UQAC
Université du Québec
à Chicoutimi

DÉTECTION D'IMAGES FACIALES GÉNÉRÉES PAR STYLEGAN À L'AIDE DES TRANSFORMATEURS DE VISION ET DE L'ATTENTION LATENTE

PAR SOKHNA BALLY TOURE

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI EN VUE DE L'OBTENTION DU GRADE DE MAÎTRISE ÈS SCIENCES (M. SC.) EN INFORMATIQUE

QUÉBEC, CANADA

RÉSUMÉ

L'évolution rapide des réseaux antagonistes génératifs (GAN), en particulier StyleGAN, a conduit à une augmentation sans précédent des images synthétiques hautement réalistes. Bien que cette technologie ouvre des perspectives passionnantes dans divers domaines, elle pose des défis importants en matière de sécurité numérique et d'authenticité du contenu.

Pour répondre à ce problème, notre étude se concentre sur le développement d'une méthode robuste de détection des images faciales générées par StyleGAN. Nous proposons un modèle Vision Transformers (ViT) optimisé qui tire parti de l'apprentissage par transfert et intègre un module d'attention latente. Cette approche améliore les capacités de détection du modèle et permet d'identifier efficacement les images générées par StyleGAN.

Une évaluation complète qui comprend de vastes ensembles de données d'images réelles et générées, démontre les performances remarquables du modèle. Le modèle proposé atteint une précision de 99,83%, un AUC de 1 et un score F1 de 0,9983. Le modèle présente de fortes capacités de généralisation sur des ensembles de données externes, ce qui confirme son efficacité dans divers scénarios de détection de deepfake. En outre, grâce à l'intégration tardive de l'attention, le coût de calcul peut être réduit de 42%, atteignant une réduction de 85% pour un ensemble de données spécifique.

Les résultats obtenus, comparés à ceux de six méthodes de référence, montrent que notre approche offre de meilleures performances pour détecter les deepfakes générés par StyleGAN. Cette méthode contribue ainsi efficacement à l'authentification des contenus numériques et à la détection d'images synthétiques.

ABSTRACT

The rapid advancement of Generative Adversarial Networks (GANs), particularly StyleGAN, has led to an unprecedented increase in highly realistic synthetic images. While this technology opens up exciting opportunities across various fields, it poses significant challenges to digital security and content authenticity.

To address this issue, our study focuses on developing a robust method for detecting facial images generated by StyleGAN. We propose an optimized Vision Transformers (ViT) model that leverages transfer learning and incorporates a latent attention module. This approach enhances the model's detection capabilities, effectively identifying StyleGAN-generated images.

A comprehensive evaluation, which includes large datasets of real and generated images, demonstrates the model's remarkable performance. The proposed model achieves an accuracy of 99.83%, an AUC of 1, and an F1-score of 0.9983. Furthermore, the model exhibits strong generalization abilities on external datasets, confirming its efficacy in various deepfake detection scenarios. Furthermore, due to the late attention integration, the computational cost can be reduced by 42%, achieving an 85% reduction for a specific dataset.

We extensively validated our approach on three diverse StyleGAN-generated deepfake datasets and compared its performance to six baseline methods, demonstrating its superiority in detecting StyleGAN-generated deepfakes and its contribution to digital content authentication and synthetic image detection.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

AUC Area Under the ROC Curve
BERT Bidirectional Encoder Representations from Transformers
CGAN Conditional Generative Adversarial Network
CNN Convolutional NeurAl Network
CycleGAN Cycle Generative Adversarial Network
DCGAN Deep Convolutional Generative Adversarial Network
DiscoGAN Discover Cross-Domain Relations with Generative Adversarial Network
DL Deep Learning
FN Faux négatifs
FP Faux positifs
GAN Generative Adversarial Networks
GPT Generative Pre-trained Transformer
IA Intelligence Artificielle
iGPT Image Generative Pre-trained Transformer
LN Layer Normalization
MHA Multi-Head Attention
MHSA Multi-Head Sel-Attention
ML Machine Learning
MLM Masked Language Modeling
MLP Multi-Layer Perceptron
NLP Natural Language Preprocessing
ProGAN Progressive Growing Generative Adversarial Network
PRNU Photo Response Non Uniformity
RGB Red-Green-Blue
ROC Receiver Operating Characteristic
StarGAN Star Generative Adversarial Network
StyleGAN Style Generative Adversarial Network
ViT Vision Transformers
VN Vrais négatifs
VP Vrais positifs

DÉDICACE

Á mon Papa, qui n'a pas eu l'occasion de voir ses prières se réaliser ; j'espère, depuis le Paradis, que tu es fier de ta fille chérie,

Á ma Maman, pour tous ses sacrifices, son amour, sa tendresse, son soutien et ses prières tout au long de mes études,

Á mes chers frères et sœurs, pour leurs encouragements permanents, et leur soutien moral,

Á mes chers oncles, pour leur appui et leur encouragement,

Á Mme Seynabou Diouf pour son soutien tout au long de mon parcours universitaire,

Á mon tuteur et oncle M. Aziz Mbacké, sans qui tout ceci ne serait possible,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

REMERCIEMENTS

Au nom d'ALLAH le Tout Miséricordieux le Très Miséricordieux

Je rends grâce à ALLAH (Exalté soit-IL) de m'avoir gratifié le privilège d'éditer ce mémoire. Prières sur son prophète **Muhammad (PSL)**.

Je tiens à exprimer ma profonde gratitude à plusieurs personnes qui ont contribué à la réalisation de ce travail.

Tout d'abord, j'exprime mes plus sincères remerciements à mon mentor **Pr Haïfa Nakouri**, dont la patience et les précieux conseils et encouragement ont grandement enrichi ma réflexion.

Mes sincères remerciements vont également à mon directeur de recherche Pr Bob-Antoine Jerry Mélénas, ainsi qu'aux professeurs et aux membres du personnel du Département d'informatique et de mathématique de l'UQAC, qui m'ont donné les outils nécessaires pour réussir ma formation.

Je remercie tout particulièrement mon oncle Serigne Fallou Mbacké, pour sa confiance indéfectible et son précieux soutien.

Je suis également reconnaissant pour l'amitié indéfectible de mes amis et à ma famille pour leurs prières et encouragements.

Enfin, j'exprime ma gratitude à M. Aziz Mbacké qui s'est démené pour contribuer à ma réussite scolaire.

Je leur suis infiniment reconnaissante.

AVANT-PROPOS

Ce mémoire représente le résultat d'un parcours académique riche en apprentissage. J'ai eu l'occasion de plonger en profondeur dans un sujet qui me passionne : l'intelligence artificielle. Grâce aux cours théoriques, j'ai développé de solides compétences en IA. Cela m'a donné la chance d'examiner l'impact de l'IA sur le monde numérique, notamment les modèles de génération d'images comme les GAN. Avec les progrès continus des architectures GAN, nous entrons dans l'ère des StyleGAN, capables de produire des images si réalistes qu'elles sont difficiles à différencier des vraies.

L'objectif de ce travail est de proposer un modèle qui permet de détecter les images générées par StyleGAN et d'analyser l'explicabilité de celui-ci. Cela vise à offrir une meilleure compréhension des choix effectués par le modèle.

Mon expérience de recherche a été jalonnée de défis et de stress, mais aussi de découvertes, de réussites et d'échecs qui ont enrichi mon parcours. Ce rapport est le résultat de nombreuses heures de travail et de réflexion, alimentées par une passion profonde.

Étant donné que tout travail humain peut être perfectionné, je serai ravi de recevoir vos retours, vos critiques et vos suggestions, lesquels m'aideront à continuer mes recherches et à élargir mes connaissances.

Je souhaite que cette modeste contribution soit bénéfique pour les études à venir et qu'elle inspire d'autres travaux dans ce domaine.

INTRODUCTION

CONTEXTE

L'émergence des modèles génératifs, et notamment des réseaux antagonistes génératifs (GAN), a marqué une révolution dans la création d'images photoréalistes. Introduit en 2014 par Goodfellow *et al.* (2014), les GANs sont basés sur une architecture composée de deux réseaux neuronaux concurrents : un générateur, qui génère des images réalistes, et un discriminateur, qui est chargé de distinguer les images réelles des images synthétiques. Cette rivalité conduit à une amélioration continue de la qualité des images générées, atteignant un réalisme impressionnant, comme le montre la Figure 4.2. Parmi les variantes de GAN, StyleGAN se distingue par sa capacité à manipuler les styles et les caractéristiques faciales à différentes échelles, devenant ainsi une référence pour la génération de visages réalistes.

PROBLÉMATIQUE

Ces avancées technologiques permettent aujourd'hui de produire des images synthétiques d'une qualité telle qu'il est de plus en plus difficile de les différencier des images réelles. Si elles ouvrent des perspectives prometteuses dans des domaines tels que le divertissement, la publicité ou la médecine, elles soulèvent également des défis majeurs en matière de sécurité et d'audit. Les images synthétiques peuvent être utilisées à des fins malveillantes, telles que la désinformation, l'usurpation d'identité ou la fraude numérique. Face à ces risques, le développement de méthodes de détection robustes est devenu une priorité.

OBJECTIF

Les approches de détection traditionnelles s'appuient principalement sur les réseaux neuronaux convolutionnels (CNN), qui analysent des caractéristiques faciales spécifiques

telles que les yeux, la texture de la peau ou les artefacts récurrents. Cependant, avec l'évolution rapide des architectures GAN, ces méthodes montrent leurs limites. Les modèles modernes, tels que StyleGAN Karras *et al.* (2021), produisent des images si réalistes que la détection de faux visages devient extrêmement complexe. Cela souligne la nécessité de développer des approches innovantes pour identifier les images générées par ces modèles.

Parallèlement, les modèles Vision Transformer (ViT) Dosovitskiy *et al.* (2020) ont récemment démontré des performances exceptionnelles dans diverses tâches de vision par ordinateur. Basés sur des mécanismes d'attention, ils peuvent capturer des dépendances à long terme et des modèles complexes dans les images, ce qui en fait des candidats prometteurs pour la détection d'images synthétiques. On a pour objectif d'optimiser les avantages des transformateurs visuels pour développer un modèle qui permet de détecter de manière efficace les images générées par StyleGAN.

CONTRIBUTIONS

Notre approche combine les avantages du ViT pré-entraîné avec un réglage fin spécifique à la tâche, ce qui nous permet de tirer parti des connaissances générales du modèle tout en l'adaptant à la détection d'anomalies subtiles dans les visages synthétiques.

Nos principales contributions sont les suivantes :

1. L'optimisation du modèle ViT pré-entraîné, dans lequel un module d'attention latente est intégré pour remplacer l'auto-attention multi-têtes. Ce dernier réduit la complexité du modèle tout en se concentrant sur les régions les plus discriminantes pour la détection de fausses images.
2. Une explicabilité des décisions du modèle, utilisant la technique Grad-CAM, quant à savoir si une image est fausse ou non est également proposée. Cela permet de mieux comprendre les éléments qui ont influencé la décision de notre modèle pour la détection.

Pour situer au mieux ces contributions, ce rapport suit la structure suivante : le chapitre 1 aborde les différentes architecture GAN et leur spécifications. Le chapitre 2 parle de l'intelligence artificielle avec les différentes techniques d'apprentissage. Dans le chapitre 3 nous faisons une revue de la littérature sur les différentes méthodes de détection de deepfake. Le chapitre 4 présente notre méthodologie et les détails de l'implémentation de notre modèle. Dans le chapitre 5 on présente les résultats obtenus, l'étude d'ablation, l'explicabilité suivi d'une discussion et des limites du modèle. Et enfin, nous avons la conclusion générale.

CHAPITRE I

LES RÉSEAUX ANTAGONISTES GÉNÉRATIFS

1.1 INTRODUCTION

Un modèle génératif a la capacité de générer de nouvelles données réalistes qui ne ressemblent pas nécessairement aux données d'entraînement initial. La variété des taches et des objectifs à atteindre fait qu'il existe une diversité de modèles génératifs pour répondre à chaque besoin. Et parmi ces modèles nous avons les GANs (Generative Adversarial Networks).

Introduites en 2014 par Goodfellow *et al.* (2014), le modèle GAN entraîne deux réseaux neuronaux à se faire concurrence afin de générer de nouvelles données plus authentiques à partir d'un jeu de données d'entraînement donné. Le terme antagoniste souligne qu'il oppose deux réseaux différents. L'un génère des données nouvelles en utilisant un échantillon de données d'entrée, qu'il modifie de diverses manières Amazon Web Services (2022). L'autre réseau essaie de prédire si les données générées appartiennent au jeu de données d'origine. La Figure 1.1 présente l'architecture de base d'un GAN.

Il existe différents types de modèles GAN en fonction des formules mathématiques utilisées et des différentes manières dont le générateur et le discriminateur interagissent.

Nous listons ci-dessous quelques modèles populaires, bien que cette liste ne soit pas exhaustive. Il existe une multitude d'autres types de GAN, chacun visant à résoudre des problèmes variés.

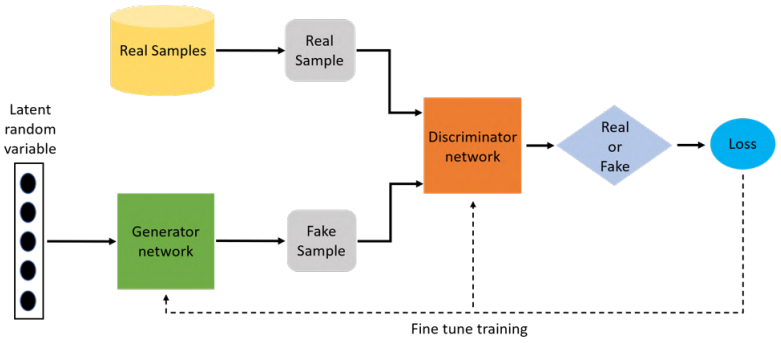

FIGURE 1.1 : Architecture de base d'un GAN Dash *et al.* (2023)

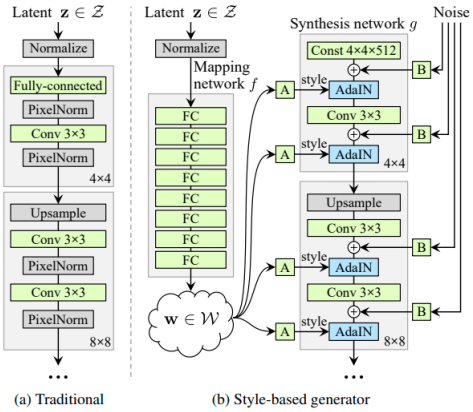

FIGURE 1.2 : Générateur traditionnel vs Générateur StyleGAN Karras *et al.* (2021)

1.2 STYLEGAN

Développé en fin 2018 par des chercheurs de NVIDIA, SyleGAN est un GAN qui génére des images à très haute résolution, même de 1024*1024, rendant les visages générés quasi indiscernables de véritables visages. L'idée est de construire une pile de couches où les couches initiales sont capables de générer des images à faible résolution (à partir de 2*2) et où les couches suivantes augmentent progressivement la résolution. La Figure 1.2 représente l'architecture typique de StyleGAN. Le vecteur de l'espace latent z passe par une transformation de mise en correspondance comprenant 8 couches entièrement connectées, tandis que le réseau de synthèse comprend 18 couches, où chaque couche produit une image

de 4 x 4 à 1024 x 1024. La couche de sortie produit une image RGB par le biais d'une couche de convolution séparée. Cette architecture comporte 26,2 millions de paramètres et, en raison de ce nombre très élevé de paramètres entraînables, ce modèle nécessite un très grand nombre d'images d'entraînement pour construire un modèle performant.

Chaque couche est normalisée à l'aide de la fonction de normalisation d'instance adaptative (*AdaIN*) comme suit :

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}.$$

où chaque carte x_i est normalisée séparément, puis mise à l'échelle et biaisée à l'aide des composantes scalaires correspondantes du style y. La dimensionnalité de y est donc deux fois supérieure au nombre de cartes de cette couche.

1.2.1 STYLEGAN2 ET STYLEGAN3

Nous avons exploré le fonctionnement de la version originale de StyleGAN, mais depuis la fin de 2018, les chercheurs de Nvidia ont continué à améliorer leur programme ! En fait, le 3 décembre 2019, ils ont mis en ligne leur deuxième article sur StyleGAN2, une mise à jour de StyleGAN, où ils ont modifié certaines couches du générateur et amélioré sa régularisation (ce qui a optimisé ses performances). Ensuite, en juin 2021, Nvidia a publié son dernier document concernant StyleGAN, qui introduit StyleGAN3, dont le fonctionnement est similaire à celui de StyleGAN2, mais il se distingue par sa façon de représenter les images, permettant une translation et une rotation plus efficaces Kassel (2022).

