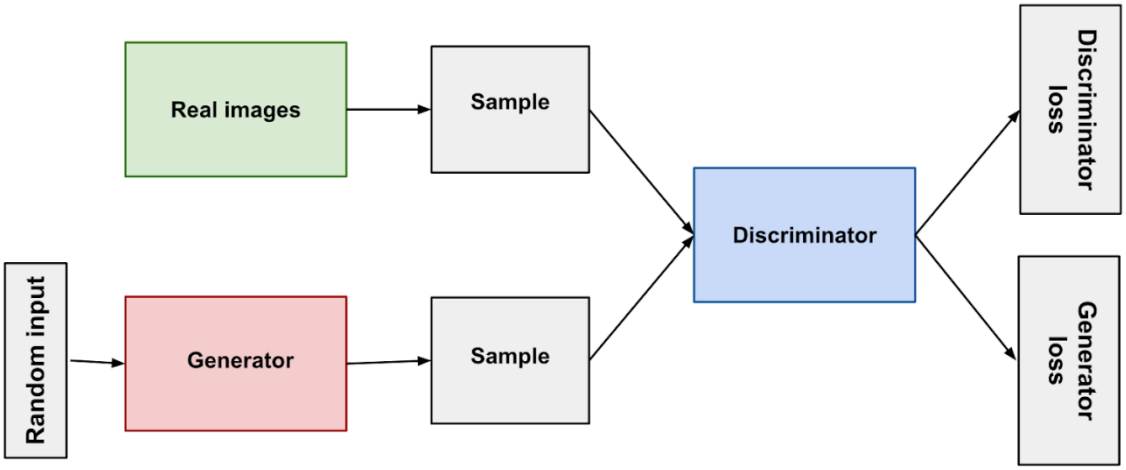

Figure 7 : Architecture du GAN

Le générateur et le discriminateur sont des réseaux de neurones. La sortie du générateur est directement connectée à l'entrée du discriminateur comme l'illustre la figure 7. Grâce à la rétropropagation, la classification du discriminateur fournit un signal permettant au générateur de mettre à jour ses pondérations.

2.2.1.1. Le discriminateur

Le rôle du discriminateur dans un GAN consiste à effectuer une classification des données générées par le générateur et des données réelles. En d'autres termes, il cherche à les distinguer les unes des autres. Pour réaliser cette tâche, le discriminateur peut être construit avec n'importe quelle architecture de réseau CNN, RNN, DNN … qui convient au type de données qu'il doit classifier.

La figure 8 illustre généralement le processus de rétropropagation dans un GAN en montrant les connexions entre le discriminateur, le générateur et les données réelles/générées.

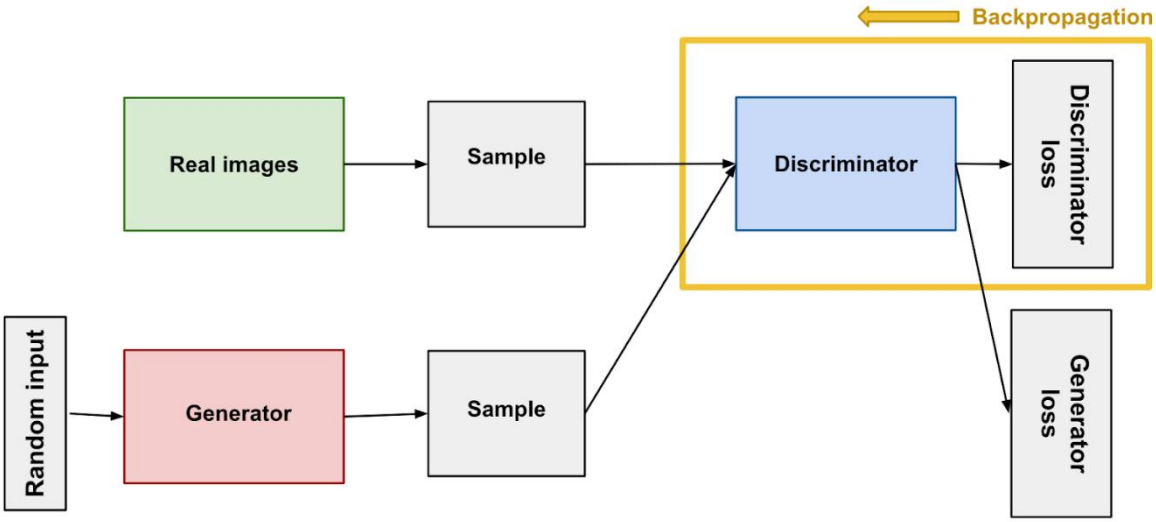

Figure 8:Rétropropagation de l'entraînement discriminateur

- **Les données d'entraînement du discriminateur**

Le discriminateur d'un GAN utilise deux sources de données pour son entraînement :

Les instances de données réelles, telles que des images réelles de personnes, sont utilisées comme exemples positifs.

Les instances de données fictives créées par le générateur sont utilisées comme exemples négatifs.

Ces deux sources de données sont représentées par les cases "Sample"(échantillon) dans la figure 8. Pendant l'entraînement du discriminateur, le générateur ne s'entraîne pas, ses poids sont figés, et il génère des exemples pour permettre au discriminateur de s'entraîner sur des exemples positifs et négatifs.

- **Entraîner le discriminateur**

(1) Premièrement, prenez un exemple réel aléatoire x de l'ensemble de données d'entraînement donné.

(2) Maintenant, obtenez un nouveau vecteur aléatoire z et, en utilisant le réseau du générateur, synthétisez un exemple faux ŷ.

(3) Utilisez le réseau du discriminateur pour distinguer entre ŷ et y.

(4) Trouvez l'erreur de classification et effectuez une rétropropagation. Ensuite, essayez de minimiser l'erreur de classification pour mettre à jour les biais et les poids du discriminateur.

2.2.1.2. Le générateur

Le générateur dans un modèle GAN (Generative Adversarial Network) est un réseau de neurones qui prend en entrée un vecteur aléatoire (également appelé vecteur de bruit) et produit en sortie une image (ou un autre type de données) qui doit ressembler à une image réelle. Le but du générateur est d'apprendre à produire des sorties qui soient suffisamment réalistes pour tromper le discriminateur, c'est-à-dire le deuxième réseau de neurones dans le modèle GAN. Au fur et à mesure de l'entraînement, le générateur est modifié pour améliorer la qualité de ses sorties afin de tromper le discriminateur.

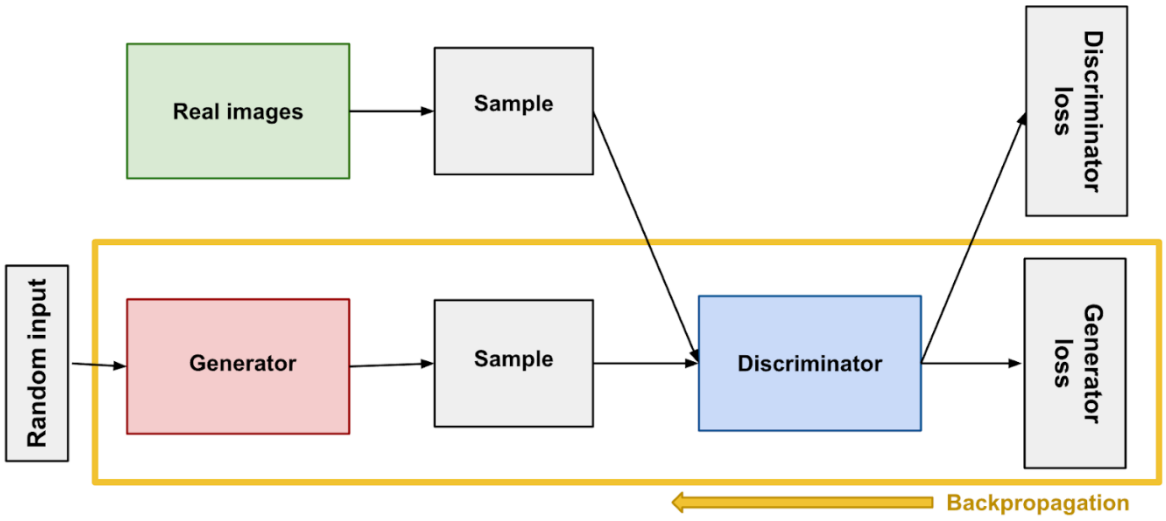

Figure 9:Rétropropagation de l'entraînement du générateur

La rétropropagation du gradient du générateur, comme illustré dans la figure 9, consiste à propager l'erreur de classification du discriminateur vers l'arrière à travers les

couches du réseau du générateur, afin d'ajuster les poids et les biais et d'améliorer progressivement sa capacité à générer des données plus réalistes.

- **Les données d'entraînement du générateur**

Dans sa forme la plus basique, un GAN prend un bruit aléatoire comme entrée. Le générateur transforme ensuite ce bruit en une sortie significative. En introduisant du bruit, nous pouvons demander au GAN de produire une grande variété de données, à partir d'échantillons de différents endroits de la distribution cible.

Les tests suggèrent que la répartition du bruit n'a pas d'importance. Nous pouvons donc choisir un élément facile à échantillonner, comme une distribution uniforme. Pour plus de commodité, l'espace à partir duquel le bruit est échantillonné est généralement de dimension inférieure à celle de l'espace de sortie.

- **Entraîner le générateur**

Nous allons donc entraîner le générateur de la manière suivante :

(1) Exemple de bruit aléatoire

(2) Génère une sortie de générateur à partir du bruit aléatoire échantillonné.

(3) Obtenez une classification "réelle" ou "fausse" du discriminateur pour la sortie du générateur.

(4) Calculez la perte à partir de la classification des discriminateurs

(5) Effectuez une rétropropagation via le discriminateur et le générateur pour obtenir des gradients.

(6) Utilisez les gradients pour modifier uniquement les pondérations du générateur.

2.2.2. La convergence

À mesure que le générateur s'améliore avec l'entraînement, les performances du discriminateur sont moins bonnes, car le discriminateur ne fait pas la différence entre le vrai et le faux. Si le générateur réussit parfaitement, le discriminateur a une précision de 50 %. En fait, le discriminateur sort une pièce de monnaie pour effectuer sa prédiction.

Cette progression pose un problème pour la convergence du GAN dans son ensemble : les commentaires des discriminateurs deviennent moins significatifs au fil du temps. Si le GAN continue de s'entraîner au-delà du moment où le discriminateur envoie des commentaires

complètement aléatoires, le générateur commence à s'entraîner sur les commentaires indésirables et sa qualité peut s'effondrer.

Pour un GAN, la convergence est souvent un état temporaire, plutôt que stable.

2.2.3. La fonction de perte

Les GAN cherchent à reproduire une distribution de probabilité, ce qui nécessite l'utilisation de fonctions de perte reflétant la distance entre la distribution des données générées par le GAN et celle des données réelles. Mais comment mesurer la différence entre deux distributions dans les fonctions de perte du GAN ? Cette question fait l'objet d'une recherche active, et plusieurs approches ont été proposées.

Dans l'article présentant les GAN, le générateur tente de minimiser la fonction suivante,

Tandis que le discriminateur tente de l'optimiser : [2]

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

D(x) est une estimation de la probabilité que l'instance de données x réelle soit réelle.

Ex est la valeur attendue sur toutes les instances de données réelles.

G(z) est la sortie du générateur lorsqu'il reçoit une valeur de z pour le bruit.

D(G(z)) est l'estimation de la probabilité qu'une fausse instance soit réelle.

Ez est la valeur attendue sur toutes les entrées aléatoires du générateur (en effet, il s'agit de la valeur attendue sur toutes les fausses instances générées, G(z)).

La formule est dérivée de l'entropie croisée entre les distributions réelles et générées.

Le générateur ne peut pas directement affecter le terme log(D(x)) dans la fonction. Par conséquent, le fait de minimiser la perte revient à réduire le log (1 - D(G(z))).

L'article fondateur sur les GANs souligne que la fonction de perte MinMax présentée précédemment peut entraîner un blocage du GAN au début de l'entraînement, lorsque le rôle du discriminateur est très facile. Pour remédier à ce problème, l'article suggère de modifier la perte du générateur afin qu'elle tente de maximiser log D(G(z)).

2.3. Les réseaux antagonistes génératifs conditionnels

En 2014, Mehdi Mirza (un doctorant à l'Université de Montréal) et Simon Osindero (un architecte chez Flickr AI) ont publié un article sur les réseaux antagonistes génératifs conditionnels (C-GAN) [3], dans lequel le générateur et le discriminateur du modèle GAN original sont conditionnés lors de l'apprentissage sur des informations externes. Ces informations peuvent être une étiquette de classe ou des données provenant d'autres modalités.

L'idée est simple. Le générateur et le discriminateur reçoivent tous deux une étiquette de classe Y et sont conditionnés par celle-ci, comme le montrent la figure 10. Tous les autres composants sont exactement ce que vous voyez dans un cadre typique de réseaux antagonistes génératifs, cela étant plus une modification architecturale.

Pendant l'apprentissage du C-GAN :

Le générateur est paramétré pour apprendre et produire des échantillons réalistes pour chaque étiquette dans l'ensemble de données d'apprentissage.

Le discriminateur apprend à distinguer les échantillons faux et réels, en tenant compte des informations d'étiquettes.

Cependant, leurs rôles ne changent pas. Le générateur et le discriminateur continuent de générer et de classer des images comme avant, mais avec des informations auxiliaires conditionnelles.