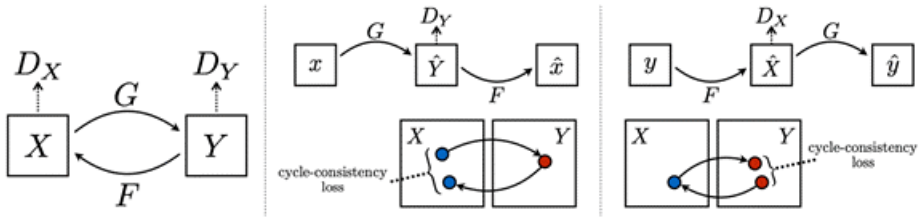

FIGURE 1.3 : Architecture CycleGAN Zhu *et al.* (2017)

1.3 CYCLEGAN

Un CycleGAN est une variante de GAN qui crée la transformation d'une image dans un autre domaine. Le concept consiste à rassembler deux ensembles de données, chaque ensemble représentant un domaine différent. Par exemple, si l'objectif est de former un modèle pour transformer un paysage d'été en paysage d'hiver, deux ensembles de données seront constitués, un avec de nombreux paysages d'été et l'autre avec de nombreux paysages d'hiver, sans lien direct entre les paysages Éric Debeir (2019). Grâce à une architecture spécifique, comme illustré dans la Figure1.3, dans certaines conditions, le modèle apprendra à générer correctement la transformation d'une nouvelle entrée. FaceApp est un des exemples les plus connus de CycleGAN, dans lequel les visages humains sont modifiés pour montrer différents âges. La Figure1.4 montre quelques transformations d'images réalisées par CycleGAN.

1.4 PROGAN

Le ProGAN (Progressive Growing GAN) est une version avancée du GAN qui peut produire des images de haute qualité, comme des visages réalistes mesurant 1024 × 1024 pixels, en optimisant l'entraînement du modèle générateur. Il augmente graduellement le nombre de couches pendant le processus d'apprentissage, ce qui donne lieu au Progressive GAN ou ProGAN.

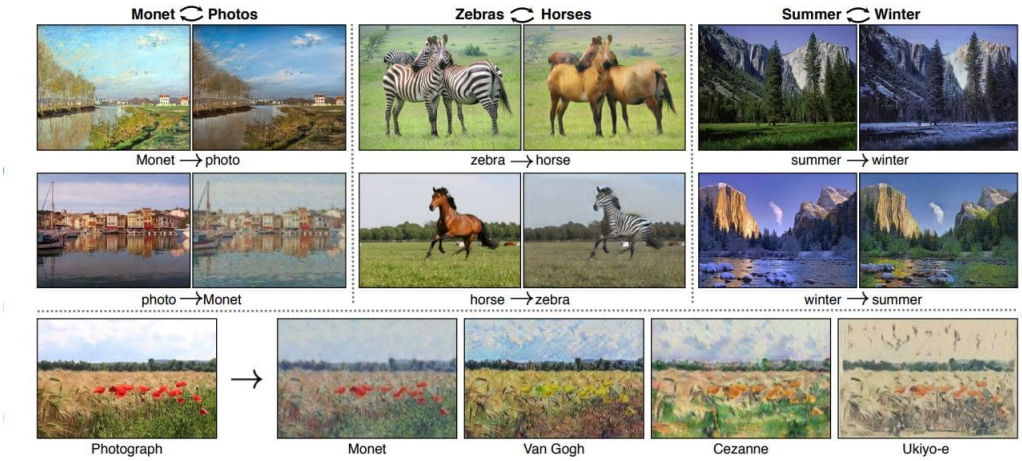

FIGURE 1.4 : Exemple de transformation d'image par CycleGAN Zhu *et al.* (2017)

Le GAN à croissance progressive fonctionne en ajoutant des blocs de couches de manière contrôlée et en ajustant ces blocs nouvellement ajoutés au lieu de les intégrer directement. De plus, durant l'entraînement, toutes les couches, y compris celles déjà présentes, sont maintenues ajustables chaque fois qu'une nouvelle couche est intégrée. Ce type de GAN débute avec des images de très petite taille et utilise à la fois un générateur et un discriminateur qui possèdent la même structure de base, comme 4×4 pixels. Ensuite, de nouveaux blocs convolutifs sont ajoutés pendant l'apprentissage du générateur et du discriminateur, pour créer des images de haute définition de 1024x1024 (voir Figure1.5). Cette expansion progressive des réseaux du générateur (G) et du discriminateur (D) permet une meilleure appréhension des détails complexes dans un premier temps, puis de se concentrer sur la compréhension des caractéristiques fines des images à haute résolution (1024×1024). Cela accroît la stabilité du modèle et réduit la probabilité d'un "effondrement du mode".

Ils utilisent également des techniques de lissage appelées Mini Batch standard deviation et Pixel-wise normalization au lieu de la norme de lot pour la sortie des nouvelles couches afin d'éviter les chocs soudains sur la couche à plus petite résolution déjà bien entraînée lors de l'ajout de nouveaux blocs de couches.

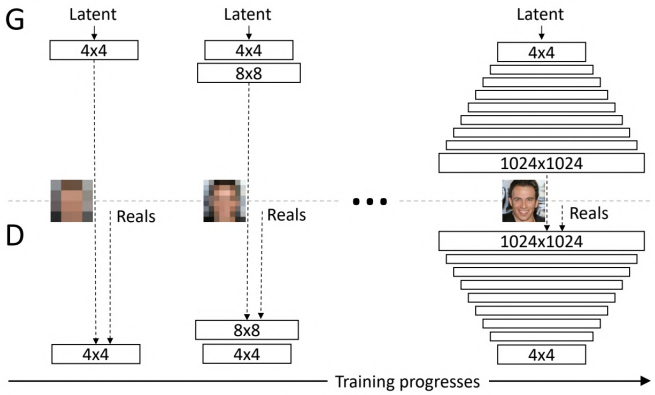

FIGURE 1.5 : Architecture ProGAN Dash *et al.* (2023)

1.5 DCGAN

Afin d'améliorer la qualité de génération des images synthétiques, la technique des GANs a été combinée avec des réseaux de neurones convolutifs (CNN), reconnus pour leur efficacité en vision par ordinateur, notamment dans les tâches de reconnaissance d'images. Cette synergie a conduit à la proposition du Deep Convolutional Generative Adversarial Network (DCGAN) Dash *et al.* (2023). Le DCGAN repose sur une séquence d'opérations convolutionnelles, intégrant notamment des techniques de suréchantillonnage spatial dans le générateur, afin d'améliorer la qualité des images produites.

L'architecture DCGAN a été conçue pour réduire le phénomène d'effondrement de mode, un problème fréquent dans les GANs où le générateur se contente de produire un nombre limité de sorties répétitives, ignorant ainsi la diversité des données réelles. Grâce à sa conception, le DCGAN offre une plus grande stabilité lors de l'apprentissage de la génération d'images et constitue aujourd'hui une base solide pour de nombreuses architectures GAN ultérieures.

Les directives architecturales pour garantir cette stabilité consistent à remplacer les couches de pooling classiques par des convolutions à pas (strided convolutions) dans le

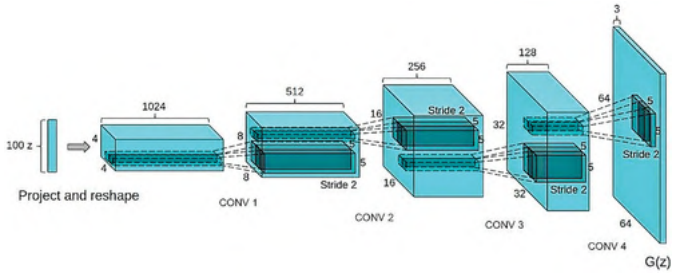

FIGURE 1.6 : Architecture du Générateur de DCGAN Dash *et al.* (2023)

discriminateur, et par des convolutions à pas fractionnés (fractionally-strided convolutions) dans le générateur (voir la Figure 1.6). Ces dernières permettent respectivement de réduire ou d'augmenter la dimension spatiale des données tout en laissant le réseau apprendre directement ces transformations.

1.6 DISCOGAN

DiscoGAN (Discover Cross-Domain GAN) est un modèle basé sur les GANs, conçu pour apprendre automatiquement les relations entre deux domaines d'images non appariées, sans supervision explicite. L'architecture repose sur deux générateurs G_{AB} et G_{BA}, chacun responsable de la traduction dans un sens (A$\rightarrow$B et B$\rightarrow$A), ainsi que de deux discriminateurs D_A et D_B, comme représenté dans la Figure 1.7. Chaque transformation est suivie d'une reconstruction inverse, assurée par le second générateur, et comparée à l'image d'origine à l'aide d'une perte de reconstruction. L'entraînement est guidé à la fois par des pertes adversariales (via les discriminateurs) et des pertes de reconstruction (pour garantir la cohérence des allers-retours). Cette approche permet de découvrir des correspondances bidirectionnelles et d'éviter l'effondrement de mode, assurant une meilleure diversité dans les images générées et un apprentissage stable des relations inter-domaines.

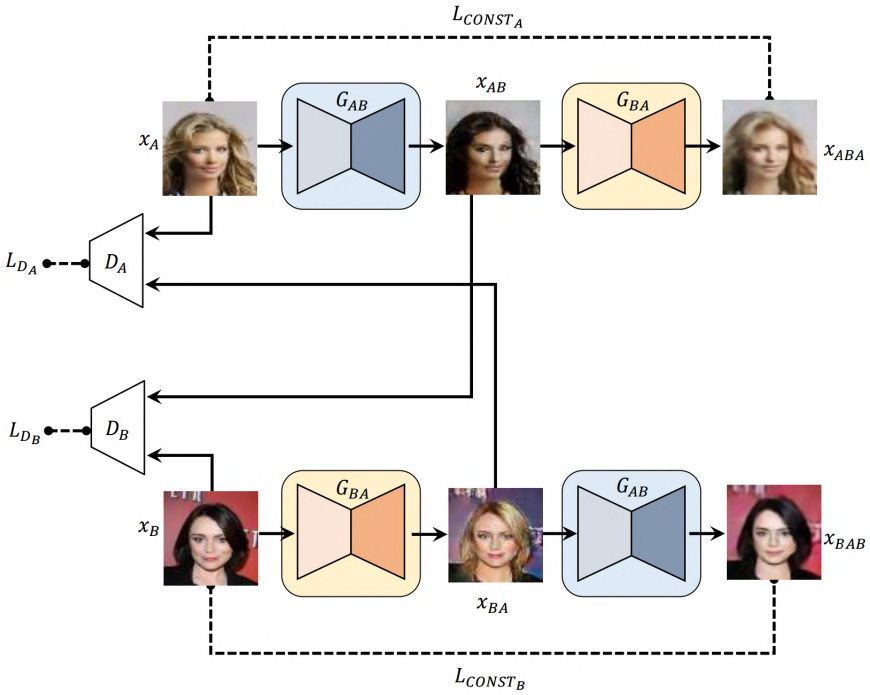

FIGURE 1.7 : Architecture DiscoGAN Kim *et al.* (2017)

1.7 STARGAN

StarGAN introduit une nouvelle approche de la traduction multi-domaine d'image à image, qui permet de transformer des images d'un domaine à l'autre tout en gérant simultanément plusieurs attributs. Voici quelques-uns des principaux composants et méthodes de StarGAN :

1. Modèle unifié : Contrairement aux méthodes précédentes qui nécessitaient des modèles séparés pour chaque domaine ou transformation d'attribut, StarGAN utilise un seul générateur et un seul discriminateur pour tous les domaines, voir la Figure 1.8. Ce modèle unifié simplifie le processus d'apprentissage et réduit la charge de calcul.
2. GAN conditionnel : StarGAN utilise un réseau adversarial génératif conditionnel (cGAN), où le générateur est conditionné à la fois par l'image d'entrée et par le domaine

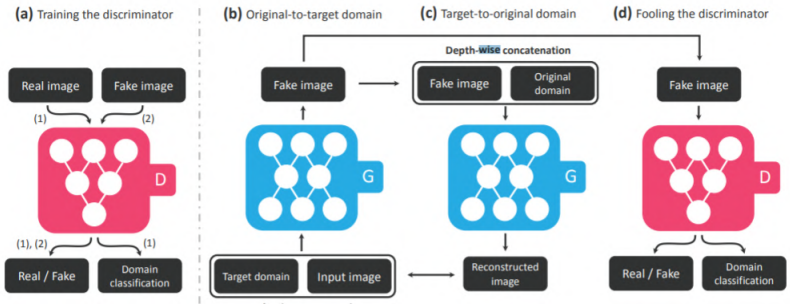

FIGURE 1.8 : Architecture StarGAN Choi *et al.* **(2018)**

cible ou l'étiquette de l'attribut. Cela permet une manipulation contrôlée des attributs pendant la génération de l'image.

3. Perte d'adversité : la perte d'adversité encourage le générateur à créer des images réalistes dans le domaine cible et confond le discriminateur qui classe les images générées comme étant réelles.
4. Perte de cohérence de cycle : Pour maintenir la qualité de l'image et assurer la réversibilité des traductions, StarGAN incorpore une perte de cohérence de cycle, inspirée du CycleGAN. Cette perte impose que la traduction d'une image d'un domaine à un autre, puis le retour, doit produire une image similaire à l'originale.

1.8 CGAN

Un GAN conditionnel est un algorithme de formation d'un réseau GAN, dans lequel le générateur est conditionné par des informations supplémentaires, telles qu'une étiquette ou un vecteur de valeurs de caractéristiques. Ces informations supplémentaires peuvent être utilisées pour améliorer la qualité des échantillons générés, par exemple en permettant au générateur de mieux correspondre à la distribution des données d'apprentissage.

L'un des avantages d'un GAN conditionnel est que le générateur peut être réglé de manière à générer des échantillons plus proches des données d'apprentissage. Cela peut être utile pour des tâches telles que la synthèse d'images, où il est important de produire des images réalistes qui correspondent à la distribution des données d'apprentissage.

Un autre avantage d'un GAN conditionnel est que les informations supplémentaires peuvent être utilisées pour améliorer la qualité des échantillons générés. Par exemple, dans le contexte de la synthèse d'images, les informations supplémentaires pourraient être utilisées pour améliorer le réalisme des images générées en permettant au générateur de mieux correspondre à la distribution des données d'apprentissage.

1.9 CONCLUSION

Dans ce chapitre nous avons passé en revue les différents types de GAN, leurs architectures ainsi que leurs fonctionnalités. Dans le prochain chapitre, nous parlerons des techniques d'apprentissages automatiques ainsi que des Vision Transformers.

CHAPITRE II
L'INTELLIGENCE ARTIFICIELLE

2.1 DÉFINITION DE L'IA

L'intelligence artificielle (IA) est la création et l'utilisation d'algorithmes dans un environnement informatique pour imiter l'intelligence humaine. Elle est utilisée dans de nombreux domaines tels que la conduite automatique, la détection de pathologies médicales ou encore les assistants virtuels. Cette montée croissante de l'IA est due aux développements de ses branches principales, représentées dans la Figure 2.1, qui sont :

— **Le Machine Learning** ou apprentissage automatique à partir de données, permet aux machines d'adapter leur comportement et de faire des prédictions ;

— L'apprentissage profond plus connu sous le nom de **Deep Learning** en anglais, est une branche de l'apprentissage automatique. Le Deep Learning se base sur des réseaux de neurones artificiels profonds afin de résoudre des problèmes complexes et d'apprendre des hiérarchies de caractéristiques.

2.2 MACHINE LEARNING

Le principe du Machine Learning (ML) repose sur la capacité d'une machine à apprendre à partir d'exemples, lui permettant ainsi de réagir à des situations où elle n'a pas été explicitement programmée. Tout d'abord, les machines passent par un processus de formation où les données et leurs solutions sont transmises à la machine. Cela lui permet de déduire des règles pour associer des données à leurs solutions respectives. Pour optimiser la recherche de la meilleure solution, les performances des machines sont souvent mesurées par la fonction d'analyse. Une fois la phase d'apprentissage terminée, la machine est exposée à des données

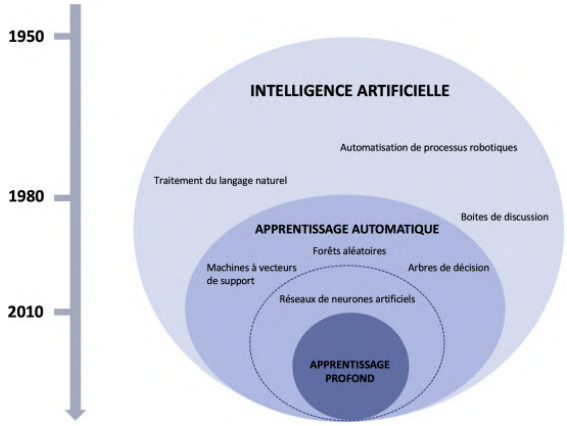

FIGURE 2.1 : Intelligence artificielle et ses branches Bunod *et al.* (2022)

inconnues. Grâce aux règles définies lors de l'apprentissage, il peut faire des prédictions sur ces nouvelles données TOURE (2021).

Pour faire apprendre à un ordinateur on a plusieurs méthodes d'apprentissage automatique.

2.2.1 ML SUPERVISÉ

Le principe de cette méthode est d'assister l'ordinateur dans son apprentissage. En effet, on lui donne les Entrées/Sorties de données étiquetées qu'il doit apprendre.

Cette méthode est la plus utilisée et la plus efficace pour la détection d'objet dans une image. Un exemple très illustratifs est la recherche de chat dans une image. En effet, les données d'entrées ce sont les images et les données de sorties c'est la présence ou non d'un chat.

Il existe un certain nombre de modèles d'apprentissage supervisé que l'on peut implémenter sous forme d'algorithmes, à la fois mathématiques et informatiques. Ces modèles

diffèrent dans leur approche de la formation des données ainsi que dans le type d'étiquettes à prédire, qu'il s'agisse d'une valeur continue ou d'une valeur de classe.

L'un des modèles les plus populaires en apprentissage supervisé pour prédire des valeurs continues est la régression linéaire. Par exemple, ce modèle peut être utilisé pour prédire le prix d'une maison en fonction de sa taille, du nombre de chambres et de son emplacement.

Bien que la régression linéaire soit très efficace pour capter les relations linéaires entre variables explicatives et à expliquer, notamment grâce à ses variations (comme la version régularisée pour éviter le sur-apprentissage), elle montre sa limite lorsque la relation entre variables est plus complexe qu'une simple linéarité.

Dans d'autres tâches supervisées telles que la classification, plusieurs modèles sont disponibles, tels que des modèles basés sur des arbres de décision (tels que RandomForest), des variantes de la régression telles que la régression logistique et des machines à vecteurs de support (SVM) TOURE (2021).

L'apprentissage supervisé va au-delà de ces algorithmes, même s'ils représentent l'état de l'art de l'apprentissage automatique classique.

L'apprentissage profond, basé sur l'utilisation de réseaux de neurones profonds, est également largement utilisé pour effectuer un apprentissage supervisé dans le cadre de problèmes complexes tels que la classification de données non structurées (images, sons, vidéo). Il est même utilisé pour obtenir de meilleurs résultats dans des problèmes d'apprentissage automatique classiques.

2.2.2 ML NON SUPERVISÉ

Comme son nom l'indique, avec ce type d'apprentissage on ne connaît pas les données de sorties. Donc, on présente à l'ordinateur les données d'entrées et c'est à lui d'apprendre tout seul sans assistance.

Son objectif principal est de cataloguer automatiquement les données soumises pour identifier différentes entités telles que les chats, les voitures, les animaux et les personnes. L'un des problèmes d'apprentissage non supervisé les plus courants est la segmentation, où une personne essaie de regrouper des données en différents groupes, tels que des catégories, des classes ou des clusters. Par exemple, nous pouvons grouper des images de voitures ou de chats. De nombreuses perspectives prometteuses se concentrent sur la détection d'anomalies, particulièrement utilisée pour la maintenance prédictive, la cybersécurité et la détection précoce des maladies TOURE (2021).

En général, le but de l'algorithme est d'atteindre deux résultats : premièrement, il vise à rendre les données homogènes dans des groupes de données, et deuxièmement, il cherche à former des groupes aussi distincts que possible. Selon le contexte, différents algorithmes sont choisis pour classer les données, en fonction de critères tels que la densité ou le gradient de densité des données. Dans le cas de la détection d'anomalies, l'accent est plutôt mis sur l'identification de valeurs ou de modèles extrêmes ou atypiques dans les données. La métrique de base joue un rôle important dans la détermination de ce qui est considéré comme la norme et de ce qui en diffère.
Cette méthode est utilisée lorsque les données de sortie sont incontrôlées ou inconnues.

2.2.3 ML PAR RENFORCEMENT

L'apprentissage par renforcement fait référence à une classe de problèmes d'apprentissage automatique qui visent à apprendre à partir d'expériences successives. Dans ce type de problème, la machine interagit avec son environnement pour trouver la solution optimale. Contrairement aux problèmes supervisés et non supervisés, l'apprentissage par renforcement se distingue par ses propriétés interactives et itératives TOURE (2021). La machine explore différentes solutions, observe les réactions de l'environnement et ajuste son comportement pour trouver la meilleure stratégie.

L'apprentissage par renforcement est appliqué dans de nombreux domaines tels que la robotique, la gestion des ressources, le vol en hélicoptère et la chimie. Cette méthode a été couronnée de succès dans divers problèmes tels que le contrôle des robots, le pendule inversé, la planification des tâches et les télécommunications TOURE (2021).

2.3 DEEP LEARNING

L'apprentissage profond est une approche innovante qui met l'accent sur l'apprentissage par couches successives à l'aide d'un réseau de neurones artificiels. Le terme *profond* fait référence à la superposition de ces classes de représentation Chollet & Chollet (2021). La profondeur d'un modèle est déterminée par le nombre de ces couches.

Un système d'apprentissage profond parvient à reconnaître une personne en faisant correspondre hiérarchiquement les contours et les caractéristiques du corps de la personne PRADEEP *et al.* (2018).

Les champs d'application du Deep Learning sont très vastes :

- Classification des images ;

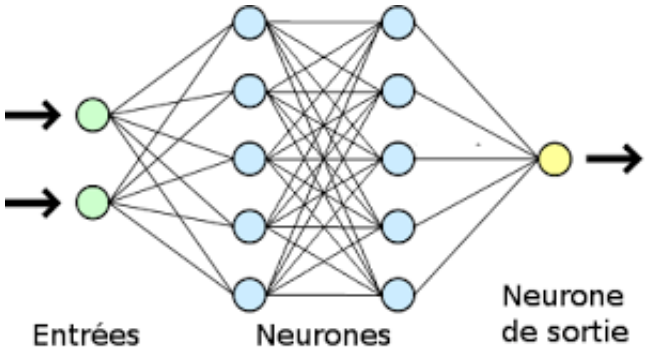

FIGURE 2.2 : Réseaux de neurones Guandamatoko (2019)

- Reconnaissance vocal ;
- Traduction automatique améliorée ;
- Conduite automatique, etc.

2.3.1 LES RÉSEAUX DE NEURONES

Les algorithmes d'apprentissage profond sont basés sur des réseaux de neurones artificiels, qui sont des modèles de traitement de l'information conçus pour imiter le comportement de notre système nerveux biologique. Au cœur de ces réseaux se trouvent les neurones, qui représentent les unités de base les plus simples.

Un réseau de neurones artificiels est constitué d'une série de neurones organisés en couches (voir Figure 2.2). Les couches d'entrée et de sortie sont les couches visibles du réseau, tandis que les couches intermédiaires sont considérées comme les couches cachées. Les neurones sont généralement interconnectés, ce qui signifie que chaque neurone d'une couche est connecté à tous les neurones de la couche suivante.

2.3.2 FONCTIONNEMENT D'UN RÉSEAU DE NEURONES

Pendant la phase d'apprentissage, le réseau de neurones est exposé à des données pour lesquelles une solution est connue. Chaque nœud d'entrée a un poids qui lui est associé, et une somme pondérée peut être calculée en multipliant le poids par la valeur du nœud d'entrée. Le résultat de cette somme est envoyé à une fonction d'activation qui détermine la valeur de sortie du neurone. La sortie de ce neurone est ensuite envoyée à un autre neurone. Le dernier niveau capture les interactions les plus complexes.

Les fonctions d'activation jouent un rôle important dans les réseaux de neurones. Cela introduit de la non-linéarité dans le modèle, nous permettant d'étendre l'espace des hypothèses. Sans cette fonction d'activation, le modèle est limité à l'apprentissage de transformations linéaires, ce qui limite sévèrement l'espace des hypothèses.

Les fonctions d'activation les plus courantes sont :

— La fonction ReLU (Rectified Linear Activation)

— La fonction sigmoïde

— La fonction Softmax

LA FONCTION RELU

La fonction d'activation ReLU Bai (2022) est largement utilisée dans les réseaux de neurones car elle laisse les valeurs positives inchangées et met les valeurs négatives à zéro pour les supprimer. Cela évite la sortie négative du neurone.

LA FONCTION SIGMOÏDE

La fonction sigmoïde Nantomah (2019) renvoie une valeur comprise entre 0 et 1, ce qui en fait la fonction d'activation la plus utilisée dans la dernière couche des réseaux de neurones. L'objectif est d'obtenir un score pouvant être interprété comme une probabilité à la sortie du réseau. Cette fonction est largement utilisée et est souvent appelée fonction sigmoïde logistique ou logistique.

LA FONCTION SOFTMAX

La fonction Softmax Raghuram *et al.* (2022) est une généralisation de la régression logistique et est utilisée pour classer plusieurs classes. Contrairement à d'autres types de fonctions, la sortie des neurones d'une couche utilisant la fonction Softamx dépend de la sortie de tous les neurones. En effet, la somme de toutes les sorties doivent être égale à 1.

Une fois toute les couches traversées, une prédiction est faite pour cette donnée. dans le cas d'une régression on a une seule valeur alors que dans la classification à plusieurs classes on a un score d'appartenance à chaque classe.

Une fonction de perte est ensuite utilisée pour confronter la valeur prédite à la vraie valeur de la donnée. Cette fonction calcule la différence entre les valeurs réelles et prédites. L'optimiseur utilise ensuite la valeur de perte pour ajuster les pondérations afin de réduire l'écart entre les valeurs prévues et réelles. Par conséquent, les pondérations du réseau jouent un rôle important dans la sauvegarde de la formation du modèle. Ce processus est appelé *Backpropagation* Chollet & Chollet (2021).

Les données sont ensuite présentées à nouveau au réseau neuronal et les étapes ci-dessus sont répétées un nombre défini de fois. Si la précision obtenue après la formation n'est pas satisfaisante, cela peut indiquer que l'architecture du modèle doit être réévaluée. Cela inclut l'ajout ou la suppression de couches et la modification des paramètres.

LA FONCTION DE PERTE ET L'OPTIMISEUR

L'optimiseur et la fonction de perte jouent un rôle important dans les réseaux de neurones. Les fonctions de perte sont utilisées pour évaluer les performances du réseau en fournissant une valeur de perte qui doit être minimisée. Le choix de la fonction de perte est très important dans les modèles d'apprentissage profond car il mesure le succès de la classification. Le choix de cette fonction dépend du problème à traiter, qu'il s'agisse d'une classification binaire, multi−classes ou autre.

Quelques fonctions de perte :

— $Binary - crossentropy$: pour la classification binaire ;

— $Categorical - crossentropy$: pour une multi−classes ;

— Mean Squared Error (mse) : pour la régression vers des valeurs arbitraires.

L'optimiseur joue également un rôle important dans les performances du modèle car il facilite les mises à jour de poids visant à minimiser la fonction de perte. Il est basé sur un algorithme de gradient stochastique qui utilise des dérivées pour réduire la fonction objective.

LES RÉSEAUX ENTIÈREMENT CONNECTÉS

Ce réseau spécial est utilisé pour transformer une liste d'entrées en une liste de sorties. Dans une couche entièrement connectée, les neurones sont connectés à toutes les sorties de

la couche précédente. Par conséquent, leurs fonctions d'activation peuvent être calculées en effectuant une multiplication matricielle suivie d'un décalage de polarisation.

2.4 ALGORITHMES DE DEEP LEARNING

2.4.1 LE PERCEPTRON

Inventé en 1957 par Frank Rosenblatt au Cornell Aeronautical Laboratory, le perceptron est basé sur le concept original d'un neurone artificiel. Il agit comme une unité de réseau neuronal et effectue des calculs pour identifier les modèles et les tendances dans les données fournies. Cet algorithme d'apprentissage supervisé permet une classification binaire. Grâce à cet algorithme, des neurones artificiels peuvent apprendre et traiter des éléments d'un ensemble de données Guandamatoko (2019).

Un perceptron, tel que représenté dans la Figure 2.3, contient un nombre variable d'entrées (X_1 à X_n) (généralement des entiers ou des réels) transmises à l'unité. À chaque entrée, un poids W est lié à X, et cela sera utilisé pour déterminer la sortie Y. De plus, une connexion de biais est incluse, car elle est essentielle pour que le perceptron fonctionne correctement Guandamatoko (2019). Le calcul effectué pour obtenir la sortie Y est le suivant :

$$Y = \sum W_n X_n \neq \theta \tag{2.1}$$

LE PERCEPTRON MULTICOUCHE

Les perceptrons simples ont rapidement atteint leurs limites techniques. En fait, un perceptron monocouche ne peut séparer les classes que si les classes sont linéairement séparables.

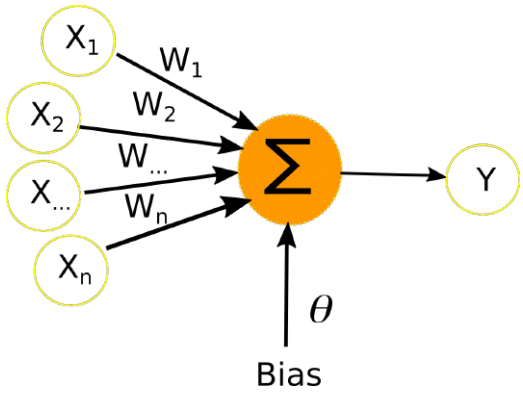

FIGURE 2.3 : Perceptron Guandamatoko (2019)

Pour surmonter ces limitations, des perceptrons multicouches ont été découverts pour pouvoir classer des groupes linéairement inséparables.

Les Perceptrons Multicouches, également connus sous le nom de MLP (Multilayer Perceptron), sont des réseaux à propagation. Il se compose de plusieurs couches, chacune avec un nombre variable de neurones (voir Figure 2.4). La dernière couche de neurones est la sortie du système global Turpin (2023).

Ce modèle peut être utilisé pour toutes les tâches de classification supervisées. Aujourd'hui, il est très populaire et de nombreuses bibliothèques telles que TensorFlow, Weka et Scikit-Learn l'implémentent. Les neurones des perceptrons multicouches peuvent être considérés comme une série de perceptrons interconnectés. Une caractéristique topologique de ce réseau est que chaque neurone d'une couche est connecté à chaque neurone de la couche suivante. Ainsi, chaque neurone a n entrées, où n représente le nombre de neurones dans la couche précédente, et la sortie est envoyée à tous les neurones de la couche suivante.

Chaque connexion neuronale est associée à un poids W, similaire aux entrées du perceptron. Le calcul de la sortie du neurone est déterminé par la fonction d'activation sélectionnée.

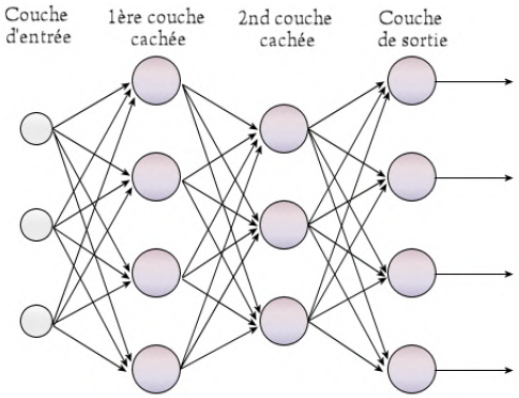

FIGURE 2.4 : Perceptron multicouche Guandamatoko (2019)

Voir la section Fonctionnement d'un réseau de neurones pour plus de détails sur les fonctions d'activation.

2.4.2 LES RÉSEAUX DE NEURONES DITS CONVOLUTIFS (CNN)

Un CNN (Convolutional Neural Network), également connu sous le nom de réseau de neurones convolutifs, suppose une configuration spatiale particulière en entrée. En particulier, les entrées adjacentes dans l'entrée d'origine sont considérées comme sémantiquement liées. Un CNN se compose de différentes couches, chaque couche transformant un volume d'activation en un autre à l'aide d'une fonction différentiable. Il existe trois principaux types de couches utilisées pour construire ce type de réseau : la couche de convolution, la couche de Pooling et la couche entièrement connectée ou *Fully Connected*.

FONCTIONNEMENT D'UNE CONVOLUTION

Dans le processus de convolution, l'image est divisée en sections plus petites appelées tuiles et analysée à l'aide de noyaux de convolution.