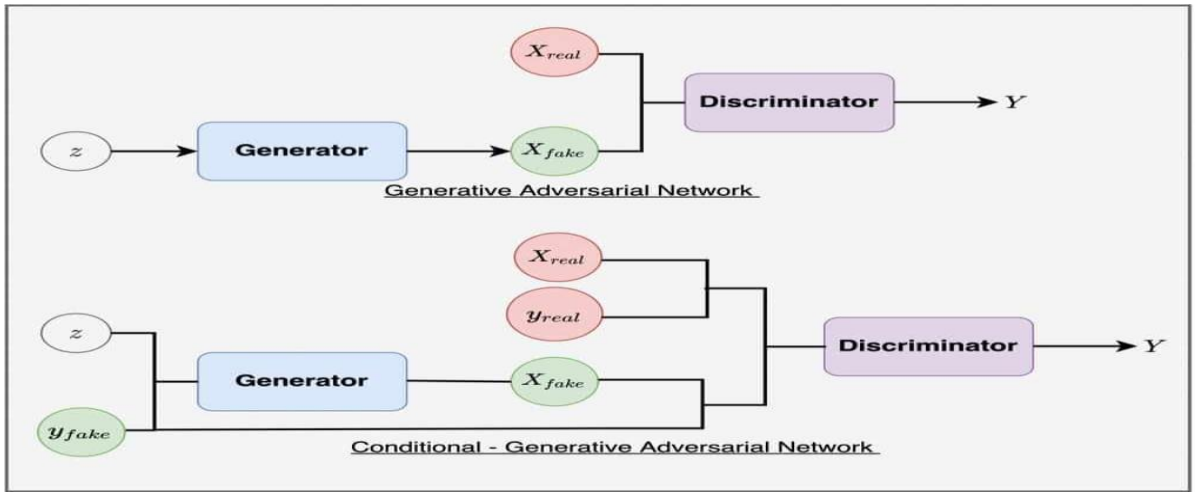

Figure 10:représentation des GANs et CGANs

2.3.1. Le générateur C-GAN

En général, le générateur a besoin d'un vecteur de bruit pour générer un échantillon. Dans une génération conditionnelle, il a également besoin d'informations auxiliaires qui indiquent au générateur quelle classe d'échantillon produire.

Appelons l'étiquette de conditionnement y. Le générateur utilise le vecteur de bruit z et l'étiquette y pour synthétiser un faux exemple :

G (z, y) = x|y

(x conditionné par y, où x est le faux exemple généré).

Cet exemple faux vise à tromper le discriminateur en ressemblant autant que possible à un exemple réel pour l'étiquette donnée.

Maintenant, il ne suffit pas que le Générateur produise des données ayant l'air réalistes ; il est également important que les exemples générés correspondent également à l'étiquette. Ainsi, si une étiquette de classe particulière 1 est passée au Générateur, il devrait produire une image de classe 1. On peut donc clairement remarquer que le Générateur conditionnel assume maintenant beaucoup plus de responsabilités que le GAN simple

Une fois que le Générateur est entièrement entraîné, on peut spécifier l'exemple que vous souhaitez maintenant que le Générateur conditionnel produit en lui passant simplement l'étiquette désirée.

2.3.2. Le Discriminateur du C-GAN

Le Discriminateur reçoit à la fois des exemples réels et des exemples faux avec des étiquettes. Il apprend non seulement à reconnaître les données réelles des données fausses, mais aussi à trouver des paires correspondantes. Une paire est considérée correspondante lorsque l'image a une étiquette correcte qui lui est attribuée. Le Discriminateur renvoie finalement une probabilité indiquant si l'entrée est réelle ou fausse. Son objectif est d'apprendre à :

Accepter toutes les paires d'échantillons réels avec étiquettes correspondantes.

Rejeter toutes les paires d'échantillons faux (l'échantillon correspond à l'étiquette).

Rejeter également tous les échantillons faux si les étiquettes correspondantes ne sont pas cohérentes.

Par exemple, le Discriminateur doit apprendre à rejeter :

La paire dans laquelle l'image est générée de classe 1, mais l'étiquette est de classe 2, peu importe si l'exemple est réel ou faux. La raison est que cela ne correspond pas à l'étiquette donnée.

Toutes les paires image-étiquette dans lesquelles l'image est fausse, même si l'étiquette correspond à l'image.

2.3.3. La fonction de perte

Dans le C-GAN, le générateur est conditionné non seulement à un bruit aléatoire, mais aussi à une variable de conditionnement y qui peut être une étiquette de classe ou toute autre information auxiliaire. Cette variable y est également fournie au discriminateur. La fonction de perte du C-GAN utilise la divergence de Kullback-Leibler (KL)pour mesurer la distance entre la distribution de probabilité réelle des données conditionnées et la distribution de probabilité prédite par le modèle génératif. L'objectif du C-GAN est de minimiser cette distance tout en maximisant la capacité du discriminateur à distinguer les données réelles conditionnées de celles générées par le générateur. [3]

La fonction objective d'un jeu à deux joueurs minimax serait comme Equation de min G maxD

V (D, G) = Ex∼pdata(x) [log D(x|y)] + Ez∼pz(z) [log(1 − D(G(z|y)))].

Dans le contexte du C-GAN, La Binary Cross-Entropy (BCE) est une fonction de coût souvent utilisée pour les tâches de classification binaire, qui mesure la différence entre la distribution de probabilité réelle et la distribution prédite par le modèle pour des données binaires.

La BCE est souvent utilisée comme alternative à la KL pour mesurer la distance entre la distribution réelle des données conditionnées et la distribution prédite par le générateur. La BCE mesure la différence entre les probabilités de classe réelle et prédite pour chaque observation dans les données conditionnées.

Dans le modèle C-GAN, la fonction de perte BCE est utilisée pour aider à atteindre les objectifs des deux réseaux, le générateur et le discriminateur. [4]

binary_cross_entropy = tf.keras.losses.BinaryCrossentropy()

2.3.3.1. La perte du générateur

La perte du générateur mesure à quel point les exemples générés sont réalistes tout en correspondant au label spécifié.

```
def generator_loss(label, fake_output):
 gen_loss = binary_cross_entropy(label, fake_output)
 print(gen_loss)
 return gen_loss
```

2.3.3.2. La perte du discriminateur

La perte du discriminateur mesure à quel point le discriminateur est capable de distinguer les exemples réels des exemples générés tout en corrélant correctement chaque exemple avec son label.

Les prédictions de sortie réelles (images originales) ont une étiquette de 1.

Les prédictions de sortie fausses ont une étiquette de 0.

```
def discriminator_loss(label, output):
disc_loss = binary_cross_entropy(label, output)
print(total_loss)
return disc_loss
```

2.3.3.3. Entraînement du GAN conditionnel

Lors du passage en avant (forward pass), dans les deux modèles, le générateur conditionnel et le discriminateur conditionnel, nous fournissons une liste de tenseurs en entrée. Pendant la passe en avant du générateur, nous alimentons le vecteur de bruit et l'étiquette (label), tandis que pour la propagation avant (forward propagation) du discriminateur, nous fournissons l'image réelle / fausse et l'étiquette (label). Ensuite, nous calculons la perte pour chaque modèle en fonction de ces entrées, que nous utilisons pour mettre à jour les poids et les biais (paramètres) pendant la rétropropagation (backpropagation). Ce processus est répété pour plusieurs époques jusqu'à ce que le générateur et le discriminateur atteignent une convergence satisfaisante

2.4. Le modèle pix2pix

2.4.1. Introduction

Le modèle Pix2Pix [5] a été introduit par les chercheurs de l'Université de Berkeley en 2016. Il utilise une architecture de type GAN, qui est composée de deux parties principales : un générateur et un discriminateur. Le générateur prend en entrée une image de faible qualité ou incomplète, telle qu'un croquis, et produit une image de haute qualité et réaliste. Le discriminateur évalue ensuite la qualité de l'image produite par le générateur et détermine si elle est similaire à une image réelle ou non.

Le modèle Pix2Pix utilise également une technique appelée "conditional GAN" où le générateur est conditionné à l'entrée, de sorte qu'il génère des images cohérentes avec l'entrée fournie. Cette technique permet au modèle de générer des images très précises et réalistes.

En termes de performances, le modèle Pix2Pix a obtenu d'excellents résultats dans divers domaines. Par exemple, dans le domaine médical, il a été utilisé pour la restauration d'images de tomographie par ordinateur (CT) et d'imagerie par résonance magnétique (IRM) endommagées ou incomplètes. Dans le domaine de l'architecture, il a été utilisé pour générer des images de bâtiments à partir de plans d'architectes. Dans le domaine de la photographie, il a été utilisé pour convertir des images de nuit en images de jour, et dans le domaine de l'art, il a été utilisé pour traduire des croquis en images photoréalistes.

Le modèle Pix2Pix est une architecture de réseau de neurones profonds très performante qui a fait ses preuves dans de nombreux domaines différents. Sa capacité à générer des images de haute qualité à partir d'entrées de faible qualité ou incomplètes en fait une technologie très prometteuse pour l'avenir.

Pour notre problématique qui est de générer un sourire réaliste (Smile Design) pour des images de dents à partir d'un croquis, le modèle Pix2Pix est une solution très adaptée. Nous pouvons entraîner le modèle sur une base de données d'images de dents et de sourires réalistes, en fournissant une image de croquis comme entrée et une image de sourire réaliste correspondante comme cible. Le modèle sera alors capable de générer des images de sourires très réalistes à partir de ces croquis.

Il est important de noter que pour obtenir des résultats optimaux, nous devrons disposer d'une base de données d'images de dents et de sourires réalisés de haute qualité et en quantité

suffisante pour entraîner le modèle de manière efficace. Nous devrons également nous assurer que les croquis fournis en entrée sont suffisamment détaillés pour permettre au modèle de générer des images de sourires précises et réalistes.

En utilisant le modèle Pix2Pix pour générer des sourires à partir de croquis de dents, nous pourrons potentiellement révolutionner l'industrie de la dentisterie esthétique en permettant aux patients de visualiser le résultat final avant même de commencer le traitement, ce qui peut aider à réduire l'anxiété et à améliorer la satisfaction des patients

2.4.2. Paired Image-to-Image Translation

Paired Image-to-Image Translation (traduction des pairs d'images à image) est une technique de traitement d'images qui consiste à créer une correspondance entre des images d'entrée et des images de sortie. Le but est de traduire une image d'un domaine vers une autre en utilisant un modèle de réseau de neurones profond. Pour cela, le modèle est entraîné sur un ensemble de données appariées (paired images), où chaque paire d'images est liée par une correspondance.

Dans la traduction des pairs d'images à image, les domaines d'entrée et d'images de référence sont alignés. Bien que l'obtention d'échantillons d'entraînement puisse être difficile, ce type de traduction conduit souvent à d'excellents résultats. Dans l'image la figure 11, nous avons différentes tâches de traduction d'images à image. Par exemple :

Une carte de segmentation d'une scène urbaine est traduite en une image RGB (scène de rue) avec tous les contenus de l'image d'entrée préservés.

Il y a la tâche de segmentation sémantique (étiquette) vers une scène de rue, qui nécessite un entraînement apparié car vous ne voulez pas générer une scène complètement aléatoire, étant donné une étiquette d'un scénario spécifique.

Dans une autre tâche de traduction, l'image en noir et blanc d'une fleur est traduite en une image en couleur, avec la fleur et les contenus de l'image d'entrée globale très présents dans l'image en couleur traduite.

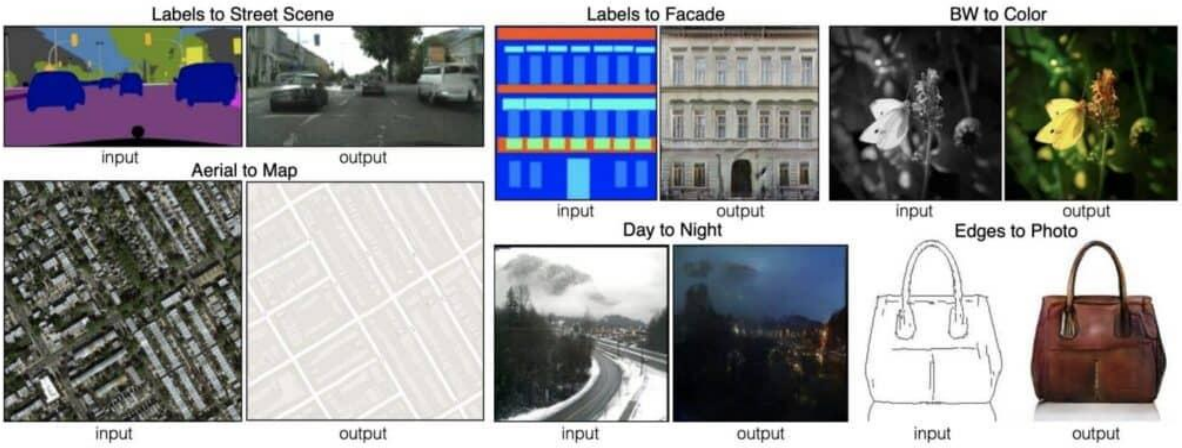

Figure 11 : représentation des images pairs pour le modèle pix2pix

2.4.3. Le modèle génératif pix2pix

En 2016, un groupe de chercheurs dirigé par Phillip Isola, à Berkeley AI Research a publié un article intitulé "Image-to-Image Translation with Conditional Adversarial Networks" et l'a ensuite présenté à CVPR (Conference on Computer Vision and Pattern Recognition)2017. Cet article a déjà recueilli plus de 7400 citations !

Pix2pix était inspiré du Conditional GAN (CGAN), qui était à nouveau une extension de l'architecture DCGAN.

Récapitulons rapidement quelques points :

Qu'il s'agisse d'un GAN vanille ou d'un DCGAN, l'architecture était composée de deux réseaux : le générateur et le discriminateur. Les deux étaient entraînés en tandem, le générateur essayant de reproduire la distribution de données réelles, tandis que le réseau discriminateur apprenait à classifier les données réelles des données fausses (générées).

Après l'entraînement, le générateur utilisait du bruit aléatoire en entrée pour produire des images réalistes similaires à celles de l'ensemble de données.

Bien que le générateur produit des images ayant l'air réalistes, nous n'avions certainement aucun contrôle sur le type ou la classe d'images générées. C'est alors que le CGAN est apparu, permettant une génération contrôlée. Extension de l'architecture GAN, le CGAN a généré des images conditionnées par des étiquettes de classe.

Dans la figure 12 le générateur recevait un vecteur de bruit aléatoire conditionné par l'étiquette de classe.

Et le discriminateur recevait des images réelles ou fausses (générées) conditionnées par l'étiquette de classe.