Ce noyau est généralement de la même taille que la tuile, souvent 3x3 ou 5x5. La zone de balayage, appelée champ récepteur, est légèrement plus grande que le noyau en raison du pas ajouté, ce qui permet aux champs récepteurs de se chevaucher. Cette technologie améliore la représentation de l'image et augmente la cohérence du traitement.

L'analyse des propriétés de l'image est effectuée par un noyau de convolution qui effectue un filtrage en attribuant un poids à chaque pixel. L'application d'un filtre à une image s'appelle la convolution.

La convolution donne une carte de caractéristiques qui représente l'image de manière abstraite. Les valeurs de cette carte dépendent des paramètres du noyau de convolution utilisé et des valeurs de pixels de l'image d'origine Madeleine (2021).

LA COUCHE DE CONVOLUTION

La couche principale du CNN est constituée d'un empilement de convolutions. En pratique, une image est analysée par plusieurs noyaux de convolution pour produire plusieurs cartes de caractéristiques de sortie. Chaque noyau de convolution a des paramètres spécifiques pour détecter des informations spécifiques dans l'image. Par exemple, un noyau de convolution de type filtre Sobel utilise des paramètres spécifiques pour identifier les contours d'une image.

Les paramètres du noyau de convolution sont choisis en fonction de la tâche à effectuer. Celles-ci sont automatiquement apprises par l'algorithme à partir des données d'apprentissage à l'aide de techniques d'apprentissage automatique. La rétropropagation du gradient joue un rôle important dans l'ajustement des paramètres en fonction du gradient de la fonction de perte. Une fonction qui quantifie l'erreur entre la valeur prédite et la valeur cible Madeleine (2021).

LA COUCHE DE POOLING OU DE MISE EN COMMUN

Pour réduire le nombre de paramètres et de calculs dans le réseau, les couches de pooling sont généralement insérées entre deux couches convolutives consécutives dans les réseaux de neurones convolutifs. Cela permet de lutter contre le surapprentissage ou *overfitting*.

La méthode de pooling la plus couramment utilisée est MaxPool(2x2, 2). Il est plus efficace que la moyenne car il privilégie les activations les plus fortes. Cette opération est appliquée à la sortie de la couche précédente comme un filtre de convolution de taille (2x2) et se déroule en deux étapes. Cela comprime la feature map obtenue en sortie de la couche de pooling d'un facteur 4 Madeleine (2021).

FULLY CONNECTED OU COUCHE ENTIÈREMENT CONNECTÉE

Située en bout de réseau, cette couche est chargée de classer les images en fonction des caractéristiques extraites par blocs de traitement successifs. Il est entièrement connecté, ce qui signifie que toutes les entrées de cette couche sont connectées aux neurones de sortie. Par conséquent, ces neurones ont accès à toutes les informations d'entrée. Chaque neurone attribue à une image une valeur de probabilité appartenant à l'une des classes possibles. Contrairement à la phase d'extraction de caractéristiques, les neurones de traitement sont indépendants les uns des autres et ne peuvent accéder qu'aux informations du champ reçu qu'ils traitent. Madeleine (2021).

Ces réseaux de neurones à convolution sont utilisés principalement dans le traitement d'images et de vidéos, où ils excellent Madeleine (2021).

2.4.3 LES RÉSEAUX DE NEURONES DITS RÉCURRENTS (RNN)

En anglais Recurrent Neural Network (RNN) et en français réseaux de neurones récurrents, permettent la propagation bidirectionnelle de l'information, ce qui inclut la transmission des couches profondes aux couches plus proches de l'entrée. Cette caractéristique les rapproche du fonctionnement réel du système nerveux, qui n'est pas unidirectionnel.

Ces réseaux sont dotés de connexions récurrentes, ce qui signifie qu'ils peuvent conserver des informations en mémoire et prendre en compte des états passés à un instant donné. C'est pourquoi les RNN sont particulièrement adaptés aux applications qui nécessitent de prendre en compte le contexte, notamment le traitement de séquences temporelles telles que l'apprentissage et la génération de signaux. Dans ce cas, les données forment une suite et ne sont pas indépendantes les unes des autres. Cependant, pour les applications impliquant de longs écarts de temps, comme la classification de séquences vidéo, cette "mémoire à court-terme" des RNN n'est pas suffisante Wikipédia (2021).

En effet, les RNN "classiques" (réseaux de neurones récurrents simples ou Vanilla RNNs) sont limités à mémoriser uniquement le passé récent et commencent à "oublier" après environ cinquante itérations. La nature bidirectionnelle de la transmission de l'information complique considérablement leur entraînement, et ce n'est que récemment que des méthodes efficaces, comme les LSTM (Long Short Term Memory), ont été développées.

Ces réseaux dotés d'une "mémoire court-terme" étendue ont révolutionné des domaines tels que la reconnaissance vocale par les machines (Speech Recognition) ou la compréhension et la génération de texte (Natural Language Processing). D'un point de vue théorique, les RNN offrent un potentiel bien plus vaste que les réseaux de neurones classiques : des recherches ont démontré qu'ils sont "Turing-complets", c'est-à-dire qu'ils peuvent théoriquement simuler

n'importe quel algorithme. Cependant, cela ne fournit aucune indication sur la façon de les construire concrètement Wikipédia (2021).

2.4.4 LES TRANSFORMERS

Les Transformers ont été introduits par Vaswani *et al.* (2017), ce modèle repose sur une architecture de type encodeur-décodeur, mais se distingue par l'intégration de mécanismes d'attention novateurs. Initialement, l'architecture propose un empilement symétrique d'encodeurs et de décodeurs, bien que ce ne soit plus systématiquement le cas aujourd'hui. Chaque encodeur se compose d'un bloc d'auto-attention (self-attention) suivi d'une couche linéaire, tandis que le décodeur intègre également un mécanisme d'attention supplémentaire. L'un des apports majeurs des transformers réside dans l'introduction de l'attention multi-têtes (multi-head attention). Bien qu'ils offrent d'excellentes performances, ces modèles restent très gourmands en ressources. En effet, leur architecture riche en paramètres engendre des temps d'apprentissage prolongés ainsi qu'un besoin important en données pour assurer une convergence efficace Vaswani *et al.* (2017).

MÉCANISME D'ATTENTION

Le mécanisme d'attention s'inspire du concept d'attention cognitive, permettant de focaliser le traitement sur les éléments les plus pertinents d'une entrée tout en minimisant l'importance des informations moins significatives. Plus précisément, l'attention établit une correspondance entre un vecteur de requête et un ensemble de paires clé-valeur afin de générer un vecteur de sortie pondéré.

L'une des variantes les plus utilisées est l'attention par produit scalaire mis à l'échelle (Scaled Dot-Product Attention), introduite par Vaswani *et al.* (2017). Cette méthode est

formellement définie par l'équation suivante :

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

où Q, K et V sont respectivement les matrices des requêtes (queries), des clés (keys) et des valeurs (values). Le terme d_k correspond à la dimension des vecteurs de requêtes et de clés. Le facteur de mise à l'échelle $\frac{1}{\sqrt{d_k}}$ est appliqué pour éviter que le produit scalaire QK^T ne prenne des valeurs trop élevées lorsque d_k est important, ce qui pourrait entraîner une saturation de la fonction softmax et compromettre l'efficacité de la rétropropagation des gradients.

ATTENTION MULTI-TÊTE

L'attention multi-tête (Multi-Head Attention) est une extension du mécanisme d'attention introduite dans le modèle Transformer par Vaswani *et al.* (2017). Son principe repose sur l'application parallèle de plusieurs mécanismes d'attention, appelés « têtes », permettant au modèle de capturer simultanément différentes relations ou caractéristiques présentes dans les données d'entrée. Ce procédé est particulièrement pertinent dans des tâches comme le traitement du langage naturel, où il est nécessaire d'appréhender diverses dépendances entre les mots d'une phrase.

Concrètement, avant d'appliquer le mécanisme d'attention, les vecteurs de requêtes (Q), de clés (K) et de valeurs (V) sont projetés linéairement à l'aide de matrices de poids spécifiques, transformant les entrées en vecteurs de dimensions d_k, d_k et d_v, respectivement. Contrairement à l'attention simple qui effectue cette opération une seule fois, l'attention multi-tête la réalise h fois, avec des projections distinctes pour chaque tête. Chaque projection

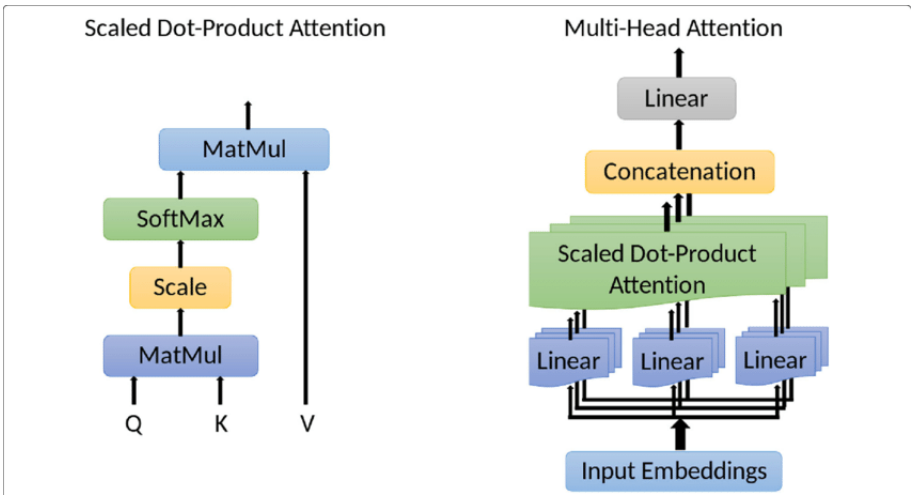

FIGURE 2.5 : Schéma représentant le scaled dot product attention et le multi-head self-attention.

est ensuite traitée par le mécanisme d'attention, puis les sorties des différentes têtes sont concaténées et à nouveau projetées linéairement.

Le mécanisme peut être formalisé comme suit :

$$\begin{aligned} Multi\text{-}Head(Q,K,V) &= Concat(head_1,\ldots,head_h)W^O \\ \text{avec} \quad head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.3}$$

où $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ sont les matrices de projection pour chaque tête, et $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ est la matrice utilisée pour la projection finale après concaténation.

Enfin, lorsque les matrices de clés et de valeurs sont identiques ($K = V$), on parle d'attention « self-attention ». Par extension, l'attention multi-tête appliquée dans ce contexte est appelée **Multi-Head Self-Attention (MHSA)**. On peut le voir en détail dans la Figure 2.5.

ENCODAGE POSITIONNEL

L'architecture Transformer traite l'ensemble des éléments d'une séquence en parallèle, sans tenir compte explicitement de leur ordre. Cette absence d'information sur la position des éléments peut entraîner une perte de la structure séquentielle. Pour pallier cette limitation, un encodage positionnel est ajouté aux représentations d'entrée, permettant au modèle d'intégrer des informations sur l'ordre des éléments.

Plusieurs méthodes existent pour implémenter cet encodage. L'approche la plus courante consiste à utiliser des fonctions sinusoïdales de différentes fréquences, comme proposé par Vaswani *et al.* (2017). Cet encodage positionnel est défini de manière déterministe selon les équations suivantes :

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{aligned} \tag{2.4}$$

où pos représente la position de l'élément dans la séquence, i l'indice de la dimension courante, et d_{model} la dimension du vecteur d'entrée. Ce schéma permet au modèle de capturer efficacement les relations de position relatives entre les éléments, tout en conservant une structure facilement intégrable aux embeddings d'entrée.

BLOC MLP ET BLOC TRANSFORMER GLOBAL

Après le calcul du mécanisme de self-attention sur les entrées, le résultat est transmis à un bloc Multi-Layer Perceptron (MLP), composé de couches entièrement connectées. Ce bloc joue un rôle crucial en ajoutant de la non-linéarité et en enrichissant la capacité de représentation du modèle. Formellement, pour une entrée x, le bloc MLP peut être défini

comme suit :

$$MLP(x) = ((xW_1 + b_1)\, Activation)W_2 + b_2 \tag{2.5}$$

où W_1, b_1, W_2 et b_2 sont des poids et biais appris, et la fonction d'activation utilisée est typiquement l'unité linéaire rectifiée (ReLU). En parallèle, des connexions résiduelles sont ajoutées après les blocs Multi-Head Self-Attention (MHSA) et MLP, suivies d'une normalisation de couche (Layer Normalization, LN), comme suggéré par Ba *et al.* (2016). Cela favorise une meilleure stabilité du réseau et facilite l'apprentissage. Ainsi, en considérant une entrée x_l au niveau de la couche l du bloc Transformer, le fonctionnement du bloc Transformer global peut être formellement exprimé par :

$$\begin{aligned} x'_l &= LN(x_l + MHSA(x_l)) \\ x_{l+1} &= LN(x'_l + MLP(x'_l)) \end{aligned} \tag{2.6}$$

Cette structure permet au modèle de capturer efficacement les dépendances locales (via MHSA) et globales (via MLP), tout en maintenant une stabilité lors de l'entraînement grâce aux normalisations et aux connexions résiduelles (voir l'architecture dans la Figure 2.6).

2.4.5 TYPES DE TRANSFORMERS COURAMMENT UTILISÉS

Il existe principalement deux types de modèles Transformers largement utilisés, chacun étant adapté à un domaine spécifique.

TRANSFORMERS POUR LE TRAITEMENT DU LANGAGE NATUREL (NLP)

Les Transformers ont largement contribué à l'essor de plusieurs modèles performants dans le domaine du NLP. Parmi les architectures les plus emblématiques, on retrouve le **GPT**

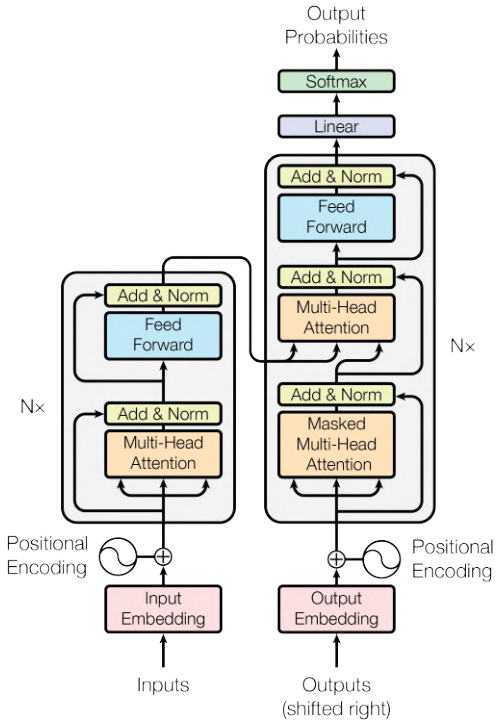

FIGURE 2.6 : Architecture global du transformer Vaswani *et al.* (2017).

(Generative Pre-trained Transformer) et le **BERT (Bidirectional Encoder Representations from Transformers)**.

Generative Pre-trained Transformer (GPT) Proposé par Radford *et al.* (2018), GPT constitue une avancée majeure dans le domaine du NLP. Ce modèle repose exclusivement sur des blocs décodeurs de type Transformer, sans inclure d'encodeur. Contrairement aux décodeurs standards, GPT omet le module d'attention croisée, puisqu'il ne dépend pas d'une représentation issue d'un encodeur. Sa structure se compose donc du codage positionnel, d'une attention auto-dirigée multi-têtes masquée, d'un réseau feedforward, et de mécanismes de normalisation de couche et de résidus.

GPT est principalement entraîné de manière non supervisée sur de vastes ensembles de données textuelles issues d'Internet. Ce pré-entraînement permet au modèle d'être performant

sur diverses tâches de NLP telles que la génération de texte, la traduction ou encore le résumé automatique Ghojogh & Ghodsi (2020).

Bidirectional Encoder Representations from Transformers (BERT) Introduit par Devlin *et al.* (2019), BERT repose uniquement sur des blocs encodeurs de Transformer, conçus pour le pré-entraînement de modèles sur des tâches de NLP telles que l'analyse de sentiment, les systèmes de question-réponse ou encore le résumé de texte.

BERT adopte une approche en deux étapes : un pré-entraînement sur de larges corpus (BooksCorpus et Wikipedia), puis un ajustement fin (fine-tuning) sur des tâches spécifiques. Son pré-entraînement repose sur deux stratégies : le *Masked Language Modeling* (MLM), où certains mots sont masqués et prédits à partir du contexte, et la *Next Sentence Prediction* (NSP), où le modèle apprend à prédire si une phrase suit logiquement une autre. Deux versions principales existent : BERT-base et BERT-large Acheampong *et al.* (2021). Le fine-tuning est rendu efficace en n'ajustant que la couche de sortie spécifique à la tâche, tandis que les paramètres du modèle pré-entraîné sont conservés.

TRANSFORMERS POUR LA VISION PAR ORDINATEUR (CV)

Fort de leur succès dans le NLP, les Transformers ont été adaptés au domaine de la vision par ordinateur, donnant naissance à des architectures performantes telles que l'**Image GPT (iGPT)** et le **Vision Transformer (ViT)**.

Image GPT (iGPT) Proposé par Chen *et al.* (2020), iGPT étend l'approche de GPT à la génération d'images. Le modèle est entraîné sur ImageNet en traitant les pixels d'une image sous forme séquentielle, à l'image du traitement des tokens textuels. Toutefois, iGPT reste

limité en raison de sa forte exigence en puissance de calcul et de la qualité inférieure des images générées Berroukham *et al.* (2023).

Vision Transformer (ViT) Introduit par Dosovitskiy *et al.* (2020), le Vision Transformer est conçu pour des tâches de classification d'images. Contrairement aux réseaux convolutifs classiques, ViT divise une image en une séquence de patchs, chacun étant traité comme un token. Cette approche permet au modèle de capturer efficacement les relations globales et les dépendances à longue portée entre différentes régions de l'image. ViT a démontré des performances compétitives, voire supérieures, sur plusieurs benchmarks en vision par ordinateur.

2.4.6 VISION TRANSFORMERS (VIT)

Inspirée par le succès des Transformers dans le domaine du traitement du langage naturel (NLP), plusieurs travaux ont cherché à transposer cette architecture aux tâches de vision par ordinateur. La première tentative aboutie dans ce domaine a été réalisée par l'équipe de Google Brain, qui a proposé le *Vision Transformer (ViT)* Dosovitskiy *et al.* (2020). À l'instar de l'encodage de séquences de mots dans les modèles NLP, ViT divise une image en une séquence de patchs, chacun étant traité comme un token. L'architecture repose uniquement sur des blocs encodeurs, sans recourir aux couches convolutives traditionnelles, et vise à prédire les étiquettes de classe associées à l'image.

L'ARCHITECTURE VIT

L'architecture du *Vision Transformer (ViT)* représentée dans la Figure 2.7, est conçue pour traiter les images sous forme de séquences, à l'image du traitement des tokens en NLP.

ViT segmente les images en patchs non chevauchants, chacun étant ensuite traité comme un token.

L'un des principaux avantages de l'attention dans les tâches de vision est sa capacité à capturer les dépendances globales, en évaluant l'importance relative des différentes régions d'une image. Contrairement aux réseaux convolutifs qui opèrent localement, le mécanisme d'attention permet une analyse globale du contenu visuel.

Contrairement aux modèles Transformers d'origine basés sur une architecture encodeur-décodeur, ViT adopte uniquement une structure d'encodeur. Ce dernier est constitué de l'intégration des patchs suivie d'une succession de blocs composés de **Multi-Head Self-Attention (MHSA)** et de **Multi-Layer Perceptron (MLP)**. Un **MLP Head** est ajouté à la fin de l'architecture pour effectuer la classification. Les différentes composantes du ViT sont les suivantes :

— **Multi-Head Self-Attention (MHSA)** : Cette couche applique plusieurs mécanismes d'attention en parallèle. Les sorties issues de chaque tête sont concaténées linéairement, facilitant l'apprentissage conjoint des dépendances locales et globales au sein de l'image.

— **Multi-Layer Perceptron (MLP)** : Chaque bloc MLP est composé de deux couches entièrement connectées, séparées par une fonction d'activation *GELU* (Gaussian Error Linear Unit).

— **MLP Head** : Il s'agit d'un MLP appliqué à un jeton spécial appelé *jeton [classe]* ([class] token), extrait à la sortie du dernier bloc d'attention. Ce jeton est utilisé pour produire la prédiction finale de la classe de l'image.

Tokenisation d'entrée et encodage positionnel Traiter chaque pixel individuellement comme un token serait prohibitif en raison de la complexité quadratique $O(n^2)$ du mécanisme

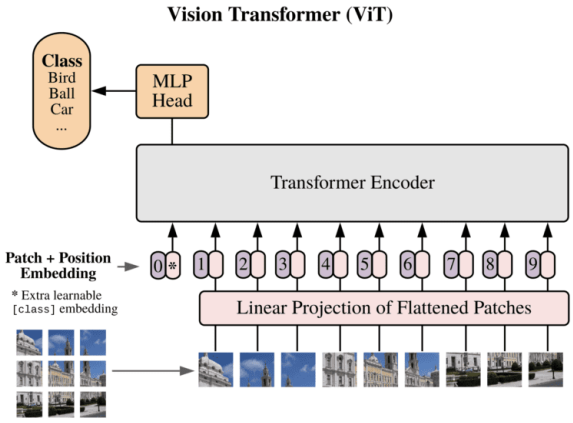

FIGURE 2.7 : Architecture du Vision Transformer (ViT) Dosovitskiy *et al.* (2020)

d'attention, aussi bien en mémoire qu'en temps d'exécution. Pour contourner cette limitation, Dosovitskiy *et al.* (2020) proposent de diviser l'image en patchs non chevauchants. Chaque patch est aplati et projeté linéairement dans un espace de dimension d_{model} via une couche convolutionnelle avec un pas égal à la taille du patch, ce qui permet d'obtenir une séquence de tokens d'entrée réduite. De plus, contrairement au Transformer original proposé par Vaswani *et al.* (2017), ViT utilise des encodages positionnels **apprenables** plutôt que des codages fixes et prédéfinis. Ces encodages sont ajoutés aux représentations des patchs pour conserver l'information spatiale.

Class Token Une particularité du ViT réside dans l'introduction d'un jeton spécial, appelé *jeton [classe]*, inséré au début de la séquence des tokens d'entrée. Ce vecteur est un paramètre apprenable, et il est traité en parallèle avec les autres tokens tout au long du modèle. À la sortie du dernier bloc de self-attention, ce jeton est extrait et transmis au MLP Head pour produire la prédiction finale. Le jeton [classe] est conçu pour apprendre à focaliser l'attention sur les parties significatives de l'image afin de faciliter la tâche de classification.

APPLICATIONS DES VISION TRANSFORMERS (VIT)

Les Vision Transformers (ViT) connaissent un large éventail d'applications dans le domaine de la vision par ordinateur. Leur capacité à capturer des relations globales dans les données visuelles les rend particulièrement adaptés à plusieurs tâches clés, parmi lesquelles :

— **Classification d'images** : ViT est utilisé pour classer des images dans des catégories spécifiques, surpassant parfois les performances des réseaux convolutifs traditionnels.

— **Détection d'objets et segmentation** : Grâce à leur mécanisme d'attention globale, les ViT sont capables de localiser et segmenter avec précision les objets présents dans une image.

— **Reconnaissance d'actions** : ViT est employé dans l'analyse de séquences vidéo pour identifier les actions effectuées par des sujets.

— **Tâches multi-modales** : Les ViT sont intégrés dans des systèmes combinant plusieurs modalités, comme la réponse visuelle aux questions (Visual Question Answering), l'ancrage visuel (Visual Grounding), ou encore le raisonnement visuel, permettant des interactions avancées entre texte et image.

— **Modélisation générative** : ViT est également utilisé dans des modèles génératifs pour des applications telles que la génération d'images réalistes ou la prévision de séquences vidéo.

— **Traitement vidéo** : Des tâches comme la reconnaissance d'activités et la prévision vidéo bénéficient des capacités temporelles et spatiales des ViT.

— **Amélioration d'images** : Les ViT sont appliqués à des tâches de traitement d'image avancées telles que l'amélioration de la qualité d'image, la colorisation automatique et la super-résolution.

— **Analyse 3D** : Enfin, les ViT trouvent leur place dans l'analyse tridimensionnelle, notamment pour la segmentation et la classification des nuages de points.

Ces multiples cas d'utilisation soulignent la flexibilité et l'efficacité des Vision Transformers dans divers domaines de la vision par ordinateur.

2.4.7 APPROCHE DE TRANSFERT LEARNING

L'apprentissage par transfert (*Transfert Learning*) est une technique incontournable en apprentissage profond. Elle consiste à exploiter les connaissances acquises par un modèle préalablement entraîné sur une tâche source, pour les réutiliser dans la résolution d'une nouvelle tâche cible, souvent avec un jeu de données plus restreint. Ce procédé permet d'accélérer l'entraînement, de réduire les besoins en ressources computationnelles, tout en maintenant d'excellentes performances. Le concept repose sur l'idée que les représentations apprises sur un large ensemble de données sont suffisamment générales pour être transférées à d'autres contextes.

L'apprentissage par transfert est particulièrement efficace lorsqu'il est appliqué à des modèles complexes, tels que les **Vision Transformers (ViT)**, qui nécessitent habituellement un volume conséquent de données et une puissance de calcul importante pour un entraînement à partir de zéro.

Le principe général consiste à reprendre le cœur du modèle pré-entraîné (par exemple, les blocs d'attention dans le cas des ViT) et à adapter uniquement certaines couches finales pour la nouvelle tâche. Plusieurs stratégies peuvent être employées, en fonction de la taille du dataset et de sa similitude avec le dataset d'origine :

- **Fine-tuning complet**

  Cette approche consiste à remplacer la dernière couche du modèle par un classifieur

adapté au nouveau problème. L'ensemble du modèle est ensuite ré-entraîné sur le nouveau dataset. Cette méthode est recommandée lorsque le jeu de données cible est suffisamment volumineux, permettant de réapprendre sans risque de surapprentissage.

- **Extraction de caractéristiques**

  Ici, le modèle pré-entraîné est utilisé comme un extracteur de caractéristiques. Seules les représentations produites par les couches internes sont conservées, tandis que les paramètres du modèle sont gelés. Une nouvelle couche de classification est entraînée sur ces représentations. Cette technique est particulièrement adaptée lorsque le dataset cible est de petite taille et présente des similarités avec le dataset source.

- **Fine-tuning partiel**

  Un compromis entre les deux méthodes précédentes consiste à ne ré-entraîner qu'une partie des couches du modèle (généralement les plus hautes couches), tout en laissant les couches inférieures gelées. Cette stratégie est pertinente lorsque le dataset cible est de taille modérée et présente des différences notables par rapport aux données utilisées pour le pré-entraînement.

Les Vision Transformers bénéficient pleinement de ces techniques, notamment grâce à la disponibilité de nombreux modèles ViT pré-entraînés sur des jeux de données tels qu'ImageNet. Les bibliothèques modernes comme PyTorch, TensorFlow ou Keras offrent des versions pré-entraînées prêtes à être utilisées pour appliquer efficacement l'apprentissage par transfert à diverses tâches de vision par ordinateur.

2.5 CONCLUSION

Dans ce chapitre, nous avons passé en revue les principales approches de l'intelligence artificielle, en mettant l'accent sur l'apprentissage automatique, l'apprentissage profond, l'apprentissage par transfert, ainsi que sur les Transformers et leurs différentes variantes. Ces

méthodes constituent des outils puissants, largement adoptés dans de nombreux domaines, et plus particulièrement dans le domaine de la détection de d'images fausses.

Le chapitre suivant sera consacré à l'état de l'art sur les approches existantes pour la détection d'images générées, afin de situer notre travail dans le contexte des recherches actuelles.

CHAPITRE III
REVUE DE LA LITTÉRATURE

3.1 INTRODUCTION

D'importants efforts de recherche ont été réalisés dans le domaine de la détection en raison de la prolifération des deepfakes. Pour situer les contributions de cette étude, nous devons comprendre ces avancées. De nombreux documents de recherche ont fourni différents points de vue sur la manière de détecter efficacement les contenus synthétiques au moyen de méthodologies et d'algorithmes.

3.1.1 MÉTHODES BASÉES SUR LE TRANSFERT D'APPRENTISSAGE

Kumar *et al.* (2021) ont exploré la détection des images DeepFake à l'aide de réseaux convolutifs (CNN) combinés au transfert d'apprentissage. Leur approche repose sur l'utilisation de modèles CNN pré-entraînés, ajustés via fine-tuning à de nouveaux jeux de données. Les performances sont jugées satisfaisantes sur des images générées par divers GANs. Toutefois, la robustesse se dégrade significativement en présence de post-traitements (compression, flou), limitant l'usage en conditions réelles.

Al-Dulaimi & Kurnaz (2024) ont proposé une architecture hybride combinant CNN et LSTM, exploitant à la fois les caractéristiques spatiales et temporelles. Le CNN extrait les informations visuelles tandis que le LSTM modélise les séquences d'images. Cette combinaison améliore la détection dans les vidéos, mais le modèle reste lourd et peu adapté aux images fixes ou aux scénarios non faciaux.

Joshi & Nivethitha (2024) ont testé l'efficacité de l'architecture Xception, connue pour ses convolutions séparables en profondeur, sur des images prétraitées pour accentuer les artefacts GAN. Malgré de bonnes performances sur les données de test, le modèle s'avère sensible au changement de domaine (dataset shift).

Rajakumareswaran *et al.* (2024) ont également adopté Xception dans un cadre de transfert d'apprentissage. Le modèle pré-entraîné est ajusté sur des images synthétiques, et les résultats confirment sa pertinence pour la détection. Cependant, la dépendance à la qualité des données d'entraînement et l'absence d'évaluation sur images post-traitées en limitent la portée.

Dans une étude comparative, Vajpayee *et al.* (2023) ont confronté différentes architectures pré-entraînées (ResNet, Inception, Xception, InceptionResNet) sur des jeux d'images falsifiées. Xception et InceptionResNet se sont démarqués, mais l'étude reste limitée à un petit ensemble de générateurs et n'analyse pas la généralisation inter-domaines.

Abhineswari *et al.* (2024) ont évalué plusieurs modèles sur le dataset FaceForensics++, incluant MobileNetV2, ResNet50, InceptionV3, EfficientNet, Xception, NASNetMobile et un CNN personnalisé. MobileNetV2 a obtenu la meilleure performance, mais l'étude ne teste pas la robustesse à d'autres modèles GAN ni à des transformations.

3.1.2 MÉTHODES ROBUSTES ET SPÉCIALISÉES

Chen *et al.* (2021) ont proposé une méthode de détection robuste aux post-traitements, en exploitant les composantes de luminance et de chrominance dans les espaces colorimétriques RGB et YCbCr. Leur architecture Xception améliorée intègre un double flux d'entrée ainsi que des modules d'attention (CBAM) et d'agrégation multi-niveaux. Le modèle surpasse

les méthodes classiques face à des distorsions comme la compression JPEG, le flou ou les corrections gamma, bien que le coût computationnel puisse être plus élevé.

Guo *et al.* (2022) ont quant à eux développé un modèle basé sur l'analyse des reflets oculaires. À l'aide de Mask-RCNN, ils extraient les iris et comparent les reflets entre les deux yeux via un Residual Attention Network (RAN). La combinaison d'une perte cross-entropy et d'une approximation AUC (WMW) permet de gérer efficacement les déséquilibres de données. Le modèle démontre d'excellentes performances sur le dataset FFHQ-GAN, mais dépend fortement de la qualité visuelle des yeux.

Raj *et al.* (2024) ont introduit un modèle attentionnel compact exploitant deux représentations : un filtre passe-haut bilatéral (BiHPF) et la réponse non uniforme du capteur (PRNU). Ces deux canaux sont traités par un réseau à attention multi-tête. L'approche montre une nette amélioration par rapport à l'état de l'art sur plusieurs configurations, y compris des scénarios inter-GAN et avec manipulation de couleurs. Toutefois, la méthode repose sur des prétraitements spécifiques sensibles aux altérations.

Zhang *et al.* (2024) proposent l'approche X-Transfer qui repose sur une structure à double réseau neuronal (réseau maître et auxiliaire) interconnectés par une transmission alternée des gradients. Le système combine une fonction de perte croisée (cross-entropy) avec une approximation différentiable de l'AUC via les statistiques WMW. Ils enregistrent des performances élevées dans les meilleures conditions, avec un gain de près de 10 % par rapport aux méthodes classiques. Le modèle montre également une bonne généralisation sur des jeux de données non faciaux. Bien que prometteuse, la complexité du système bi-réseau peut ralentir l'entraînement.

Ura *et al.* (2023) se concentrent sur les images générées par StyleGAN. Deux méthodes sont combinées dans une version fine-tunée d'XceptionNet : l'extraction des attributs de

couleur et l'analyse des points caractéristiques (landmarks) du visage. L'étude montre une bonne différenciation entre visages réels et synthétiques, mais les résultats restent spécifiques à StyleGAN et manquent de généralisation.