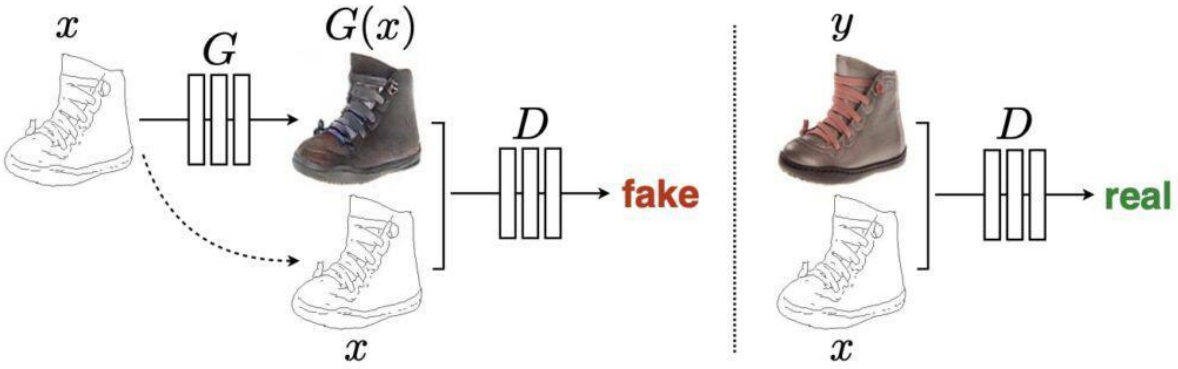

Figure 12: traduction d'images de chaussures à partir d'un croquis (image de contours)

Le GAN Pix2Pix étend encore plus l'idée du CGAN, où les images sont traduites d'une image d'entrée à une image de sortie, conditionnée par l'image d'entrée. Pix2Pix est un GAN conditionnel qui effectue une traduction d'image à image.

Le générateur de chaque GAN que nous avons vu jusqu'à présent recevait un vecteur de bruit aléatoire, échantillonné à partir d'une distribution uniforme. Mais le GAN Pix2Pix élimine totalement le concept de vecteur de bruit du générateur.

Dans Pix2Pix :

Une image est entrée dans le réseau du générateur, qui produit ensuite une version traduite.

Le discriminateur du pix2pix est un discriminateur conditionnel, qui reçoit une image réelle ou générée (fausse) qui a été conditionnée sur la même image d'entrée qui a été introduite dans le générateur.

L'objectif du discriminateur est de classer la paire d'images comme réelle (du jeu de données) ou fausse (générée).

L'objectif final de Pix2Pix GAN reste le même que celui de tous les GANs. Il cherche également à tromper le discriminateur, de sorte que le générateur apprend à traduire les images parfaitement.

Pour mieux comprendre, référons-nous à la figure 13 dans laquelle un GAN conditionnel est formé pour cartographier les contours vers des photos.

Étant donné une image de contour d'entrée (avant la traduction) x,

Le générateur G apprend à produire une photo de chaussure traduite G(x) similaire à la vraie photo de chaussure y(Ground Truth).

Le discriminateur D apprend à classer

La photo générée G(x), conditionnée sur l'entrée x, comme fausse

La vraie photo y, conditionnée sur l'entrée x, comme réelle.

Contrairement à un GAN non contrôlé, le générateur et le discriminateur observent tous deux la carte de contour d'entrée.

Figure 13:exemple d'image généré à partir d'un croquis

Dans les architectures GAN antérieures, le vecteur de bruit aidait à générer des sorties différentes en y ajoutant de l'aléatoire. Mais cela n'était pas très utile dans Pix2Pix GAN, donc les auteurs s'en sont débarrassés. Cependant, ils ont trouvé un moyen de maintenir une légère stochasticité dans la sortie du générateur, c'est-à-dire en ajoutant le Dropout dans le réseau du générateur.

2.4.4. Architecture du réseau pix2pix

2.4.4.1. Générateur pix2pix

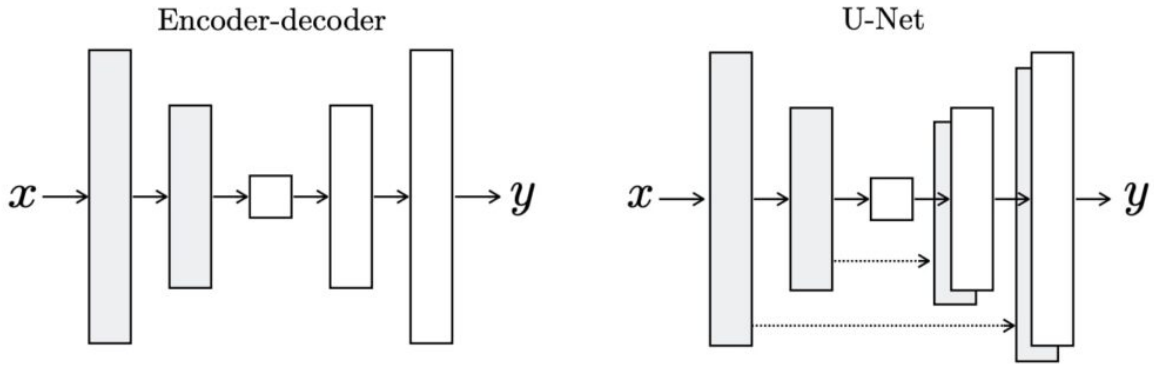

Figure 14:architecture encoder-decoder vs U-Net

Référons-nous à la figure 14, toutes les architectures de générateur que nous avons vues jusqu'à présent prennent en entrée un vecteur de bruit aléatoire (qui peut être ou non conditionné à une étiquette de classe) pour générer une image. Ainsi, tous ces réseaux de générateurs fonctionnent comme le décodeur d'un autoencodeur, c'est-à-dire qu'ils prennent un vecteur latent en entrée pour produire une image.

Mais la situation change dans Pix2Pix. Cette méthode rejette l'architecture de générateur traditionnelle pour adopter le style de l'autoencodeur, qui comporte à la fois des réseaux d'encodeurs et de décodeurs. Mais pourquoi ? La réponse est assez simple. Dans Pix2Pix, contrairement aux architectures GAN traditionnelles, l'entrée et la sortie sont toutes deux des images. Et quoi de mieux qu'un autoencodeur pour cette tâche.

Dans Pix2Pix, nous trouvons un générateur U-NET, comprenant un encodeur-décodeur, avec des connexions de saut entre les couches en miroir, dans les deux piles.

Le générateur UNET peut prendre en entrée des contours, des étiquettes de segmentation sémantique, des images en noir et blanc, etc.

La sortie peut être une photo de sac, chaussure, une scène de rue, une image en couleur, etc comme illustré dans la figure 11.

Dans un autoencodeur, la sortie y est aussi proche que possible de l'entrée x. Mais dans un générateur UNET [6] :

Cette y est une version traduite et conditionnée de x.

De plus, des connexions de saut sont introduites, ce qui aide à récupérer toutes les informations perdues lors du sous-échantillonnage de l'entrée de l'encodeur.

Pendant la rétropropagation, cela aide même à améliorer le flux de gradient en évitant le problème du gradient qui disparaît (Vanishing Gradient Prolem).

Comme mentionné précédemment, l'architecture utilisée dans Pix2Pix s'appelle U-net. U-net a été développé à l'origine pour la segmentation d'images biomédicales par Ronneberger en 2015.

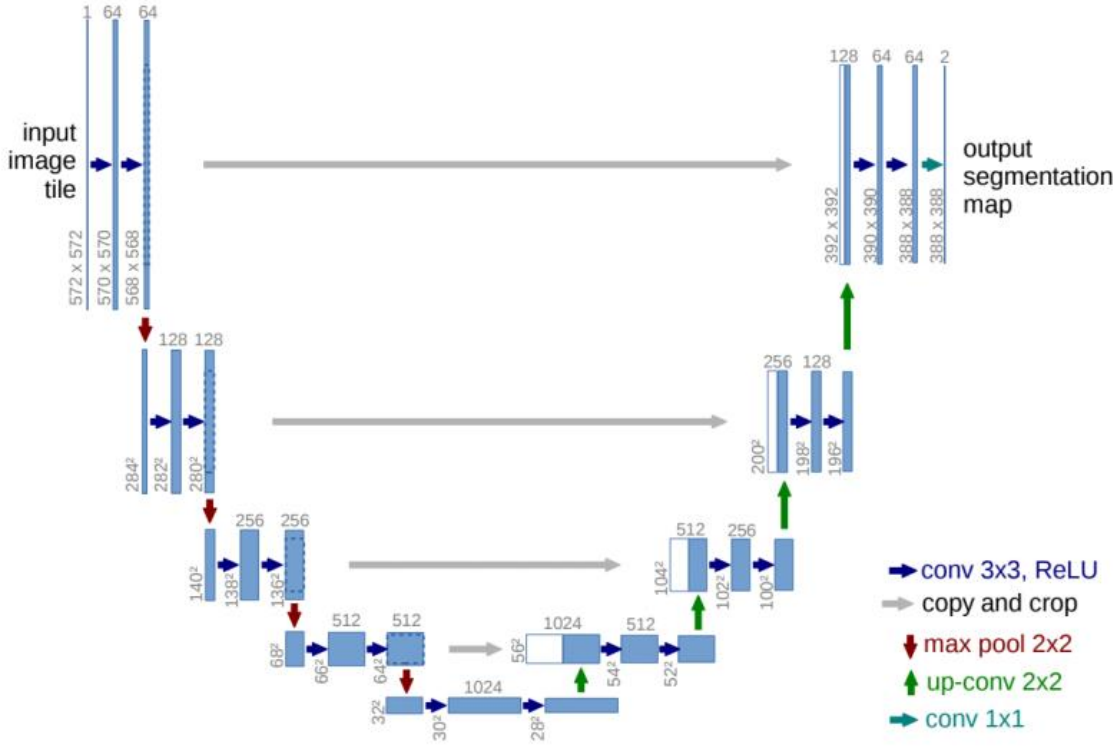

Figure 15 : Architecture U-NET

Référons-nous à la figure 15 U-net se compose de deux parties principales :

- A contracting path : Un chemin contractant composé de couches de convolution (côté gauche) qui sous-échantillonnent les données tout en extrayant des informations.
- An expansive path : Un chemin expansif composé de couches de convolution transposées (côté droit) qui sur-échantillonnent les informations.

Prenons l'exemple dans la figure 16 où notre sous-échantillonnage comporte trois couches de convolution C_l(1,2,3). Nous devons alors nous assurer que notre sur-échantillonnage comporte également trois couches de convolution transposées C_u(1,2,3).

Cela est nécessaire pour connecter les blocs correspondants de mêmes tailles en utilisant une connexion de saut.

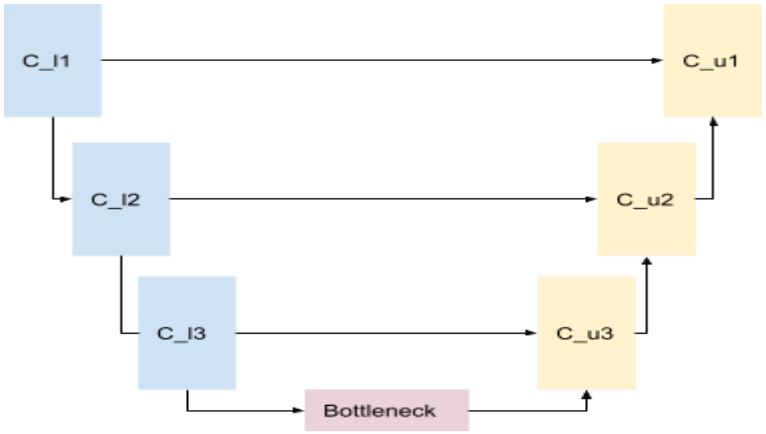

Figure 16: exemple de couche de convolution du modèle U-net

- Downsampling :

  Pendant la phase de sous-échantillonnage, chaque bloc de convolution extrait des informations spatiales et les transmet au bloc de convolution suivant pour en extraire davantage, jusqu'à ce qu'il atteigne la partie centrale appelée bottleneck. La phase de sur-échantillonnage commence à partir du bottleneck.

- Upsampling :

  Pendant la phase de sur-échantillonnage, chaque bloc de convolution transposée étend les informations du bloc précédent tout en concaténant les informations du bloc de sous-échantillonnage correspondant. En concaténant les informations, le réseau peut apprendre à assembler une sortie plus précise en se basant sur ces informations.

  Cette architecture est capable de localiser, c'est-à-dire qu'elle est capable de trouver l'objet d'intérêt pixel par pixel. De plus, U-net permet également au réseau de propager des informations de contexte des couches de basse résolution aux couches de haute résolution. Cela permet au réseau de générer des échantillons de haute résolution.

- Les skip connexions :

  Les skip-connections sont une caractéristique importante de l'architecture du générateur U-net dans Pix2Pix. Elles permettent de récupérer les informations perdues pendant la phase de sous-échantillonnage et de les utiliser pendant la phase de sur-échantillonnage pour générer des images plus précises. Les connexions de saut sont des liens directs entre les couches de

l'encodeur et du décodeur, ce qui permet de transmettre des informations à travers différentes échelles spatiales. En connectant les blocs correspondants de mêmes tailles, le réseau peut utiliser des informations provenant de différentes échelles spatiales pour générer des images plus détaillées. De plus, les skip-connections aident également à résoudre le problème du gradient qui disparaît lors de la rétropropagation, en permettant un flux de gradient plus efficace entre les couches du réseau. Pour conclure, les skip-connections sont une caractéristique clé de l'architecture du générateur U-net dans Pix2Pix, qui permet d'obtenir des résultats de haute qualité dans les tâches de traduction d'images.

2.4.4.2. Le discriminateur PatchGan

- **Terminologie patch :**

Dans le contexte de PatchGAN, le terme "patch" fait référence à une petite région carrée de l'image d'entrée, généralement de taille 70x70 pixels. La PatchGAN évalue ces patchs locaux de l'image plutôt que l'image entière pour déterminer si chaque patch est réel ou faux. Cela permet une classification plus fine et détaillée de l'image en utilisant des informations locales.