Richards *et al.* (2023) ont conçu un modèle CNN pour détecter les visages modifiés, basé sur des représentations visuelles globales extraites de visages falsifiés par des outils comme Face2Face ou DeepFake. Bien que l'approche soit simple, elle ne propose pas de mécanisme de généralisation ni d'analyse face à des perturbations comme la compression JPEG.

3.1.3 MÉTHODES EXPLICATIVES ET D'INGÉNIERIE INVERSE

Marra *et al.* (2019) ont proposé une approche pionnière d'attribution des images GAN en identifiant les empreintes numériques spécifiques à chaque modèle, à la manière des capteurs photo. Ces empreintes permettent d'attribuer une image à son générateur d'origine, même après certaines manipulations. Cependant, la robustesse aux altérations reste inégale, et des conditions d'analyse précises sont nécessaires.

Asnani *et al.* (2023) ont introduit le concept de *model parsing*, une approche d'ingénierie inverse visant à estimer la structure réseau et la fonction de perte d'un modèle génératif à partir d'une image. Leur pipeline comprend un Fingerprint Estimation Network (FEN) et un Parsing Network (PN). L'approche, testée sur 100K images générées par 116 modèles, affiche des performances *état de l'art* sur les benchmarks de détection et d'attribution. Toutefois, elle dépend d'un vaste corpus d'images synthétiques étiquetées, limitant son applicabilité hors recherche.

3.2 CONCLUSION

La détection d'images générées par GAN reste un domaine en pleine évolution. Les méthodes récentes exploitant l'attention, l'apprentissage par transfert et les caractéristiques physiques ont montré des résultats prometteurs. Cependant, la généralisation à des architectures GAN inconnues et la robustesse face aux manipulations de post-traitement demeurent des défis. Cette étude propose un modèle de détection basé sur les Vision Transformers et l'attention latente pour améliorer la détection des images DeepFake.

CHAPITRE IV
MÉTHODOLOGIE

4.1 INTRODUCTION

Dans ce chapitre, nous décrivons l'approche méthodologique adoptée pour la détection d'images faciales générées par StyleGAN. Nous commençons par présenter l'architecture de base du Vision Transformer (ViT). Ensuite, nous présentons le module d'attention latente intégré et les raisons de cette intégration. Enfin, nous détaillons les jeux de données utilisés ainsi que les paramètres liés à l'entraînement du modèle.

L'architecture complète de notre modèle est illustrée à la Figure 4.1.

4.2 ARCHITECTURE DU MODÈLE PROPOSÉ

4.2.1 VISION TRANSFORMER PRÉ-ENTRAÎNÉ

Le Vision Transformer (ViT) Dosovitskiy *et al.* (2020) est un modèle de classification d'images de pointe qui utilise l'architecture du transformateur, développée à l'origine pour les tâches de traitement du langage naturel, pour traiter les images. Il vise à minimiser la perte d'entropie croisée entre les probabilités de classe prédites et les véritables étiquettes de classe. L'architecture standard de ViT comprend les étapes suivantes :

— **Patch Embedding** : L'image est divisée en parcelles de taille fixe (par exemple, 16 x 16 pixels) et chaque parcelle est transformée en un vecteur de dimension d par projection linéaire.

— **Codage positionnel** : Les encodements de position sont ajoutés aux patchs afin de préserver les informations spatiales.

— **Transformer Encoder** : Les patchs sont traités par une série de blocs Transformer, chacun composé d'une couche d'attention à plusieurs têtes et d'une couche d'anticipation.

— **Tête de classification** : Les caractéristiques globales sont extraites (par exemple, à l'aide du jeton `[CLS]`) et passent par une couche de classification pour produire les logits finaux.

Cependant, l'auto-attention multi-têtes dans le ViT a une complexité de calcul de $\mathcal{O}(N^2)$, où N est le nombre de patchs. Cela limite son application aux images à haute résolution ou aux longues séquences. Pour surmonter cette limitation, nous remplaçons cette couche dans notre architecture par un module d'attention latente.

Notre approche s'appuie sur un ViT pré-entraîné sur le dataset ImageNet. Cela permet de tirer parti des caractéristiques générales apprises lors du pré-entraînement, tout en réduisant les coûts computationnels lors de l'entraînement sur nos données.

4.2.2 INTÉGRATION DU MODULE D'ATTENTION LATENTE

Les couches MHSA sont remplacé par un module d'attention latente (*Latte*) inspiré des travaux de Dolga *et al.* (2024). Ce mécanisme repose sur une paramétrisation de faible dimension de la matrice d'attention classique.

Étant donné une séquence d'entrée $X \in \mathbb{R}^{T \times D}$, nous introduisons L vecteurs latents appris $\{w_l^q, w_l^k\}_{l=1}^{L}$ qui servent d'intermédiaires dans le calcul des similarités :

$$p(l|t) = \frac{\exp(x_t^\top w_l^q)}{\sum_{j=1}^{L} \exp(x_t^\top w_j^q)}, \quad p(s|l) = \frac{\exp(x_s^\top w_l^k)}{\sum_{s=1}^{T} \exp(x_s^\top w_l^k)} \tag{4.1}$$

La sortie est obtenue par une combinaison convexe des valeurs projetées :

$$\tilde{x}_t = \sum_{l=1}^{L} p(l|t) \sum_{s=1}^{T} p(s|l) v_s \tag{4.2}$$

COMPLEXITÉ COMPUTATIONNELLE

Contrairement à l'attention standard dont la complexité est $\mathcal{O}(T^2 D)$, l'attention latente réduit cette complexité à :

— Temps : $\mathcal{O}(TLD)$ (linéaire en T)

— Mémoire : $\mathcal{O}(TL + LD)$

où $L \ll T$ est la dimension latente. Ce gain permet le traitement d'images haute résolution tout en maintenant une expressivité suffisante.

INTÉGRATION AU MODÈLE VIT

Concrètement, les étapes suivantes sont appliquées :

1. Projection des tokens d'entrée dans l'espace latent via W_q et W_k.
2. Calcul des distributions de probabilité latentes $p(l|t)$ et $p(s|l)$.
3. Agrégation contextuelle par convolution en profondeur afin d'assurer la diversité des représentations.

MOTIVATION DU CHOIX

L'objectif principal de l'intégration de l'attention latente est de réduire la complexité quadratique de la MHSA classique, tout en évitant la perte de performance qu'entraînerait la

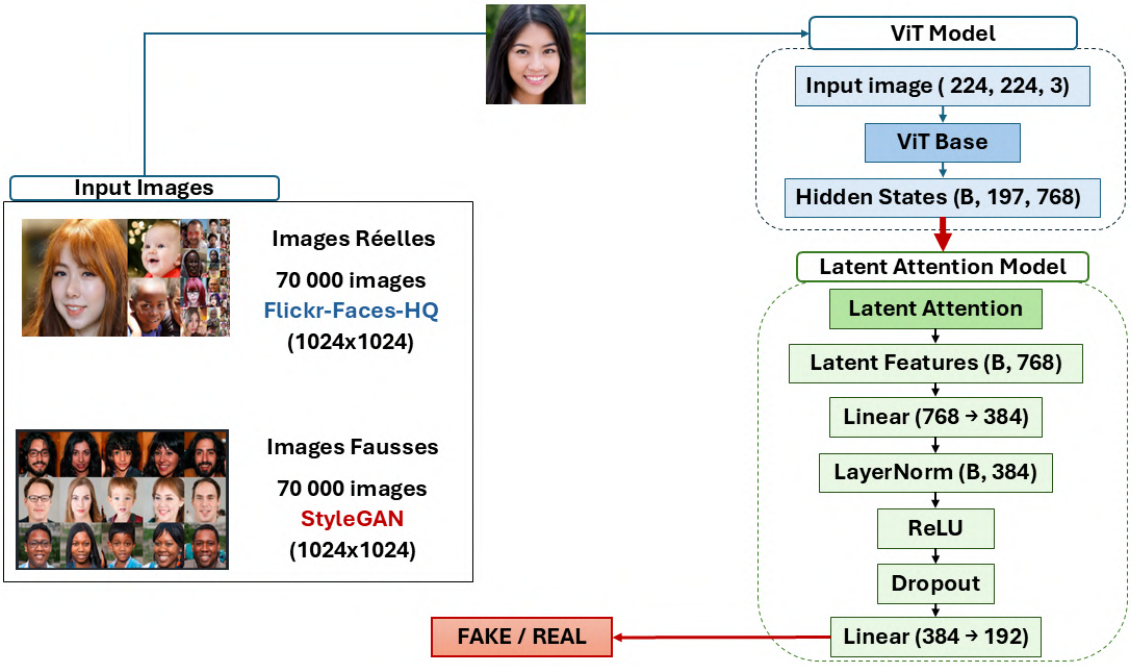

FIGURE 4.1 : Architecture complète du modèle proposé intégrant l'attention latente

simple désactivation de cette couche. Le module d'attention latente permet de conserver la capacité du modèle à capturer les dépendances globales, tout en étant plus efficient sur des images haute résolution.

4.3 JEUX DE DONNÉES

Pour détecter les images générées par StyleGAN, il est nécessaire d'avoir une collection de données d'images réelles et fausses. Le principal dataset utilisé est, dans ce travail, le **140k Real and Fake Faces** [1]. Il contient 70 000 images de visages réels issues du dataset Flickr-Faces-HQ, ainsi que 70 000 images fausses générées par StyleGAN, avec des valeurs PSI de 0.5, 0.7 et 1.0.

1. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces

Ce choix se justifie par le fait que StyleGAN, constitue aujourd'hui l'une des architectures les plus avancées pour la génération d'images faciales. Les visages produits sont indiscernables à l'œil nu (voir Figure 4.2), ce qui représente un véritable défi pour les systèmes de détection de deepfake. Par ailleurs, StyleGAN introduit des mécanismes de contrôle fins sur l'apparence (via l'espace latent), rendant la diversité des images générées plus importante et leur détection plus complexe. Compte tenu de sa large adoption dans les contenus synthétiques en ligne et de la disponibilité de datasets publics fondés sur ce modèle, il est donc pertinent de centrer nos efforts sur cette architecture. Des travaux comparatifs tels que ceux de Karras *et al.* (2021), Asnani *et al.* (2023), ou Wang *et al.* (2020) confirment la supériorité de la famille StyleGAN sur d'autres GANs, notamment ProGAN, BigGAN ou CycleGAN, en termes de qualité visuelle, diversité, et complexité de détection.

Les données sont réparties de manière équilibrée : 100 000 images pour l'entraînement (50 000 réelles et 50 000 fausses), 20 000 pour la validation (10 000 réelles et 10 000 fausses), et 20 000 pour le test (10 000 réelles et 10 000 fausses).

Pour évaluer la robustesse de notre modèle, nous avons aussi utilisé les datasets suivants :

1. L'ensemble de données introduit par Wang *et al.* (2020) contenant une variété d'images générées par des GANs, incluant ProGAN, StyleGAN, StyleGAN2, BigGAN, CycleGAN, et StarGAN, ainsi que des images réelles provenant de LSUN et FFHQ. Les images ayant des résolutions variables, elles sont toutes redimensionnées à 256×256 pixels.
2. L'ensemble de données *deepfake_faces* de Kaggle, qui est une collection de plus de 90 000 images réelles et fausses de visages humains.

Le résumé de ces ensembles de données est présenté dans le Tableau 4.1.

TABLEAU 4.1 : Jeux de données utilisés pour l'entraînement et l'évaluation

Dataset	Description	Taille originale	Taille redimensionnée	Nombre d'images	Utilisation
140k Real and Fake Faces	70k visages réels (Flickr) et 70k visages synthétiques (StyleGAN)	256 x 256	224 x 224	140 000	Entraînement et test
Wang *et al.* (2020) Dataset [2]	Images générées par ProGAN, BigGAN, StyleGAN ; images réelles issues de LSUN et ImageNet	256 x 256	224 x 224	5 626	Test uniquement
Joshi & Nivethitha (2024) Deepfake Faces [3]	Visages réels et synthétiques humains	224 x 224	299 x 299	16 293	Test uniquement

FIGURE 4.2 : Images réelles et fausses

4.4 DÉTAILS D'IMPLÉMENTATION

Le modèle est entraîné en utilisant la fonction de perte par entropie croisée et l'optimiseur AdamW. Les couches MHSA du ViT sont gelées, permettant un ajustement fin via notre module d'attention latente.

Les étapes de prétraitement des images sont les suivantes :

1. **Redimensionnement** à 224×224 pixels.
2. **Conversion en tenseurs** PyTorch.
3. **Normalisation** avec les statistiques d'ImageNet.

4. https://github.com/PeterWang512/CNNDetection/tree/master
5. https://www.kaggle.com/datasets/dagnelies/deepfake-faces

Pour améliorer la capacité de généralisation, plusieurs techniques d'augmentation de données sont appliquées sur l'ensemble d'entraînement :

— Inversion horizontale aléatoire (probabilité de 50%).

— Rotations aléatoires de ±15 degrés.

— Modification aléatoire de la luminosité, du contraste et de la saturation (color jitter).

— Découpage redimensionné aléatoire.

Les hyperparamètres ont été optimisés avec la bibliothèque Optuna. Les paramètres explorés incluent :

— Taux d'apprentissage : $1e^{-5}$ à $1e^{-3}$

— Taux de dropout : 0.1 à 0.5

— Nombre de têtes d'attention : 64 à 256

— Taille du noyau de convolution : 1 à 5

Les meilleurs hyperparamètres retenus sont :

— Taux d'apprentissage : 5×10^{-5}

— Nombre de têtes d'attention : 128

— Noyau de convolution : 3

— Dropout : 0.1

— Batch size : 32

L'entraînement a été réalisé sur 10 époques, comme le montre la Figure 5.1, en utilisant un notebook Google Colab avec un GPU T4.

4.5 CONCLUSION

Dans ce chapitre, nous avons présenté l'approche méthodologique adoptée pour la détection d'images synthétiques. Nous avons détaillé l'intégration d'un module d'attention

latente dans une architecture ViT pré-entraînée, justifiant ce choix par des considérations d'efficacité computationnelle et de performance. Nous avons également présenté les jeux de données utilisés ainsi que les étapes de prétraitement, d'augmentation de données et d'optimisation des hyperparamètres. Le chapitre suivant exposera les résultats expérimentaux obtenus, ainsi qu'une analyse comparative des performances du modèle proposé.

CHAPITRE V

RÉSULTATS ET EXPLICABILITÉ

5.1 INTRODUCTION

Dans ce chapitre, nous présentons dans le Tableau 5.1 un résumé des performances de notre modèle selon les différentes mesures d'évaluation appliquées à plusieurs ensembles de données (voir Tableau 4.1). Nous analysons ensuite ces résultats et les comparons aux modèles de pointe actuels. Une étude d'ablation est également menée afin d'évaluer l'impact de certains paramètres sur la performance du modèle. Nous proposons aussi l'explicabilité du modèle, permettant une meilleure compréhension des décisions prises lors de la détection. Enfin, on fait une discussion sur les résultats et les limites de notre approche. Un dépôt Github contenant un lien vers le code de tous les résultats de cette section est fourni par ce lien[6].

5.2 MÉTRIQUES D'ÉVALUATION

Pour évaluer les performances de notre modèle, plusieurs métriques classiques et complémentaires sont utilisées. Ces métriques permettent non seulement de mesurer la capacité globale du modèle à effectuer des prédictions correctes, mais également d'analyser finement son comportement face aux faux positifs et aux faux négatifs, ce qui est crucial dans le cadre de la détection d'images synthétiques. Les principales métriques employées sont les suivantes :

— **Exactitude (Accuracy)**

Elle mesure la proportion de prédictions correctes par rapport au nombre total de prédictions effectuées.

6. https://github.com/Thioumbour/Detecting-StyleGAN-Generated-Deepfake-Faces-with-ViT-and-LA.git

$$Exactitude = \frac{VP + VN}{VP + VN + FP + FN} \tag{5.1}$$

— **F1-Score**

Le F1-Score est la moyenne harmonique entre la précision et le rappel. Il est particulièrement utile dans les contextes où les classes sont déséquilibrées.

$$F1\text{-}score = 2 \times \frac{Prcision \times Rappel}{Prcision + Rappel} \tag{5.2}$$

avec :

— **Précision =** $\frac{VP}{VP+FP}$ (mesure la proportion de prédictions positives correctes).

— **Rappel =** $\frac{VP}{VP+FN}$ (mesure la proportion de vrais positifs correctement identifiés).