Le Discriminateur Pix2Pix a le même objectif que tout autre discriminateur GAN, c'est-à-dire de classer une entrée comme réelle (échantillonnée du jeu de données) ou fausse (produite par le générateur). Son architecture diffère un peu cependant, principalement en termes de la façon dont l'entrée est régularisée au niveau de la couche de sortie (finale).

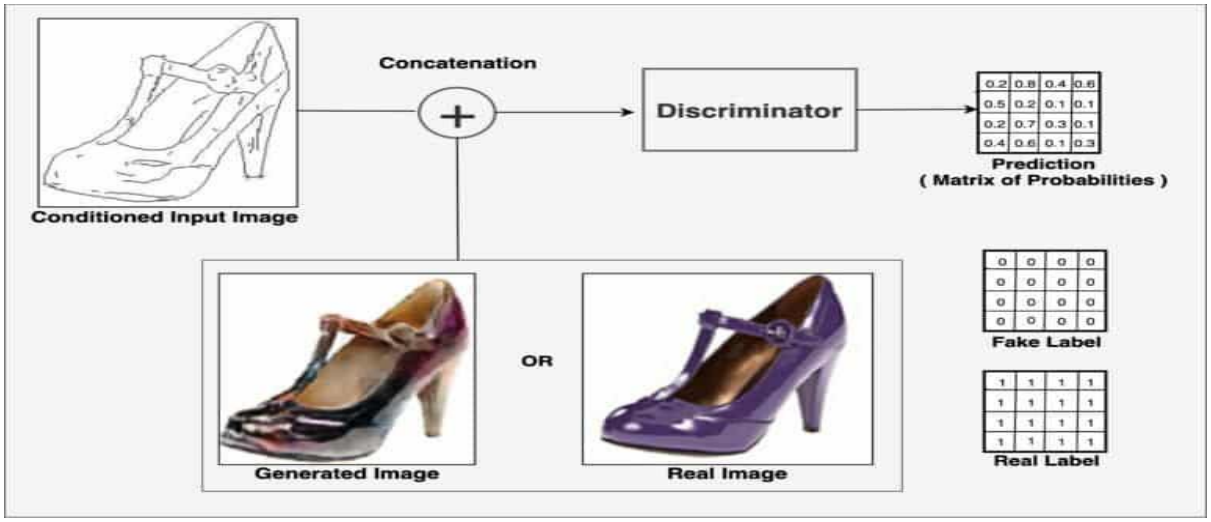

Figure 17: Pix2pix gan discriminateur

- **Le Patchgan**

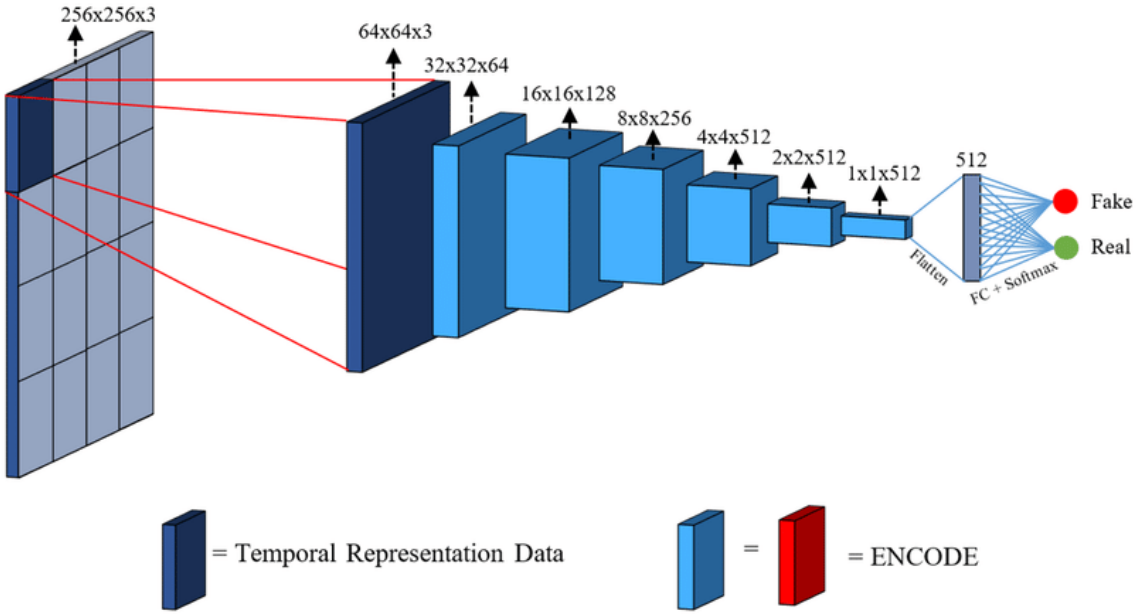

Figure 18 : architecture du patchgan

Pourquoi utiliser le PatchGAN? Principalement à cause de la matrice de valeurs que le discriminateur produit pour une entrée donnée, voir figure 17. Contrairement au modèle GAN traditionnel qui utilise un CNN avec une seule sortie pour classifier les images, le modèle Pix2Pix utilise un PatchGAN spécialement conçu pour classifier des patches (70×70) d'une image d'entrée comme réels ou faux (fake , real) Conformément à la figure 18, plutôt que de considérer l'ensemble de l'image en une seule fois.

- **Caractéristiques du patchgan**

Le discriminateur utilise des blocs de couches standard de :

-convolution

-batch normalization

-ReLU,

Le nombre de couches est configuré de telle sorte que le champ réceptif effectif de chaque sortie du réseau se rapporte à une taille spécifique dans l'image d'entrée.

Le réseau produit une seule carte de caractéristiques de prédictions réelles/fausses qui peuvent être moyennées pour donner une seule valeur de score (valeur de perte).

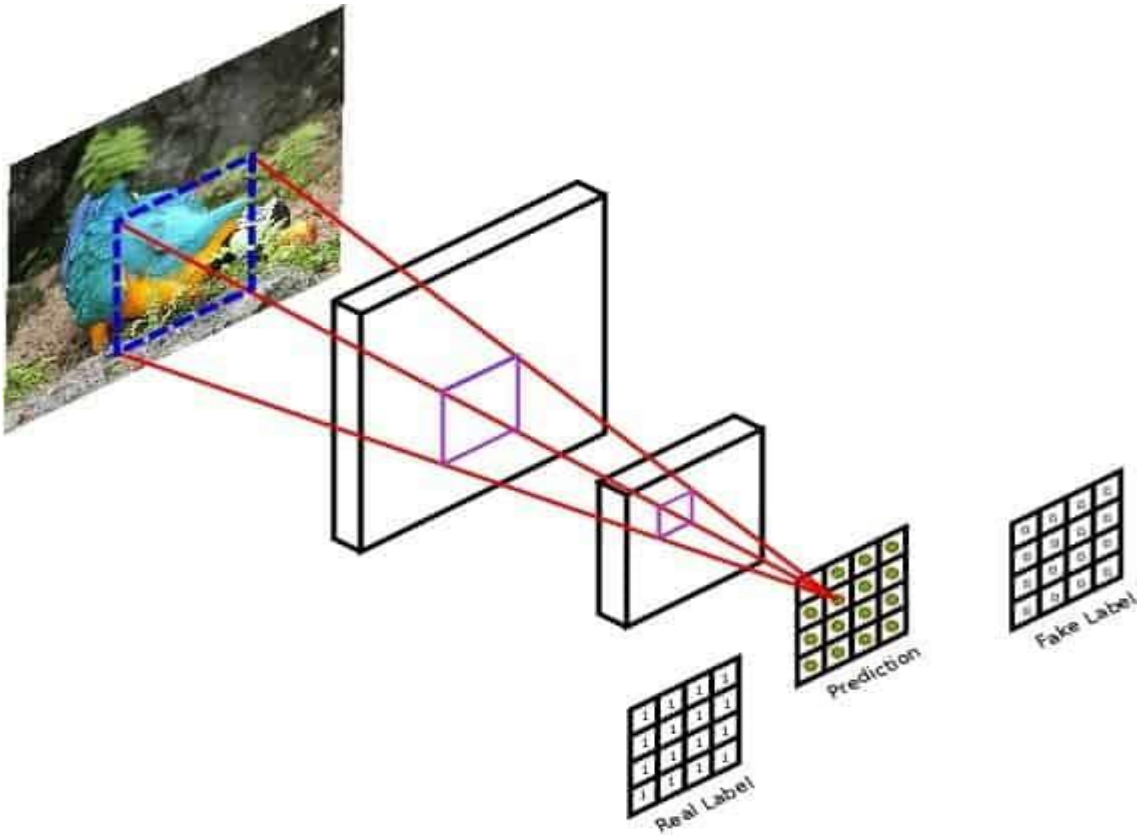

Figure 19: la prédiction du discriminateur patchgan.

La sortie comme le montre la figure 19 est une matrice et chaque élément de la matrice représente une région locale dans l'image d'entrée. Si la région locale est réelle, nous devrions obtenir 1, sinon 0.

Les deux principaux points forts du discriminateur PatchGAN sont :

Discriminateur conditionnel : Inspiré par le GAN conditionnel, le discriminateur reçoit des images réelles ou fausses conditionnées par l'image d'entrée. Ainsi, l'entrée est une version concaténée de l'image réelle ou fausse et de l'image d'entrée (bords, dans le cas de edges->photos). Avec cette condition, le discriminateur essaie de déterminer si l'image réelle ou générée ressemble réellement à une cartographie réaliste de l'image de bord d'entrée ou non.

Le patch : L'image d'entrée de dimension [256, 256, n_channels] est classée en patches, ce qui signifie que la sortie est un tenseur de dimension [30, 30], et non une valeur scalaire.

À la sortie, chaque [1, 1] dans le tenseur [30, 30] représente un patch de [70, 70] dans l'image d'entrée. Chaque grille de ce tenseur de valeurs est classée dans la plage [0, 1], où 0 est faux et 1 est vrai, comme précédemment.

Chaque valeur de la matrice de prédiction de sortie représente la probabilité du patch d'image correspondant d'être réel ou généré (faux).

Une taille de patch de 70×70 s'est avérée efficace pour un certain nombre de tâches de traduction d'image à image.

Excerpt from the paper: [5]

"We Design a discriminator architecture – which we term a PatchGAN – that only penalizes structure at the scale of patches. This discriminator tries to classify if each NxN patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D."

Extrait du papier :

Nous avons conçu une architecture de discriminateur - que nous appelons PatchGAN - qui ne pénalise la structure qu'à l'échelle des patchs. Ce discriminateur essaie de classer si chaque patch NxN dans une image est réel ou faux. Nous exécutons ce discriminateur de manière convolutive sur l'image, en moyennant toutes les réponses pour fournir la sortie ultime de D.

- **Avantages du Patchgan**

Parmis les avantages du patchgan, il donne des commentaires sur chaque région locale ou patch de l'image ce qui aide à mieux classifier les images réelles et fausses.

Cela aide également le générateur à tromper le discriminateur en générant des images encore plus réalistes.

Comme la sortie est la probabilité que chaque patch soit réel ou faux, PatchGAN peut être entraîné avec la perte GAN, c'est-à-dire la perte de croisement binaire (BCE).

Pour une image fausse générée par le générateur, le PatchGAN apprendra à produire un tenseur de toutes les valeurs nulles et l'étiquette correspondante sera également une matrice de toutes les valeurs nulles. Cela signifie que le discriminateur doit produire des zéros pour toutes les valeurs de la matrice pour atteindre une perte minimale.

Pour une image réelle, le PatchGAN apprendra à produire un tenseur de toutes les valeurs 1, et l'étiquette correspondante sera une matrice de toutes les valeurs un.

- **La fonction de perte pix2pix**

  Pour optimiser à la fois le générateur et le discriminateur, l'approche d'entraînement standard est suivie, c'est-à-dire qu'on alterne une étape de descente de gradient sur le discriminateur, suivie d'une étape sur le générateur.

  L'extrait suivant du papier donne l'essentiel de la perte de Pix2Pix :

  "The discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground-truth output in an L2 sense. We also explore this option, using L1 distance rather than L2 as L1 encourages less blurring."

- **La perte du Discriminateur**

  La fonction de perte du discriminateur de Pix2Pix est entraînée avec la même perte que les GANs précédents tels que DCGAN, CGAN, c'est-à-dire la perte adversaire (Entropie Croisée Binaire (BCE)). L'objectif du discriminateur est ici de minimiser la probabilité d'identification d'images réelles et fausses.

  Pour ralentir le taux d'apprentissage du discriminateur par rapport au générateur, les auteurs ont divisé la perte par 2 lors de l'optimisation du discriminateur.

  Discriminator loss = (BCE for real images + BCE loss for generator)/2

- **La perte du Générateur**

  Les étiquettes vraies sont utilisées pour entraîner le réseau générateur avec la perte adversaire (perte BCE) pour les images générées. Il a également une perte supplémentaire, c'est-à-dire une perte L1, qui est utilisée pour minimiser l'erreur.

  Cette perte supplémentaire est la somme de toutes les différences absolues entre la valeur réelle et la valeur prédite. La perte L1 agit comme terme de régularisation, pénalisant le générateur si la qualité de reconstruction de l'image traduite n'est pas similaire à celle de l'image cible.

  Dans le contexte du générateur, la perte L1 est la somme de toutes les différences absolues de pixels entre la sortie du générateur (version traduite de l'image d'entrée) et la vraie cible (image cible réelle/attendue).

  L1 Loss =$\sum$ |Generated Output - target|

- **La perte totale**

  Generator Loss = BCE Loss with real Labels + λ.L1Loss

  La perte combinée est régie par un hyperparamètre lambda, où lambda=100 est utilisé pour pondérer le deuxième terme. Les auteurs ont mené une étude d'ablation et ont constaté que 100 était mieux adapté à la perte BCE car cela réduisait les artefacts et produisait en même temps des images plus nettes. Ils ont également évalué la perte L2, mais ont constaté qu'elle produisait des images floues.

2.5. Conclusion

En conclusion, nous avons examiné trois approches populaires dans le domaine de la génération d'images : GAN, CGAN et Pix2Pix. Les GAN sont des réseaux de neurones génératifs qui apprennent à générer des images en utilisant un processus d'entraînement adversarial, où un générateur apprend à générer des images pour tromper un discriminateur qui essaie de les différencier des vraies images. Les CGAN ajoutent une condition supplémentaire à l'entrée du générateur, en fournissant une étiquette de classe ou une condition supplémentaire pour générer des images plus spécifiques.

Enfin, les Pix2Pix utilisent une architecture de réseau de neurones similaire à celle des CGAN, mais au lieu de générer des images à partir de bruit ou de vecteurs de conditionnement, ils apprennent à transformer des images d'entrée en images de sortie correspondantes, en utilisant des images d'entraînement appariées.

Il est important de noter que chaque approche a ses avantages et ses inconvénients, et que leur utilisation dépend des besoins spécifiques de chaque projet. Les GAN peuvent produire des résultats impressionnants mais peuvent également souffrir de problèmes de stabilité d'entraînement. Les CGAN permettent de générer des images plus spécifiques, mais nécessitent des étiquettes de classe ou des conditions supplémentaires. Les Pix2Pix sont utiles pour la transformation d'images, mais nécessitent des paires d'images appariées pour l'entraînement.

Chaque approche possède ses propres nuances et est adaptée à des cas d'utilisation spécifiques. Les GAN, CGAN et Pix2Pix sont tous des outils importants pour la génération d'images et sont susceptibles de continuer à être utilisés et améliorés dans les années à venir.

3. Chapitre 3 Expérimentation et résultat

3.1. Introduction

Le chapitre expérimentation et résultat de notre projet est crucial pour comprendre comment nous avons préparé notre dataset et appliqué le modèle Pix2Pix pour générer un sourire réaliste sur des visages. Dans ce chapitre, nous décrirons en détail toutes les étapes que nous avons suivi pour préparer les données. Nous expliquerons également comment nous avons entraîné notre modèle Pix2Pix en utilisant les données préparées pour qu'il puisse apprendre à générer des sourires réalistes. Enfin, nous présenterons les résultats de nos expériences et les performances de notre modèle.

3.2. Environnement du travail

3.2.1. Hardware

PC PORTABLE GAMER MSI KATANA GF66

I7 12É GEN

RTX 3050 4GO

16 GO RAM

3.2.2. Software

Outils de développement

Anaconda est une distribution de logiciels open source très populaire pour le traitement des données, la science des données et l'apprentissage automatique. Cette distribution contient un grand nombre d'outils, de bibliothèques et de langages de programmation tels que Python,

R, Java, C++ et d'autres, ainsi que des outils de gestion d'environnement tels que conda pour créer et gérer facilement des environnements virtuels.

Anaconda propose également une interface graphique conviviale appelée Anaconda Navigator, qui permet aux utilisateurs de gérer facilement leurs environnements et leurs packages. En plus des outils de base pour le traitement de données, l'analyse et la visualisation, Anaconda propose également des packages pour les applications spécifiques telles que l'apprentissage automatique, la bioinformatique, la géospatialité et bien d'autres domaines.

Anaconda est disponible sur plusieurs systèmes d'exploitation, y compris Windows, macOS et Linux, et est largement utilisé par les scientifiques des données, les ingénieurs et les chercheurs pour le développement de projets de données et de machine learning.

PyCharm est un environnement de développement intégré (IDE) pour le langage de programmation Python. Il est développé par JetBrains et est disponible en version professionnelle et communautaire. PyCharm est largement utilisé par les développeurs Python pour son éditeur de code avancé, son débogueur, sa gestion de projet, son support pour les frameworks populaires tels que Django, Flask et Pyramid, ainsi que ses outils d'analyse de code.

L'éditeur de code de PyCharm offre de nombreuses fonctionnalités, telles que la coloration syntaxique, la saisie semi-automatique, l'indentation automatique, la recherche et le remplacement avancés, la navigation rapide dans le code et bien plus encore. Le débogueur intégré permet aux développeurs de déboguer leur code pas à pas, d'insérer des points d'arrêt et de surveiller les variables en temps réel.

Bibliothèques

PyTorch est un puissant framework open-source d'apprentissage automatique développé principalement par Facebook AI Research. Il est largement utilisé dans le domaine de l'intelligence artificielle et de l'apprentissage profond en raison de sa flexibilité, de sa facilité d'utilisation et de sa grande efficacité. PyTorch met l'accent sur la construction de réseaux de neurones et de modèles d'apprentissage profond en utilisant un paradigme de programmation dynamique, ce qui permet aux développeurs de bénéficier d'une plus grande liberté et d'une plus grande expressivité lors de la création et de l'entraînement de modèles. Grâce à sa popularité croissante et à sa communauté active, PyTorch est devenu un outil essentiel pour les chercheurs, les ingénieurs et les passionnés d'apprentissage automatique à travers le monde.

OpenCV (Open Source Computer Vision) est une bibliothèque open-source très populaire utilisée dans le domaine de la vision par ordinateur et du traitement d'images. Elle a été développée par Intel et est maintenant maintenue par une vaste communauté de développeurs. OpenCV offre une multitude de fonctionnalités et d'outils avancés permettant de manipuler, traiter et analyser des images et des vidéos en temps réel.

Grâce à sa conception modulaire, OpenCV permet aux développeurs d'accéder à un large éventail de fonctionnalités de vision par ordinateur, notamment la détection et la reconnaissance d'objets, la segmentation d'image, la reconnaissance faciale, le suivi d'objets, la calibration de caméra, la stéréovision, la réalité augmentée, et bien d'autres encore. Il propose également des algorithmes d'apprentissage automatique intégrés, offrant ainsi des possibilités d'analyse et de prise de décision avancées.

3.3. Modèle pré-entrainés

3.3.1. Bisenet

Le modèle BiSeNet [7]est une architecture de réseau de neurones convolutionnels profonds qui a été proposée pour effectuer la segmentation d'images en temps réel. "BiSeNet" signifie "Bidirectional Segmentation Network" qui signifie réseau de segmentation bidirectionnel. Cette architecture de réseau de neurones utilise une méthode de segmentation bidirectionnelle pour améliorer la précision de la segmentation tout en conservant une faible complexité.

La méthode de segmentation bidirectionnelle consiste à utiliser deux chemins différents pour extraire les caractéristiques de l'image. Le premier chemin est appelé "pathway spatial" et utilise une résolution d'image élevée pour extraire les caractéristiques spatiales. Le deuxième chemin est appelé "pathway contextuel" et utilise une résolution d'image basse pour extraire les caractéristiques contextuelles. Les caractéristiques spatiales et contextuelles sont ensuite combinées pour produire une segmentation précise.

Plus précisément, le modèle BiSeNet utilise une architecture de réseau de neurones en forme de T où le premier chemin est composé de couches de convolutions résiduelles pour extraire les caractéristiques spatiales. Le deuxième chemin utilise une pyramide de pooling de différentes résolutions pour extraire les caractéristiques contextuelles. Les deux chemins sont ensuite fusionnés dans une seule sortie qui est utilisée pour produire la segmentation finale.

Figure 20 Image segmentée avec le modèle BiseNet

3.3.2. Detectron2

Detectron2 [8] est une bibliothèque de vision par ordinateur open-source développée par Facebook AI Research (FAIR) pour la détection d'objets, la segmentation sémantique et d'autres tâches de vision par ordinateur. Detectron2 est une amélioration de la bibliothèque Detectron originale, avec une architecture de code réécrite pour offrir des performances plus rapides et une plus grande flexibilité.

Le modèle Detectron2 utilise une architecture de réseau de neurones convolutionnels profonds appelée "Faster R-CNN" (R-CNN signifie "Region-based Convolutional Neural Network"). Cette architecture utilise une approche à deux étapes pour la détection d'objets. Dans la première étape, des régions d'intérêt sont proposées dans l'image à l'aide d'une méthode appelée "Region Proposal Network" (RPN). Dans la deuxième étape, les régions proposées sont classifiées et raffinées pour identifier les objets dans l'image.

En plus de Faster R-CNN, Detectron2 prend en charge d'autres architectures de détection d'objets tels que RetinaNet, Mask R-CNN, Cascade R-CNN, etc. Il permet également la segmentation sémantique en utilisant des architectures de réseau de neurones tels que DeepLabv3 et Panoptic FPN.

Detectron2 a été conçu pour être hautement modulaire et facile à utiliser. Il fournit une API claire et concise pour entraîner, tester et déployer des modèles de détection d'objets et de segmentation sémantique. Il prend également en charge le transfert de connaissances, ce qui signifie que les modèles pré-entraînés peuvent être utilisés pour des tâches similaires avec des ensembles de données différents, permettant ainsi une meilleure utilisation des données et une meilleure performance des modèles.

Pour conclure, Detectron2 est une bibliothèque de vision par ordinateur open-source développée par Facebook AI Research qui utilise des architectures de réseaux de neurones convolutifs pour la détection d'objets et la segmentation sémantique. Il est hautement modulaire et facile à utiliser, avec une API claire et concise pour entraîner, tester et déployer des modèles. Il prend également en charge le transfert de connaissances pour une meilleure utilisation des données et une meilleure performance des modèles.

3.3.3. Mask R-CNN

Mask R-CNN [9]est une architecture de réseau de neurones convolutionnels profonds pour la détection d'objets et la segmentation sémantique. Elle a été introduite pour la première fois en 2017 par Kaiming He, Georgia Gkioxari, Piotr Dollár et Ross Girshick dans leur article de recherche intitulé "Mask R-CNN".

Mask R-CNN est basé sur la méthode Faster R-CNN, qui est une architecture de détection d'objets à deux étapes. Dans Faster R-CNN, les régions d'intérêt (ROI) sont proposées à l'aide d'un module de proposition de région (RPN), puis les caractéristiques de chaque ROI sont extraites et utilisées pour classifier et localiser les objets.

Dans Mask R-CNN, la méthode Faster R-CNN est étendue pour inclure une troisième étape pour la segmentation sémantique. Cela permet à l'architecture de détecter les objets et de segmenter chaque objet en même temps. La méthode utilisée pour la segmentation sémantique est la segmentation de masque, qui consiste à générer un masque binaire pour chaque objet détecté. Le masque indique quels pixels de l'image appartiennent à l'objet et quels pixels ne lui appartiennent pas.

La méthode utilisée pour la segmentation de masque dans Mask R-CNN est appelée RoIAlign. RoIAlign est une méthode qui permet d'extraire des caractéristiques précises pour chaque pixel dans le masque binaire en utilisant les caractéristiques de la carte de fonctionnalités extraites pour chaque ROI. Cette méthode utilise des interpolations bilinéaires pour calculer les valeurs des caractéristiques pour chaque pixel dans le masque.

Le réseau de neurones dans Mask R-CNN est composé de deux parties principales : une partie d'extraction de caractéristiques basée sur des couches de convolution profondes pré-entraînées, et une partie de tête de réseau qui effectue la détection d'objets et la segmentation sémantique. La partie de tête de réseau est composée de plusieurs branches, dont une branche pour la classification, une branche pour la localisation et une branche pour la segmentation de masque.

Pour conclure, Mask R-CNN est une architecture de réseau de neurones convolutionnels profonds pour la détection d'objets et la segmentation sémantique. Elle est basée sur la méthode Faster R-CNN et utilise une méthode de segmentation de masque appelée RoIAlign pour générer un masque binaire pour chaque objet détecté. Le réseau de neurones est composé d'une partie d'extraction de caractéristiques pré-entraînée et d'une partie de tête de réseau qui effectue la détection d'objets et la segmentation sémantique.

3.4. Préparation des données

La base de données CelebA-HQ Dataset 1024x1024 est une collection de plus de 30 000 images de visages de célébrités au format JPEG, chacune d'une résolution de 1024x1024 pixels. Cette base de données est largement utilisée dans la recherche en vision par ordinateur et en intelligence artificielle pour la reconnaissance faciale, la détection de visages, la génération d'images et d'autres tâches connexes. Les images de la base de données Celeb 1024x1024 ont été collectées à partir de sources publiques telles que des sites de fans de célébrités et des réseaux sociaux. Cette base de données est particulièrement utile pour les tâches de génération d'images de haute qualité, car elle contient des images de haute résolution et une grande variété de poses et d'expressions faciales.

Etapes de préparation des données

La préparation des données celebA implique plusieurs étapes. Tout d'abord, nous avons chargé les 30 000 images de l'ensemble de données celebA. Ensuite, nous avons découpé le visage à partir de chaque image en utilisant des techniques de détection de visage. À partir de chaque visage découpé, nous avons segmenté les dents et la cavité buccale en utilisant des algorithmes de segmentation d'image. Ensuite, nous avons découpé la cavité buccale à partir de l'image du visage découpé en utilisant les coordonnées du polygone de la cavité buccale. Enfin, nous avons ajouté des bordures à l'image de contours et à l'image de la cavité buccale pour faciliter l'apprentissage du modèle pix2pix. Après toutes ces étapes, nous avons obtenu 15000images.

Après la préparation des données, il est courant de diviser le jeu de données en images d'entraînement, images de test. Dans notre cas, un ensemble de 15 000 images a été utilisé. Les images d'entraînement sont utilisées pour entraîner le modèle, c'est-à-dire pour lui permettre d'apprendre les motifs et les relations dans les données.

Les images de test sont réservées pour évaluer les performances du modèle sur des données qu'il n'a jamais vues auparavant. Elles servent à mesurer la capacité du modèle à généraliser et à estimer sa performance réelle sur des exemples inédits. L'utilisation d'images de test distinctes est importante pour éviter le surapprentissage, où le modèle peut trop s'ajuster aux données d'entraînement spécifiques.

1) Définition de la méthode "crop_face" qui prend en paramètre une image de visage, détecte les visages dans l'image et renvoie l'image du visage découpé.

Figure 21:Visage découpé à partir de l'image initiale

2) Définition de la méthode "crop_mouth" qui prend en paramètre l'image de visage, le masque de la cavité buccale et les coordonnées du polygone de la cavité buccale, puis découpe la cavité buccale à partir de l'image de visage.