— **Kappa de Cohen**

Le coefficient Kappa mesure l'accord entre les prédictions du modèle et les données réelles, en tenant compte de l'accord attendu par hasard.

$$Kappa = \frac{p_0 - p_e}{1 - p_e} \tag{5.3}$$

— **AUC (Surface Sous la Courbe ROC)**

L'AUC mesure la capacité du modèle à distinguer les classes en traçant la courbe ROC (*Receiver Operating Characteristic*). La courbe ROC représente le taux de vrais positifs (TPR) en fonction du taux de faux positifs (FPR) pour différents seuils de classification. L'AUC correspond à l'aire sous cette courbe.

$$AUC = \frac{1 + \left(\frac{VP}{VP+FN}\right) - \left(\frac{FP}{VN+FP}\right)}{2} \tag{5.4}$$

5.3 PERFORMANCES DU MODÈLE

Comme le montre le Tableau 5.1, le modèle proposé surpasse les approches traditionnelles basées sur le CNN et d'autres méthodes récentes de détection des deepfakes pour toutes les évaluations métriques.

Pour assurer une comparaison rigoureuse et équitable, les modèles de référence Xception Joshi & Nivethitha (2024), CNN Kumar *et al.* (2021), DFFD Richards *et al.* (2023) et XTransfer Zhang *et al.* (2024) ont été réimplémentés par nos soins, en suivant fidèlement les méthodologies décrites dans les publications originales. Ces modèles ont ensuite été entraînés et évalués sur nos propres jeux de données, selon les mêmes protocoles d'expérimentation, assurant ainsi l'homogénéité des conditions de test et la validité des comparaisons effectuées.

Plusieurs métriques nous ont permis d'évaluer les performances de notre modèle. Dans la Figure 5.1 on peut voir la perte et l'exactitude au fil des époques des ensembles d'entraînement et de validation. Nous observons une nette augmentation de l'exactitude et qui se stabilise vers les 99.8%. De même, la perte diminue au fil des époques pour se stabiliser à un niveau proche de zéro. On peut en conclure que le modèle apprend sans signe flagrant de suraprentissage.

La courbe ROC est présentée dans la Figure 5.2 et atteint une aire sous la courbe de 1. Ceci traduit, la capacité de notre modèle a distinguer les images réelles des fausses. La Figure 5.3 montre la matrice de confusion de l'ensemble de données test. Nous observons 10 erreurs de classement sur 20 000 images testées, ce quidémontre la robustesse de notre modèle.

Sur le jeu de données principal (140k images), notre modèle atteint une exactitude de 99.83%, surpassant nettement le modèle Xception (98%) ainsi que le modèle CNN (87%). Il dépasse également le modèle XTransfer, qui n'atteint que 85.4% d'exactitude, avec un F1-score de 87.2%, un AUC-ROC de 98.2% et une log loss de 0.378. Une précision élevée

(99.82%) combinée à un rappel de 99.84% témoigne de la capacité remarquable du modèle à identifier les visages synthétiques tout en minimisant les faux négatifs. Le F1-score obtenu de 99.83% reflète un équilibre optimal entre précision et rappel, aspect crucial dans les contextes de classes déséquilibrées. Par ailleurs, l'AUC-ROC de 1 démontre l'excellente capacité du modèle à distinguer les visages réels des visages synthétiques, même sous différents seuils de classification. Le coefficient Kappa de Cohen, égal à 99.66%, confirme un fort accord avec les données réelles, et une faible perte logarithmique (log loss) de 0.008 souligne la confiance des prédictions du modèle.

Afin d'évaluer la capacité de généralisation de notre modèle, nous l'avons testé sur deux jeux de données externes : le dataset Deepfake Faces ainsi que le dataset proposé par Wang *et al.* (2020). Les résultats obtenus sont résumés comme suit :

— **Dataset Deepfake Faces Joshi & Nivethitha (2024)** : Notre modèle atteint une exactitude de 90.50% et un AUC-ROC de 0.9693. Cette performance dépasse significativement celle des architectures de référence telles que Xception (66%) et les modèles basés sur CNN (49.8%), démontrant ainsi une forte robustesse face à des données inédites. Le modèle XTransfer Zhang *et al.* (2024) affiche, sur ce dataset, une exactitude de 79.5%, un F1-score de 80%, un AUC-ROC de 0.882 et une log loss de 0.465, confirmant une amélioration par rapport aux modèles classiques, mais encore inférieure à celle de notre approche.

— **Dataset proposé par Wang *et al.* (2020)** : Sur ce jeu de données, le modèle obtient une exactitude de 99.75% et un AUC-ROC de 0.9975, confirmant sa capacité d'adaptation à diverses architectures de GAN. Le modèle XTransfer, quant à lui, n'obtient qu'une exactitude de 77.5%, malgré une précision de 98.2%, son faible rappel (56%) et un F1-score de 71.3% révèlent une difficulté à identifier systématiquement les fausses

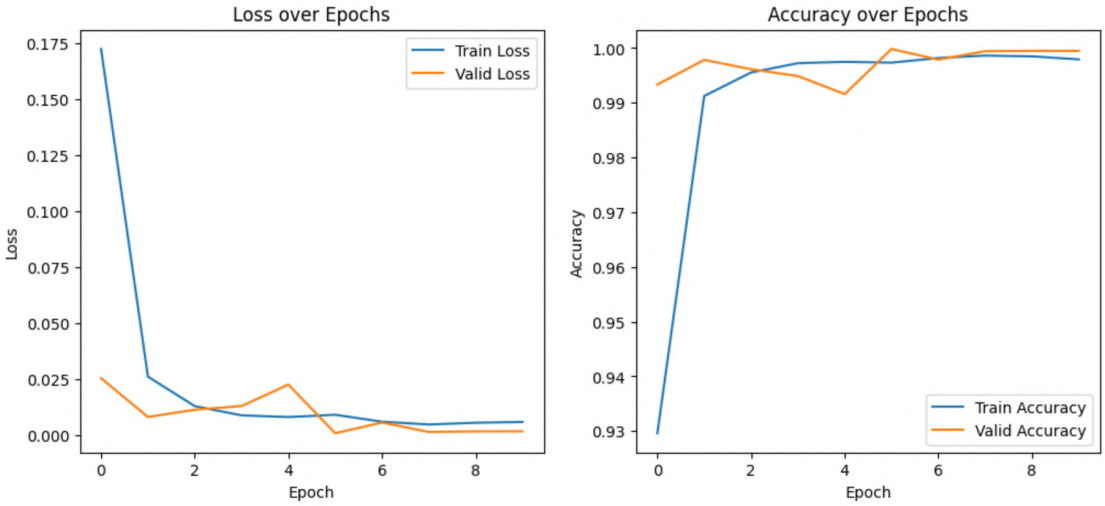

FIGURE 5.1 : Courbes d'accuracy et de perte (entraînement et de validation)

images. La log loss, relativement élevée (0.836), reflète une incertitude accrue dans les prédictions.

D'autre part, le temps de classification confirme que le module Latent Attention permet de capturer les relations clés tout en offrant un coût de calcul nettement inférieur, comme on l'avait d'abord supposé, puis comme le montre le Tableau 5.1.

Les performances démontrées confirment donc le faible coût de calcul et la pertinence de l'approche hybride ViT-Latent Attention que nous avons proposée pour la détection de visages synthétiques. Le modèle se distingue par son excellente capacité de classification, sa bonne généralisation aux données externes et son efficacité de calcul accrue.

TABLEAU 5.1 : Résultats de nos différentes expérimentations

Model	Accuracy	Precision	Recall	F1-score	Specifity	AUC-ROC	MCC	Cohen Kappa	Log Loss	Temps (s)
140k real and fake faces dataset										
Modèle proposé	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**	**1**	**0.996**	**0.996**	**0.008**	**242**
Modèle ViT	0.986	0.986	0.998	0.992	0.986	0.985	0.986	0.985	0.02	415
ViT + GAP	0.983	0.989	0.986	0.991	0.990	0.998	0.976	0.978	0.02	450
Modèle CNN Kumar *et al.* (2021)	0.870	0.860	0.880	0.870	0.860	0.850	0.830	0.830	0.55	382
Modèle DFFD Richards *et al.* (2023)	0.970	0.970	0.980	0.990	0.960	0.980	0.920	0.920	0.01	200
Modèle Xception Joshi & Nivethitha (2024)	0.980	0.980	0.980	0.980	0.980	0.990	0.960	0.960	0.04	456
Modèle Iris Guo *et al.* (2022)	0.930	0.940	0.920	0.930	0.940	0.980	0.860	0.9860	0.53	349
Modèle Xtransfer Zhang *et al.* (2024)	0.854	0.778	0.991	0.872	0.717	0.982	0.737	0.708	0.378	410
Joshi & Nivethitha (2024) Deepfake Faces Dataset										
Modèle proposé	**0.905**	**0.913**	**0.895**	**0.904**	**0.914**	**0.969**	**0.810**	**0.810**	**0.371**	**80**
Modèle ViT	0.860	0.830	0.900	0.870	0.820	0.940	0.730	0.720	0.34	79
ViT + GAP	0.854	0.977	0.726	0.832	0.982	0.977	0.733	0.708	0.328	75
Modèle CNN Kumar *et al.* (2021)	0.499	0.478	0.483	0.495	0.523	0.488	0.352	0.352	0.72	96
Modèle DFFD Richards *et al.* (2023)	0.920	0.890	0.880	0.910	0.960	0.890	0.860	0.860	0.3	79
Modèle Xception Joshi & Nivethitha (2024)	0.660	0.780	0.440	0.560	0.880	0.760	0.350	0.310	0.61	560
Modèle Iris Guo *et al.* (2022)	0.640	0.620	0.760	0.690	0.520	0.710	0.300	0.290	0.63	70
Modèle Xtransfer Zhang *et al.* (2024)	0.795	0.784	0.816	0.800	0.775	0.882	0.592	0.591	0.465	76
Wang *et al.* (2020) Dataset										
Modèle proposé	**0.997**	**0.995**	**1**	**0.997**	**0.995**	**0.997**	**0.995**	**0.995**	**0.039**	**9**
Modèle ViT	0.977	0.979	0.975	0.977	0.980	0.998	0.955	0.955	0.055	63
ViT + GAP	0.979	0.974	0.984	0.979	0.973	0.997	0.958	0.958	0.06	65
Modèle CNN Kumar *et al.* (2021)	0.485	0.466	0.473	0.475	0.423	0.462	0.402	0.402	0.73	78
Modèle DFFD Richards *et al.* (2023)	0.820	0.850	0.860	0.890	0.760	0.910	0.720	0.720	0.5	68
Modèle Xception Joshi & Nivethitha (2024)	0.650	0.620	0.760	0.680	0.530	0.700	0.300	0.300	0.64	61
Modèle Iris Guo *et al.* (2022)	0.750	0.810	0.660	0.730	0.850	0.850	0.520	0.510	0.61	70
Modèle Xtransfer Zhang *et al.* (2024)	0.775	0.982	0.560	0.713	0.990	0.962	0.609	0.550	0.836	85

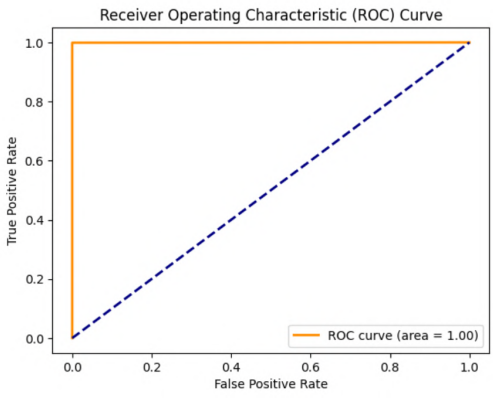

FIGURE 5.2 : Courbe ROC

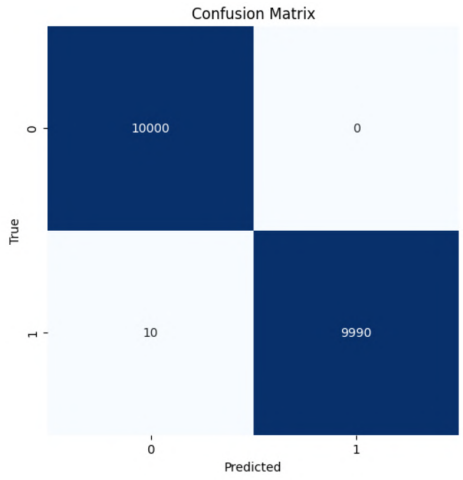

FIGURE 5.3 : Matrice de confusion

5.4 ÉTUDE D'ABLATION

5.4.1 IMPACT DU MODULE D'ATTENTION LATENTE

L'intégration du module d'attention latente vise à améliorer la capacité du Vision Transformer à se concentrer sur les caractéristiques latentes les plus informatives, augmentant ainsi à la fois la précision et l'efficacité du modèle.

Les résultats expérimentaux montrent que le **ViT avec attention latente** surpasse systématiquement le modèle de base **ViT seul**. L'exactitude est améliorée, atteignant 99.83% contre 98.63% pour le ViT standard. Bien que le rappel du ViT classique soit légèrement supérieur (99.88% contre 99.51%), le modèle intégrant l'attention latente obtient une meilleure précision (99.01% contre 98.64%) ainsi qu'un F1-score légèrement plus élevé (99.83% contre 99.25%), témoignant d'un meilleur équilibre entre faux positifs et faux négatifs. De plus, le score AUC-ROC passe significativement de 0.9859 à 1, démontrant une capacité accrue de discrimination entre les classes.

L'intégration du module d'attention latente réduit également considérablement le temps d'inférence, avec une diminution de 42% (242 secondes contre 415 secondes), ce qui souligne son efficacité et sa pertinence pour des applications en temps réel.

Ces résultats confirment que le module d'attention latente capture efficacement les relations spatiales et contextuelles essentielles entre les images tout en réduisant le coût computationnel. Son intégration améliore non seulement les performances de classification, mais optimise également l'efficacité du modèle, renforçant ainsi la valeur de l'architecture proposée.

5.4.2 IMPACT DU GLOBAL AVERAGE POOLING (GAP)

Afin d'évaluer l'impact de l'ajout d'une couche de Global Average Pooling (GAP) suivie de couches denses sur l'agrégation des caractéristiques et les performances de classification, nous avons comparé notre modèle proposé avec une variante intégrant GAP. Les résultats montrent que cette modification n'apporte pas d'amélioration significative, voire conduit à une légère dégradation des performances globales.

En effet, le modèle **ViT + GAP** présente une exactitude de 98.3%, inférieure à celle du **modèle proposé** (99.83%). De même, les métriques de précision, rappel, F1-score et spécificité sont toutes légèrement en baisse avec l'ajout de GAP. La Log Loss stagne à 0.02 contre 0.008 pour notre modèle, traduisant une perte plus importante. Par ailleurs, les indicateurs de robustesse comme le MCC (0.976 contre 0.996) et le Kappa de Cohen (0.978 contre 0.996) révèlent une diminution de la qualité de la classification. Enfin, le temps de classification est quasiment doublé (450s contre 242s), soulignant un coût computationnel supplémentaire non justifié.

Ces résultats suggèrent que la compression spatiale induite par GAP limite la capacité du Vision Transformer à capturer les détails fins, pourtant essentiels pour repérer les artefacts subtils générés par StyleGAN. Dans notre contexte, l'ajout de GAP nuit à la détection précise des fausses images. Il est donc préférable de conserver l'architecture originale du ViT sans cette opération de réduction spatiale pour maximiser la performance du modèle.

5.5 EXPLICABILITÉ DU MÉCANISME D'ATTENTION

Afin de mieux comprendre les régions sur lesquelles notre modèle se concentre lors de la détection, nous avons utilisé la méthode *Gradient-weighted Class Activation Mapping* (Grad-CAM). Cette technique permet de visualiser les zones de l'image ayant le plus

influencé la décision du modèle. Elle repose sur le calcul des gradients des couches intermédiaires du réseau, permettant ainsi de mettre en évidence les régions discriminantes.