Figure 22: cavité buccale découpé

3) Application du modèle de segmentation des dents :

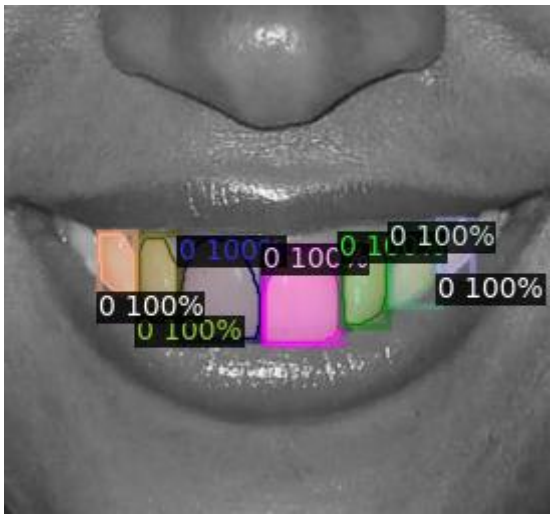

Figure 23: segmentation des dents

4) Segmenter et découper le contour de la bouche à partir de l'image de la cavité buccale :

Figure 24: sourire découpé

5) Dessiner les contours du sourire :

Figure 25: image de contours (croquis sketch)

6) concaténer les deux images concours et sourire pour former l'image d'entrée pour le modèle pix2pix

Figure 26 : image finale pour l'entrée du modèle pix2pix

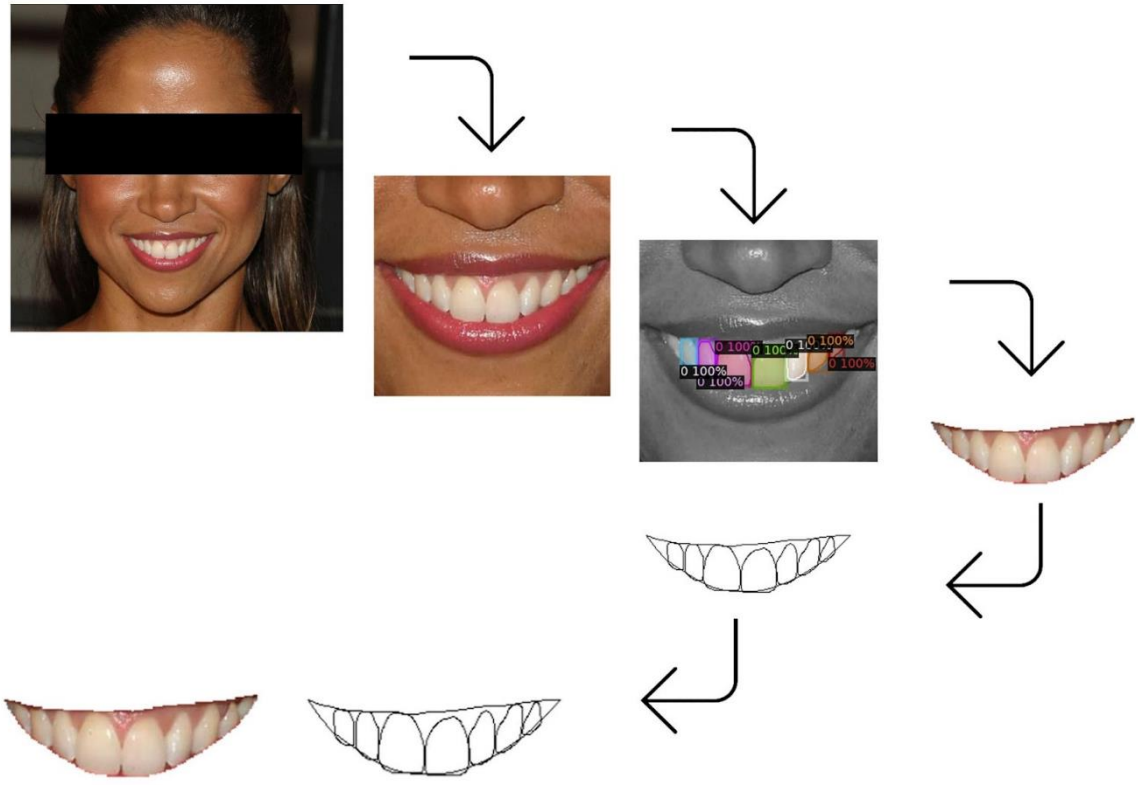

Figure 27: pipeline du préparations des données

3.5. Pix2pix pour générer un Smile Design

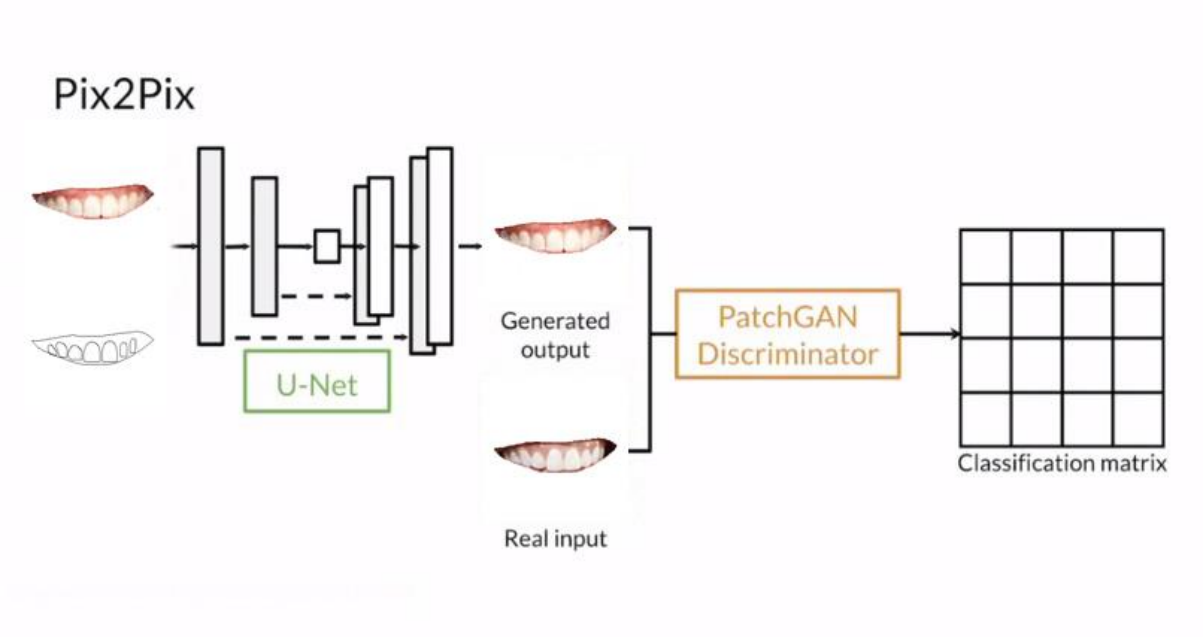

Figure 28:: architecture pix2pix

La figure 28 illustre la génération d'un Smile Design, le modèle pix2pix prend en entrée une paire d'image (le sketch et ground truth). Après génération de l'image le discriminateur prend en entrée l'image générée par le générateur et le ground truth pour procéder à la classification.

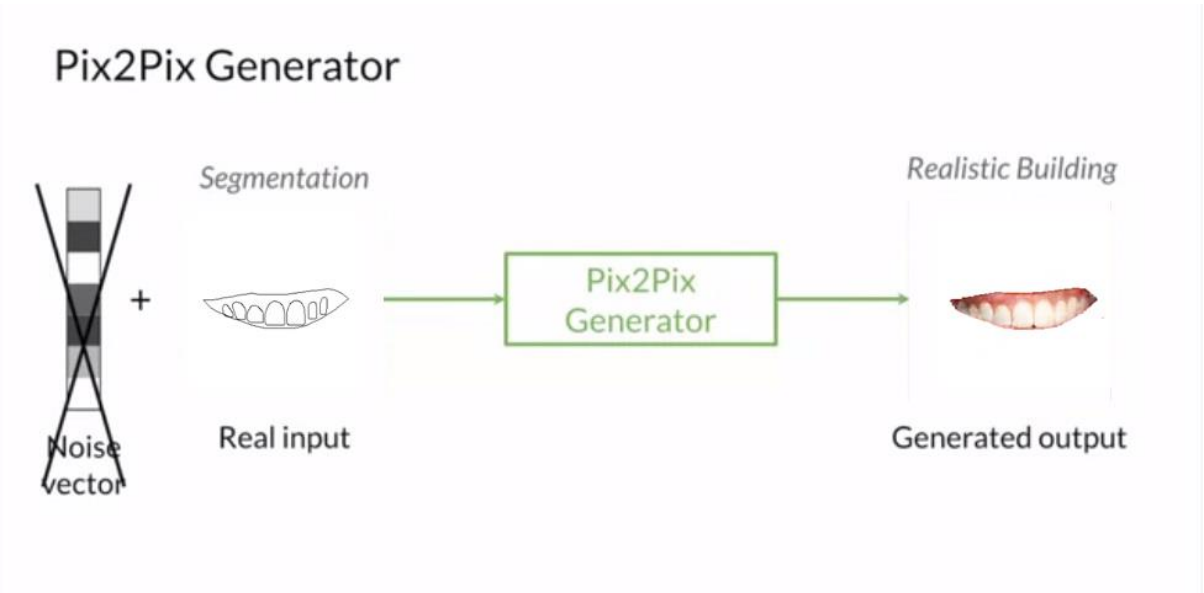

Figure 29: architecture du générateur

La figure 29 illustre l'opération de génération d'un sourie à partir d'un sketch.

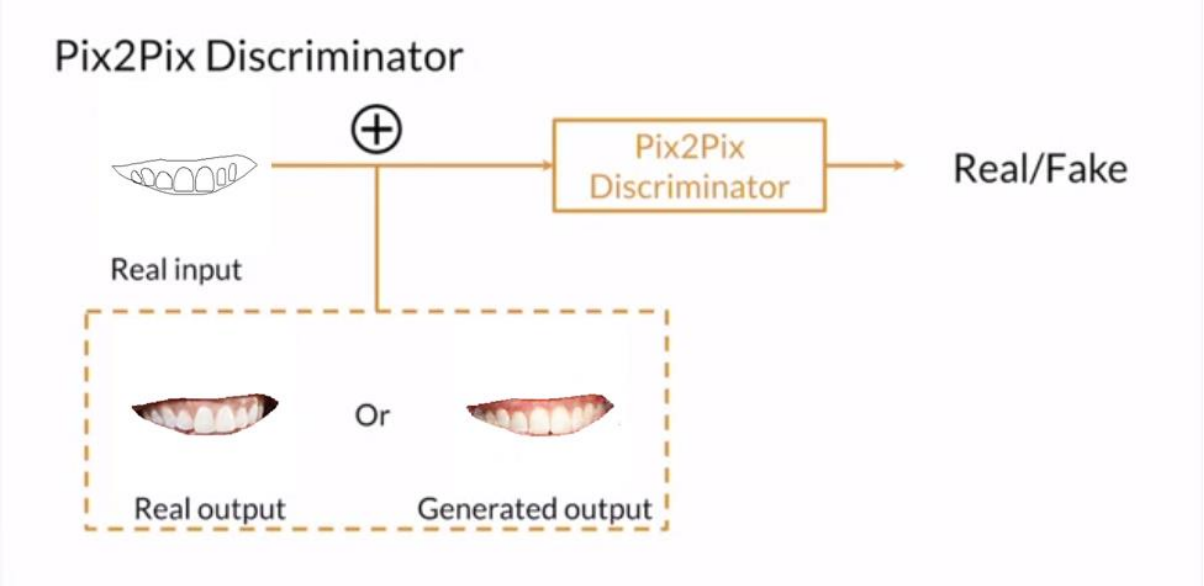

Figure 30: architecture du discriminateur

La figure 30 illustre la phase de classification du discriminateur pix2pix.

3.6. La phase d'entrainement

La phase d'entraînement du modèle Pix2Pix est une étape importante pour permettre au modèle d'apprendre à générer des images réalistes et cohérentes à partir des données d'entrée. Dans cette phase, nous avons utilisé ' N'nombres d'images pour entraîner le modèle, avec un batch size fixé à 'B' et un nombre d'epochs établi à 'E'. Pendant l'entraînement, le modèle a essayé de minimiser la différence entre les images générées et les images réelles en ajustant les poids des différentes couches du réseau de neurones. Ce processus a été répété pour chaque epoch jusqu'à ce que la perte converge vers une valeur minimale. Le batch size et le nombre d'epochs ont été choisis en fonction de la taille du jeu de données et de la complexité du modèle, afin de permettre une convergence efficace et rapide du modèle tout en évitant un surajustement. En ajustant ces hyperparamètres, nous avons réussi à entraîner un modèle Pix2Pix performant qui est capable de générer des images de haute qualité à partir de données d'entrée.

Nous avons entrainé le modèle sur trois expérimentations tout en changement à chaque fois les hyperparamètres du modèle.

En variant à chaque fois le nombres des images d'entrainement et ne nombres des epochs.

3.6.1. Résultat après la première phase d'entraînement

Nombres d'images d'entraînement = 1000 images

Nombre d'epoch = 200 epochs

Nous avons sauvegardé des échantillons d'images générées pendant la phase d'entraînement pour suivre l'évolution de notre modèle.

Images générées pendant l'entrainement

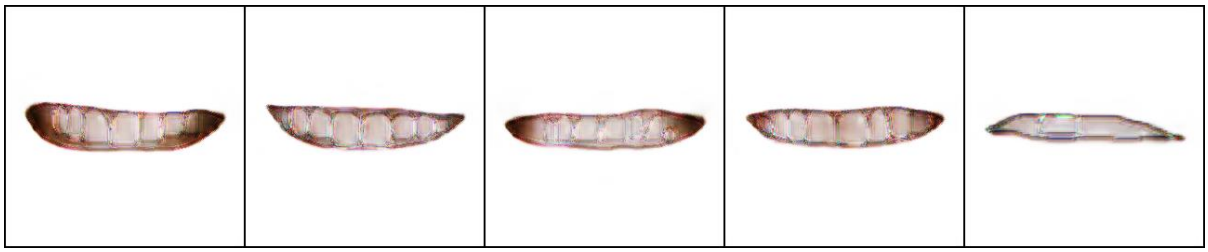

Figure 31 : Images générées après 20 epochs

Figure 32 : Images générées après 40 epochs

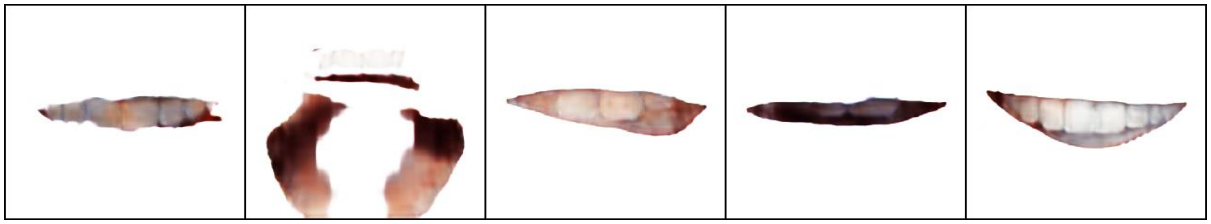

Figure 33 : Images générées après 60 epochs

Figure 34:Images générées après 80 epochs

Figure 35: Images générées après 100 epochs

Figure 36 : Images générées après 120 epochs

Figure 37 : Images générées après 120 epochs

Figure 38: Images générées après 140 epochs

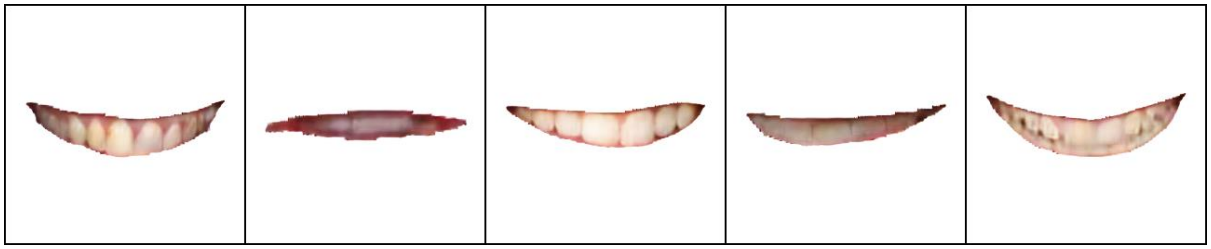

Figure 39: Images générées après 160 epochs

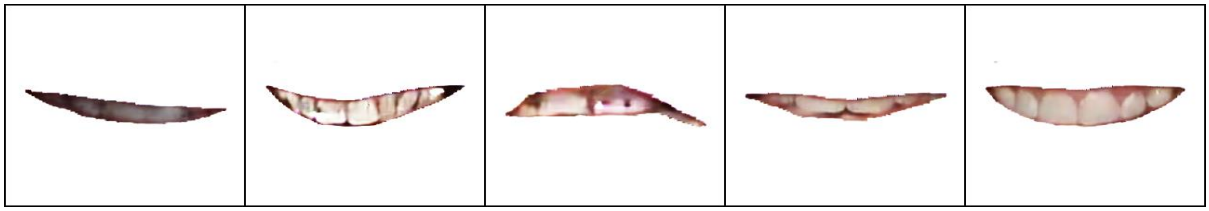

Figure 40: Images générées après 180 epochs

Les fonctions de pertes

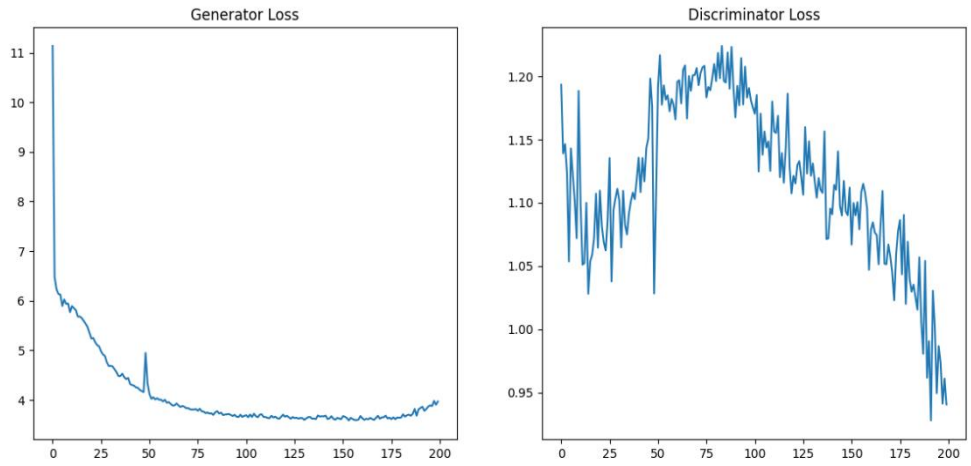

Figure 41: fonctions de pertes du générateur et du discriminateur pour 200 epochs

Interprétation

Après observation et analyse des résultats nous avons remarqué que :

Dans le cas présent, l'expérience a permis de constater que le modèle entraîné présentait une convergence et une stabilité temporaire. En effet, le modèle a été entraîné sur un ensemble de données de 1000 images et a réussi à atteindre une précision acceptable lors de l'entraînement.

Lors de l'analyse d'une expérience du modèle Pix2Pix, on a observé que la fonction de perte du générateur convergeait, mais qu'il y avait une légère augmentation de la perte vers la fin de l'entraînement.

D'autre part, la fonction de perte du discriminateur convergeait vers 0 au lieu de converger vers 1, comme on s'y attendrait normalement. Cela pourrait être le résultat d'un déséquilibre entre le générateur et le discriminateur.

En outre, l'expérience a également montré que la qualité des croquis influence la qualité des images générées par le modèle. En effet, il a été observé que les images générées par le modèle étaient plus nettes et plus détaillées lorsque les croquis étaient de meilleure qualité. Cela suggère que la qualité des croquis peut être un facteur important dans la performance globale.

3.6.2. Résultat après la deuxième phase d'entraînement

Dans la deuxième expérimentation, on envisage de modifier le nombre d'images dans la base d'entraînement ainsi que le nombre d'époques.

Nombres d'images d'entraînement = 10000 images

Nombre d'epoch = 150 epochs

Images générées pendant l'entrainement

Figure 42:Images générées après 20 epoch

Figure 43:Images générées après 40 epochs

Figure 44:Images générées après 60 epochs

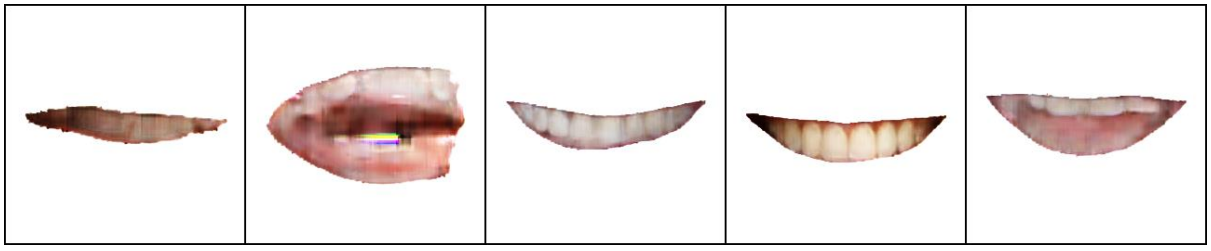

Figure 45:Images générées après 80 epochs

Figure 46:Images générées après 100 epochs

Figure 47:Images générées après 120 epochs

Figure 48:Images générées après 140 epochs

Les fonctions de pertes

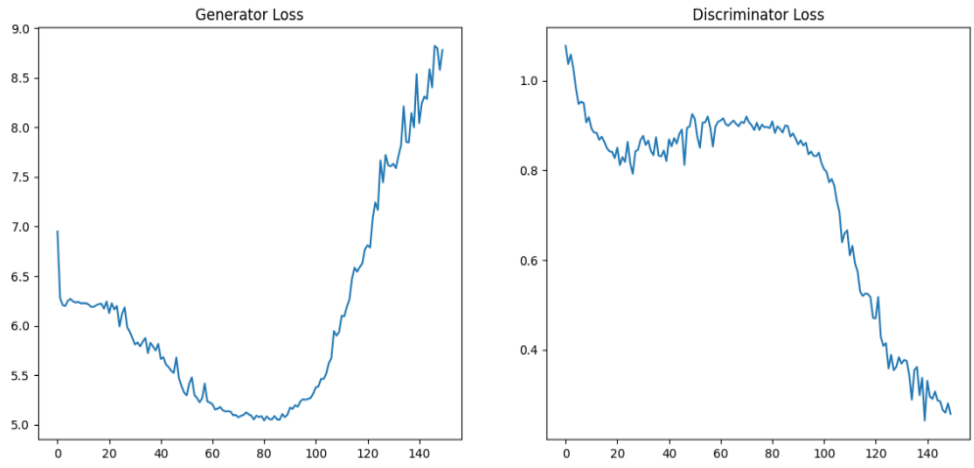

Figure 49: fonctions de pertes du générateur et du discriminateur pour 150 epochs

Interprétation

Tout d'abord, on constate que la fonction de perte du générateur converge initialement, ce qui indique que le générateur parvient à apprendre à générer des sorties qui correspondent aux images cibles. Cependant, au cours de l'entraînement, la perte du générateur augmente de manière exponentielle. Cela suggère que le générateur a des difficultés à maintenir la qualité des sorties et que ses prédictions s'éloignent de plus en plus des images cibles attendues.

En examinant la fonction de perte du discriminateur, on observe qu'elle converge vers zéro. Cela signifie que le discriminateur classe toutes les sorties comme des images générées par le générateur, ce qui peut expliquer l'augmentation exponentielle de la perte du générateur. Lorsque le discriminateur ne parvient pas à distinguer les vraies paires d'entrée-sortie des paires générées, le générateur peut avoir du mal à apprendre de ces signaux contradictoires et à générer des sorties de qualité.

Afin de corriger cette perte exponentielle, il est recommandé de rechercher les données anormales au sein de nos lots de données. Si ces anomalies semblent être des données aberrantes, il est essentiel de s'assurer que ces anomalies sont réparties de manière uniforme entre les lots en mélangeant les données. Cette approche permettra de remédier à l'augmentation de la perte observée lors de l'expérimentation.

3.6.3. Résultat après la troisième phase d'entraînement

Dans la troisième expérimentation, on envisage de modifier à nouveau le nombre d'images dans la base d'entraînement ainsi que le nombre d'époques.

Nombres d’images d'entraînement = 1000 images

Nombre d’epoch = 100 epochs

Images générées pendant l'entrainement

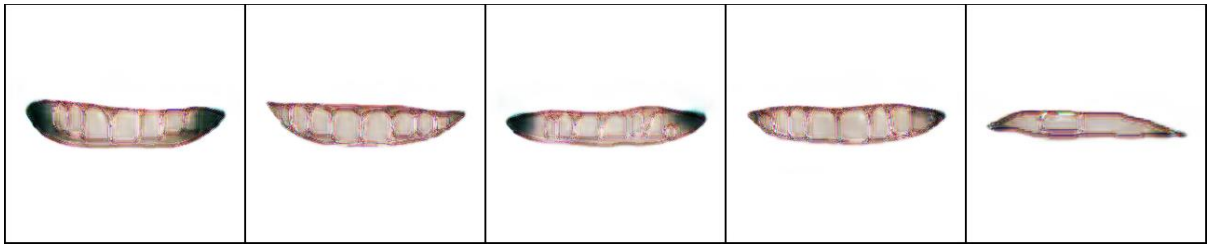

Figure 50:Images générées après 20 epochs

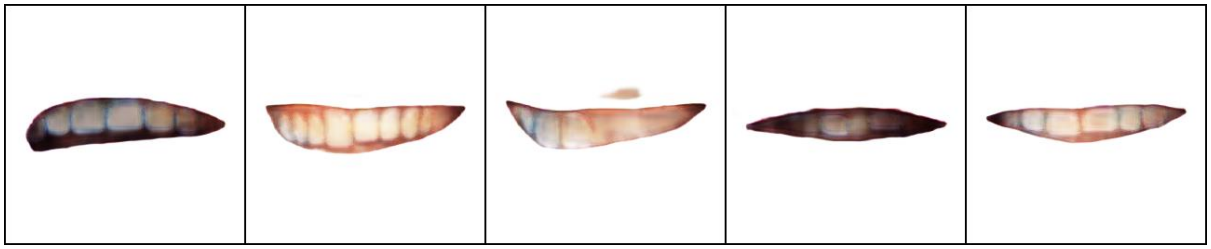

Figure 51:Images générées après 40 epoch

Figure 52:Images générées après 60 epochs

Figure 53:Images générées après 80 epochs

Figure 54:Images générées après 100 epochs

Les fonctions de pertes

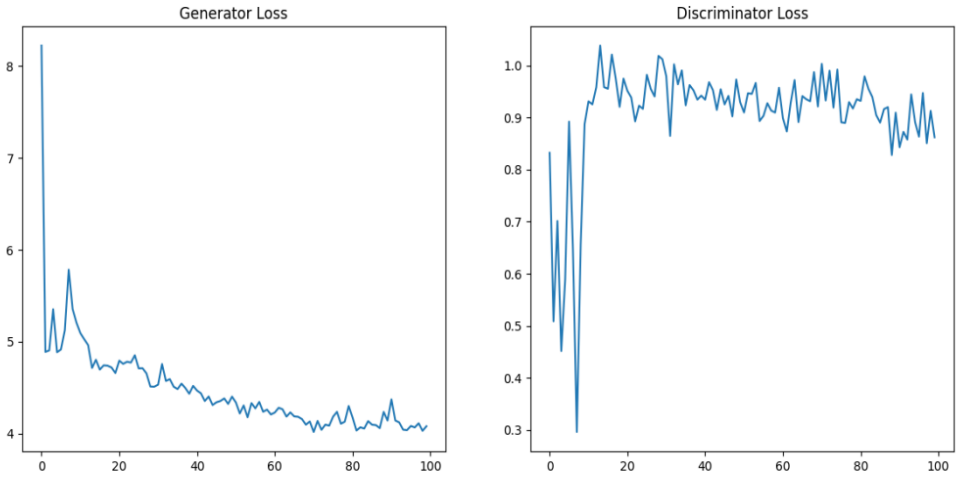

Figure 55 : fonctions de pertes du générateur et du discriminateur pour 100 epochs

Interprétation

Après observation et analyse des nouveaux résultats nous avons remarqué que :

L'interprétation de l'expérimentation met en évidence des résultats encourageants, notamment une convergence de la fonction de perte du générateur. Cela indique que le modèle a réussi à apprendre efficacement à partir des données d'entraînement.