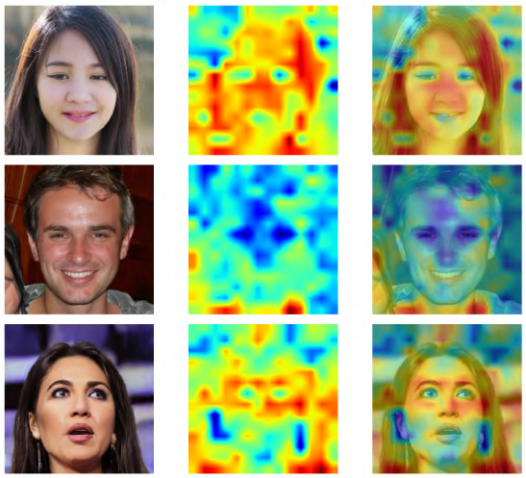

FIGURE 5.4 : Visages générées par StyleGAN détectées

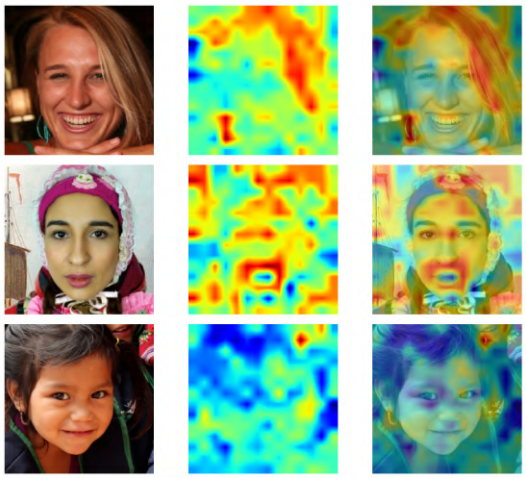

FIGURE 5.5 : Visages réelles détectées

Pour les images fausses, notre modèle porte une attention particulière à certaines régions du visage, telles que les yeux et la bouche (voir Figure 5.4), où les artefacts spécifiques à StyleGAN sont les plus présents. Ces résultats confirment que le modèle ne se contente pas d'apprendre à partir des textures globales, mais qu'il identifie des anomalies locales caractéristiques des images générées.

À l'inverse, pour les images réelles, l'attention du modèle est plus uniformément répartie (voir Figure 5.5). En effet, elle se diffuse sur l'ensemble du visage, sans concentration excessive sur des zones spécifiques. Cela démontre que les images authentiques ne présentent pas d'artefacts détectables, confirmant ainsi la capacité du modèle à ne pas surinterpréter les détails dans les visages réels.

L'application de Grad-CAM met en évidence que le modèle est capable d'identifier et de se concentrer sur les caractéristiques déterminantes pour distinguer efficacement les images réelles des images synthétiques. Ce mécanisme d'explicabilité permet d'obtenir une meilleure compréhension des décisions du modèle, renforçant ainsi la fiabilité de ses prédictions.

5.6 DISCUSSIONS ET LIMITES

Le modèle proposé parvient a identifier les images fausses avec une exactitude de 99.83% sur un temps d'inférence relativement court. Le module d'attention latente intégré à l'architecture a optimisé la détection de zones discriminantes tout en réduisant le coût computationnel. Avec le Grad-CAM, on constate que le modèle met l'accent sur des régions spécifique du visage pour prédire si une image est fausse. Par contre pour une image réelle, l'attention est uniforme et donc il ne surinterprète pas les détails présents dans les visages réelles. Ceci démontre, la capacité du modèle a identifier efficacement, en ce concentrant sur les régions discriminantes, les images fausses des réelles.

L'ablation study a démontré que le modèle proposé est plus léger et moins gourmands en ressources qu'un ViT normal ou un ViT + GAP. En effet, le module d'attention améliore les performances et réduit le temps d'inférence de 42% par rapport à un ViT normal. Le modèle proposé est à la fois performante avec une complexité de calcul réduite mais aussi explicable.

Bien que les résultats obtenus soient encourageants, plusieurs pistes d'amélioration peuvent être envisagées pour prolonger ce travail. Tout d'abord, le modèle développé se limite à la détection d'images fixes. On ne peut garantir avoir les mêmes performances sur des vidéos. Ayant utilisé une résolution fixe, les résultats du modèle pourrait être biaisé avec des images de basses résolution ou compressées.

Les images synthétiques contenu dans l'ensemble de données d'entraînement ont été générées par StyleGAN. Ce qui peut limite son extension a d'autres modèles de génération d'images comme les modèles de diffusion Mao *et al.* (2023).

Les excellente performances du modèle sur les jeux de données étudiés doivent être relativisées au vu des limites actuelles. Ces dernières fournissent des pistes d'amélioration claires pour de futures recherches et aussi pour renforcer la robustesse et la généralisation de l'approche proposé.

5.7 CONCLUSION

Ce chapitre a permis d'évaluer l'efficacité du modèle proposé, basé sur un Vision Transformer avec attention latente. Les résultats montrent une nette amélioration des performances par rapport aux approches classiques, tout en réduisant la complexité computationnelle. L'étude d'ablation a confirmé l'apport du module d'attention latente et souligné que l'ajout du GAP n'était pas pertinent. Enfin, l'utilisation de Grad-CAM a validé la capacité du modèle à se focaliser sur les régions critiques des visages, renforçant l'explicabilité et la fiabilité des prédictions.

CONCLUSION

Dans ce mémoire, nous avons étudié la détection d'images faciales générées par StyleGAN en nous appuyant sur l'architecture des Vision Transformers (ViT). Après avoir présenté les fondements des Transformers et les évolutions récentes dans le domaine de la vision par ordinateur, nous avons proposé une amélioration du modèle ViT en intégrant un module d'attention latente.

Ce dernier remplace les couches d'attention multi-tête du ViT classique, et est plus efficaces pour capturer les relations essentielles tout en réduisant significativement la complexité computationnelle. Le modèle proposé démontre des performances remarquables, atteignant une exactitude de 99.83%, un F1-score de 99.83% et un AUC de 1 sur le jeu de données principal. Le temps d'inférence peut être réduit de 42% pour atteindre une réduction de 85% pour un ensemble de données spécifique. Il montre également une forte capacité de généralisation sur des jeux de données externes, surpassant plusieurs modèles de référence dans l'état de l'art.

L'étude d'ablation a permis de valider l'impact positif du module d'attention latente, ainsi que l'importance de conserver la structure du ViT sans ajout de couches supplémentaires telles que le Global Average Pooling. Par ailleurs, l'application d'un mécanisme d'explicabilité (Grad-CAM) a confirmé que le modèle se focalise sur des régions pertinentes, renforçant ainsi la confiance dans ses décisions.

Cependant, des défis subsistent. L'extension de l'approche à la détection de vidéos truquées, en intégrant des modules capables de capturer les relations temporelles.

De plus, il serait pertinent d'explorer l'adaptation du modèle à des images de basse résolution ou compressées, afin de garantir son efficacité dans des environnements contraints ou

en présence de contenus dégradés. L'intégration de techniques d'apprentissage non supervisé ou auto-supervisé pourrait également permettre d'améliorer la robustesse et la généralisation du modèle face à des données inconnues.

Enfin, une perspective intéressante serait d'intégrer le modèle proposé au sein de systèmes d'authentification de contenus multimodaux, combinant texte, image et vidéo, afin de proposer des solutions complètes et applicables à des cas d'usage concrets.

Relevé ces défis, rendrait le modèle proposé plus robuste, plus générale et plus complet pour détecter les images synthétiques. Ainsi, face a des IA de plus en plus efficaces pour générer des images réalistes, le modèle pourrait jouer un rôle majeur dans la lutte contre les contenus deepfake.

BIBLIOGRAPHIE

Abhineswari, M., Charan, K. S., Shrikarti, B. *et al.* (2024). Deep Fake Detection using Transfer Learning : A Comparative study of Multiple Neural Networks. Dans *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, pp. 1–6. IEEE.

Acheampong, F. A., Nunoo-Mensah, H. & Chen, W. (2021). Transformer models for text-based emotion detection : a review of BERT-based approaches. *Artificial Intelligence Review*, *54*(8), 5789–5829.

Al-Dulaimi, O. A. H. H. & Kurnaz, S. (2024). A hybrid CNN-LSTM approach for precision deepfake image detection based on transfer learning. *Electronics*, *13*(9), 1662.

Amazon Web Services (2022). *Qu'est-ce qu'un GAN (Generative Adversarial Network) ?* Consulté le 16 février 2024, Repéré à https://aws.amazon.com/fr/what-is/gan/

Asnani, V., Yin, X., Hassner, T. & Liu, X. (2023). Reverse engineering of generative models : Inferring model hyperparameters from generated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(12), 15477–15493.

Ba, J. L., Kiros, J. R. & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv :1607.06450*.

Bai, Y. (2022). RELU-function and derived function review. Dans *SHS web of conferences*, Vol. 144, p. 02006. EDP Sciences.

Berroukham, A., Housni, K. & Lahraichi, M. (2023). Vision transformers : A review of architecture, applications, and future directions. Dans *2023 7th IEEE Congress on Information Science and Technology (CiSt)*, pp. 205–210. IEEE.

Bunod, R., Augstburger, E., Brasnu, E., Labbe, A. & Baudouin, C. (2022). Intelligence artificielle et glaucome : une revue de la littérature. *Journal Français d'Ophtalmologie*, *45*(2), 216–232.

Chen, B., Liu, X., Zheng, Y., Zhao, G. & Shi, Y.-Q. (2021). A robust GAN-generated face

detection method based on dual-color spaces and an improved Xception. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(6), 3527–3538.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D. & Sutskever, I. (2020). Generative pretraining from pixels. Dans *International conference on machine learning*, pp. 1691–1703. PMLR.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S. & Choo, J. (2018). Stargan : Unified generative adversarial networks for multi-domain image-to-image translation. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797.

Chollet, F. & Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

Dash, A., Ye, J. & Wang, G. (2023). A review of generative adversarial networks (GANs) and its applications in a wide variety of disciplines : from medical to remote sensing. *IEEE Access*, *12*, 18330–18357.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. Dans *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*, pp. 4171–4186.

Dolga, R., Maystre, L., Cobzarenco, M. & Barber, D. (2024). Latte : Latent Attention for Linear Time Transformers. *arXiv preprint arXiv :2402.17512*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. *et al.* (2020). An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*.

Ghojogh, B. & Ghodsi, A. (2020). Attention mechanism, transformers, BERT, and GPT : tutorial and survey.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Networks. *Science Robotics*, pp. 2672–2680. Repéré à https://arxiv.org/abs/1406.2661v1

Guandamatoko (2019). *Le perceptron multicouches | Le perceptron multi-couches - Deep learning*. Repéré à https://kongakura.fr/article/Le-perceptron-multicouches

Guo, H., Hu, S., Wang, X., Chang, M.-C. & Lyu, S. (2022). Robust attentive deep neural network for detecting gan-generated faces. *IEEE Access*, *10*, 32574–32583.

Hamid, Y., Elyassami, S., Gulzar, Y., Balasaraswathi, V. R., Habuza, T. & Wani, S. (2023). An improvised CNN model for fake image detection. *International Journal of Information Technology*, *15*(1), 5–15.

Joshi, P. & Nivethitha, V. (2024). Deep Fake Image Detection using Xception Architecture. Dans *2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST)*, pp. 533–537. IEEE.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J. & Aila, T. (2021). Alias-free generative adversarial networks. *Advances in neural information processing systems*, *34*, 852–863.

Kassel, R. (2022). *StyleGAN : Le GAN de génération d'images réalistes*. Consulté le 16 février 2024, Repéré à https://datascientest.com/stylegan

Kim, T., Cha, M., Kim, H., Lee, J. K. & Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks. Dans *International conference on machine learning*, pp. 1857–1865. Pmlr.

Kumar, N., Pranav, P., Nirney, V. & Geetha, V. (2021). Deepfake image detection using CNNs and transfer learning. Dans *2021 International Conference on Computing, Communication and Green Engineering (CCGE)*, pp. 1–6. IEEE.

Lucic, M., Kurach, K., Michalski, M., Gelly, S. & Bousquet, O. (2018). Are GANs Created Equal ? A Large-Scale Study. Dans *Advances in Neural Information Processing Systems*, Vol. 31, pp. 700–709. Repéré à https://arxiv.org/abs/1711.10337

Madeleine, S. (2021). *Présentation du réseau de neurones convolutifs appliqué aux images médicales*. Repéré à https://www.imaios.com/fr/ressources/blog/classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-cnn

Mao, J., Wang, X. & Aizawa, K. (2023). Guided image synthesis via initial image editing in diffusion model. Dans *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5321–5329.

Marra, F., Gragnaniello, D., Verdoliva, L. & Poggi, G. (2019). Do GANs Leave Artificial Fingerprints ? *Proceedings - 2nd International Conference on Multimedia Information Processing and Retrieval, MIPR 2019*, pp. 506–511.

Nantomah, K. (2019). On some properties of the sigmoid function. *Asia Mathematika*.

Ojha, U., Li, Y. & Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models. Dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24480–24489.

PRADEEP, P., Karim, M. R. & MOHIT, S. (2018). *Practical Convolutional Neural Networks*. Repéré à https://learning.oreilly.com/library/view/practical-convolutional-neural

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. *et al.* (2018). Improving language understanding by generative pre-training.

Raghuram, S., Bharadwaj, A. S., Deepika, S., Khadabadi, M. S. & Jayaprakash, A. (2022). Digital implementation of the softmax activation function and the inverse softmax function. Dans *2022 4th International Conference on Circuits, Control, Communication and Computing (I4C)*, pp. 64–67. IEEE.

Raj, S., Mathew, J. & Mondal, A. (2024). Generalized and robust model for GAN-generated image detection. *Pattern Recognition Letters*, *182*, 104–110.

Rajakumareswaran, V., Raguvaran, S., Chandrasekar, V., Rajkumar, S. & Arun, V. (2024). DeepFake detection using transfer learning-based Xception model. *Advanced Information Systems*, *8*(2), 89–98.

Richards, M. A., Varshini, E. K., Diviya, N., Prakash, P., Kasthuri, P. & Sasithradevi, A. (2023). Deep Fake Face Detection using Convolutional Neural Networks. *12th IEEE International Conference on Advanced Computing, ICoAC 2023*.

Sharma, R., Jawade, B., Agarwal, A., Setlur, S. & Ratha, N. (2023). Attention Guided Multi-attribute Architecture For Deepfake Detection. pp. 1–4. doi: 10.1109/WNYISPW60588.2023.10349650

TOURE, S. B. (2021). Master en Statistique et Informatique Décisionnelle Implémentation d ' un algorithme de Deep Learning pour la détection automatique du port de masque facial. *UADB*.

Turpin, A. (2023). *Algorithme Perceptron, Présentation et Fonctionnement*. Repéré à https://www.jedha.co/formation-ia/algorithme-perceptron

Ünsalan, C., Höke, B. & Atmaca, E. (2024). Fundamentals of Neural Networks. Dans *Embedded Machine Learning with Microcontrollers : Applications on STM32 Development Boards* pp. 229–258. Springer.

Ura, A., Kuribayashi, M. & Funabiki, N. (2023). Study on Face Landmark-based Analysis for Synthetic Media Identification Generated by Adversarial Generative Networks. Dans *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1684–1690. IEEE.

Vajpayee, H., Yadav, N., Raj, A. & Jhingran, S. (2023). Detecting deepfake human face images using transfer learning : A comparative study. Dans *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Vol. 1, pp. 1–5. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X. & Tang, X. (2017). Residual attention network for image classification. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164.

Wang, H., Fei, J., Dai, Y., Leng, L. & Xia, Z. (2023). General GAN-generated image detection by data augmentation in fingerprint domain. Dans *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1187–1192. IEEE.

Wang, J., Zeng, K., Ma, B., Luo, X., Yin, Q., Liu, G. & Jha, S. K. (2022). GAN-generated fake face detection via two-stream CNN with PRNU in the wild. *Multimedia Tools and Applications*, *81*(29), 42527–42545.

Wang, S.-Y., Wang, O., Zhang, R., Owens, A. & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. Dans *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704.

Wang, X., Guo, H., Hu, S., Chang, M.-C. & Lyu, S. (2023). Gan-generated faces detection : A survey and new perspectives. *ECAI 2023*, pp. 2533–2542.

Wang, Y., Peng, C., Liu, D., Wang, N. & Gao, X. (2022). Forgerynir : deep face forgery and detection in near-infrared scenario. *Ieee transactions on information forensics and security*, *17*, 500–515.

Wikipédia (2021). *Réseau de neurones récurrents*. Repéré à https://fr.wikipedia.org/wiki/RÃl'seau-de-neurones-rÃl'currents

Yu, N., Davis, L. S. & Fritz, M. (2019). Attributing fake images to gans : Learning and analyzing gan fingerprints. Dans *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7556–7566.

Zhang, L., Chen, H., Hu, S., Zhu, B., Lin, C.-S., Wu, X., Hu, J. & Wang, X. (2024). X-Transfer : A Transfer Learning-Based Framework for GAN-Generated Fake Image Detection. Dans *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE.

Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Dans *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Éric Debeir (2019). *Analyse IA – CycleGAN : pierre fondatrice de la non-supervision*. Consulté le 24 avril 2024, Repéré à https://www.linkedin.com/pulse/analyse-ia-cyclegan-pierre-fondatrice-de-la-non-eric-debeir/