Nous avons remarqué aussi la courbe loss du discriminateur converge vers 1 ce qui explique qu'il classe toutes les entrées comme étant des images réelles. C'est-à-dire que le discriminateur ne parvient pas à distinguer les vraies paires d'entrée-sortie des paires générées.

De plus, les résultats visuels obtenus sont satisfaisants, ce qui démontre que le modèle parvient à générer des sorties de qualité. Cette convergence vers une perte minimale et les résultats visuels satisfaisants suggèrent que le modèle a bien assimilé les caractéristiques des données d'entraînement et est capable de produire des sorties cohérentes et convaincantes. Ces résultats positifs témoignent de l'efficacité de l'approche expérimentale et de la capacité du modèle à accomplir la tâche spécifique pour laquelle il a été conçu.

3.7. La phase de test

La phase de test du modèle Pix2Pix est une étape cruciale pour évaluer la performance et la capacité de généralisation du modèle à de nouvelles données. Pendant cette phase, le générateur est utilisé pour générer de nouvelles images à partir d'images d'entrée inédites. Cela permet de vérifier si le modèle a appris avec succès les correspondances entre les images d'entrée et les images cibles.

Lors du test, le générateur prend une image d'entrée et la transforme en une image cible prédite. Cette image prédite est ensuite comparée à l'image cible réelle pour évaluer la qualité de la prédiction. Différentes mesures d'évaluation peuvent être utilisées, telles que la distance entre les pixels ou des mesures de similarité structurelle, pour quantifier l'écart entre l'image prédite et l'image cible réelle.

La phase de test permet également d'identifier les éventuels problèmes de généralisation du modèle. Si le modèle ne parvient pas à générer des images de qualité à partir de nouvelles entrées, cela peut indiquer un manque de diversité dans les données d'entraînement, un surapprentissage ou une mauvaise conception du modèle.

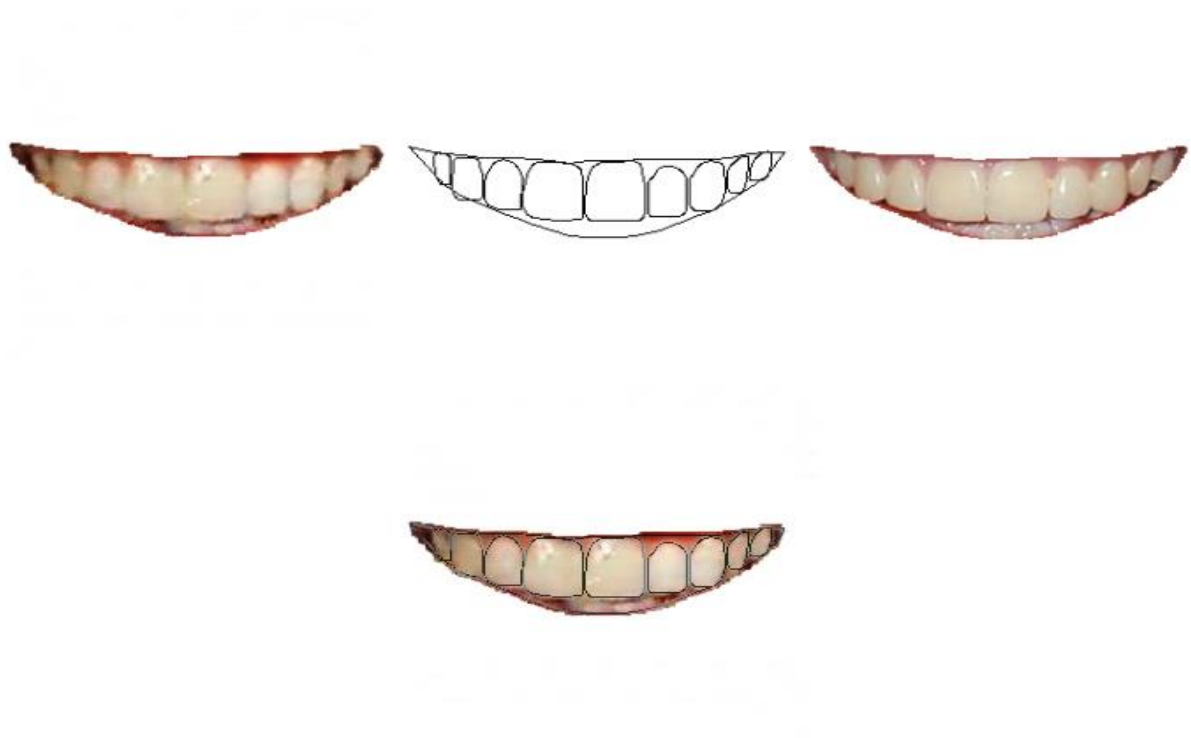

Figure 56: Exemple 1 de superposition du croquis sur l'image générée

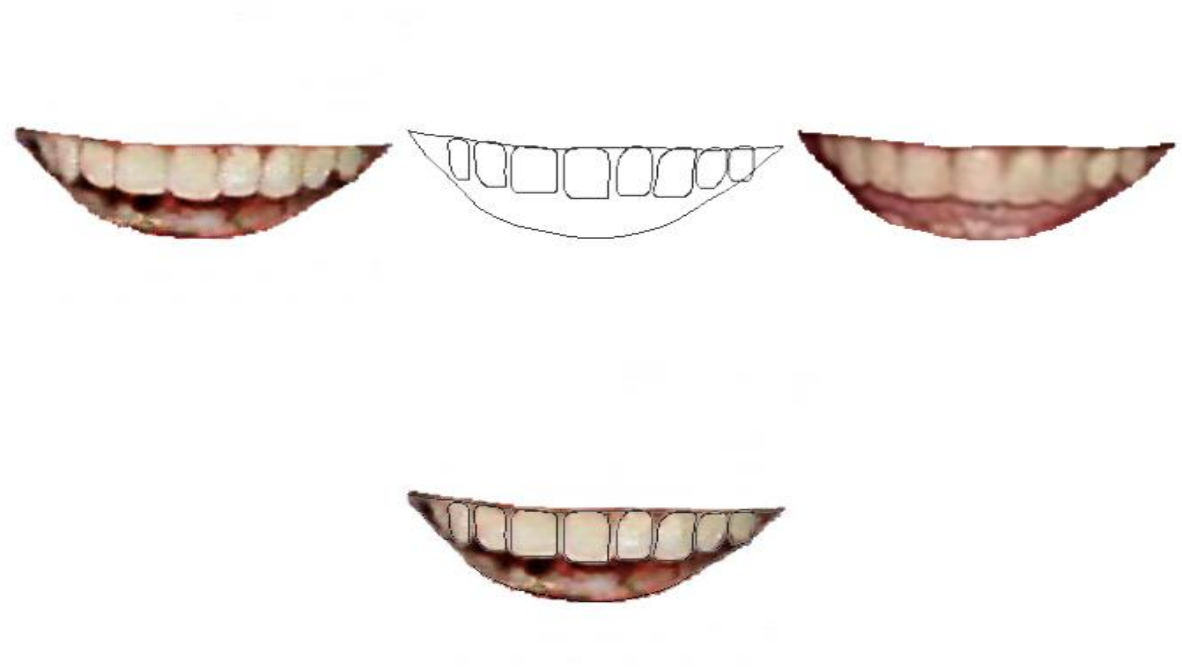

Figure 57: Exemple 2 de superposition du croquis sur l'image générée

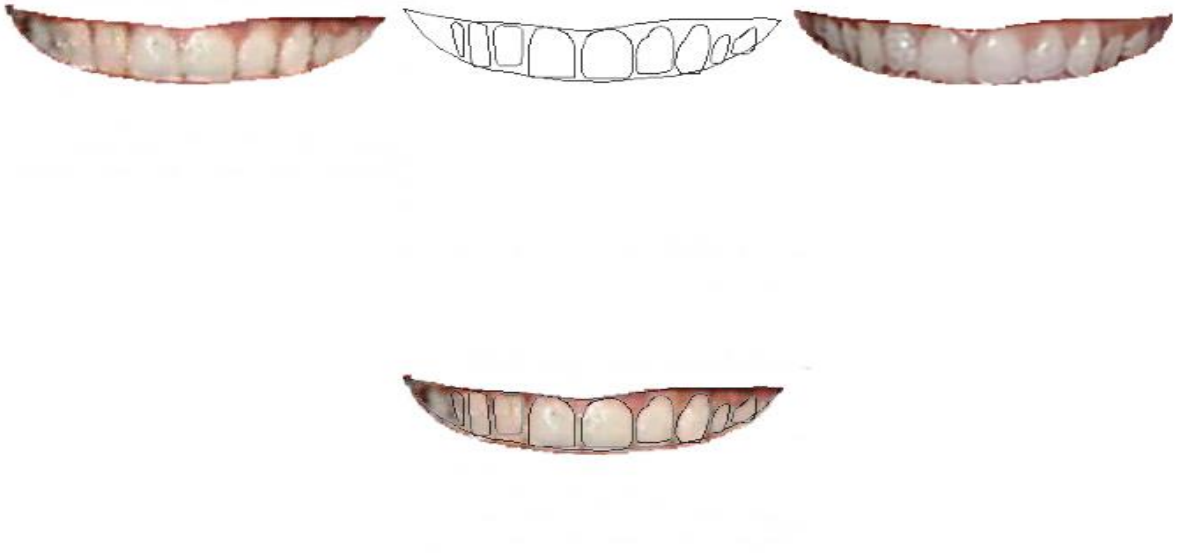

Figure 58: Exemple 3 de superposition du croquis sur l'image générée

3.8. Conclusion

En conclusion, l'entraînement et l'expérimentation du modèle Pix2Pix pour générer un Smile Design ont été un processus complexe et exigeant, mais qui a finalement permis d'obtenir des résultats satisfaisants. Le modèle a été entraîné sur un ensemble de données allant de 1000 à 10000 images d'entrée et d'epochs allant de 100 à 200 epochs.

L'utilisation d'un discriminateur PatchGAN a été utilisée pour améliorer la qualité des images générées. Des hyperparamètres tels que le nombre d'epochs, le taux d'apprentissage et la taille des lots ont été ajustés pour optimiser la performance du modèle.

Nous avons rencontré quelques problèmes de convergence lors des phases de l'expérimentations vu que l'état de stabilité du GANs est temporaire, et qu'il est très sensibles aux hyperparamètres du modèle.

Finalement, le modèle a été testé en utilisant des images d'entrée inédites, et les résultats ont montré que le modèle était capable de générer des Designs de sourire réalistes et cohérents avec les images cibles. Bien que des améliorations puissent être apportées, cette expérience démontre le potentiel des modèles Pix2Pix pour la génération d'images réalistes et peut servir de base pour de futures recherches dans ce domaine.

Conclusion générale

En conclusion, ce mémoire a exploré l'utilisation du modèle de génération d'images pix2pix pour la création de Designs de sourire en spécifiant la texture et la couleur des dents à générer. Les résultats ont montré que le modèle est capable de générer des Designs de sourire réalistes et esthétiquement plaisants en utilisant un ensemble de données approprié et en optimisant les paramètres du modèle.

Cependant, le modèle présente également certaines limites, notamment en termes de qualité de l'image générée. Bien que le modèle puisse générer des Designs de sourire cohérents avec les données d'entraînement, la qualité de l'image générée peut être limitée. Pour remédier à cela, des recherches futures pourraient se concentrer sur l'utilisation de modèles de génération d'images plus avancés tels que Pix2PixHD ou Super Resolution pour améliorer la qualité de l'image générée.

En guise de perspective, il serait intéressant d'explorer la possibilité d'ajouter au modèle la fonctionnalité de choix de la nuance de couleur pour la génération des dents. Cette fonctionnalité pourrait être particulièrement utile pour les dentistes et les orthodontistes qui souhaitent créer des Designs de sourire personnalisés pour leurs patients en fonction de leurs préférences de couleur. L'ajout de cette fonctionnalité nécessiterait une adaptation du modèle et l'ajout d'un nouvel ensemble de données d'entraînement pour permettre au modèle d'apprendre à générer des images avec différentes nuances de couleur de dents. Cette perspective ouvre des possibilités intéressantes pour la personnalisation des Designs de sourire et pour l'application de la génération d'images dans le domaine esthétiques dentaires.

Bibliographies

[1] Ian J. Goodfellow, "Generative Adversarial Nets," 10 Jun 2014.

[2] l. l. Cédric, "https://cedric.cnam.fr/vertigo/Cours/RCP211/reseaux-generatifs-antagonistes.html," [Online].

[3] M. Mirza, "Conditional Generative Adversarial Nets," 6 Nov 2014 .

[4] A. N. N. ARUN SOLANKI, GENERATIVE ADVERSARIAL NETWORKS FOR IMAGE-TO-IMAGE TRANSLATION, Mara Conner, 2021.

[5] P. I. J.-Y. Z. T. Z. A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 26 Nov 2018.

[6] P. F. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 18 May 2015.

[7] C. Yu, "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation," 2 Aug 2018.

[8] Y. W. a. A. K. a. F. M. and, "Detectron2," 2019.

[9] K. H. G. G. P. D. R. Girshick, "Mask R-CNN," 24 Jan 2018.

[10] H. M.-C. Xavier Marsault1, "Les GANs : stimulateurs de créativité en phase d'idéation," Web of Conferences 147, 06003 (2022).

[11] P. G. N. G. Swetava Ganguli, "GeoGAN: A Conditional GAN with Reconstructionand Style Loss to Generate Standard Layer of Mapsfrom Satellite Images".

[12] M. A. L. Bottou, "TOWARDS PRINCIPLED METHODS FOR TRAINING GENERATIVE ADVERSARIAL NETWORKS," 17 Jan 2017.