République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 - Guelma -
Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique
Option : Système Informatique

Thème

Détection des Fausses Informations dans les médias (sites sociaux) a base de Transfrmateur Visuel

Présenté par :
RICHI RACHA

Encadré Par :
Dr. BENCHERIET CHEMESSE ENNEHAR

Membres du Jury :
Dr.FAROU IBRAHIM
Dr.HALIMI KHALED

2022/2023

Remerciement

Tout d'abord, je remercie Dieu qui m'a aidé et m'a donné la force, la santé et la patience pour endurer toutes les épreuves pour mener à bien ce travail.

Je tiens à remercier sincèrement mon encadrant Prof. Bencheriet Chemesse Ennehar pour l'effort qu'elle a mis à mes côtés lors de la préparation de la thèse, pour ses conseils et pour toutes les facilités qu'elle m'a donnée.

J'adresse également des remerciements particuliers à tous les membres de ma famille qui m'ont soutenu à tout moment, ont prié pour moi et m'ont aidé à atteindre ce niveau.

Je remercie également tous mes amis pour leur aide et leurs réponses à mes questions, et je leur souhaite du succès dans leur vie. Je tiens également à remercier tous les professeurs du Département d'informatique de l'Université du 8 mai 1945 pour tous leurs efforts durant cinq ans.

Mes sincères remerciements vont aux membres du jury d'avoir accepté de juger mon travail.

Dédicace

Je dédie ce travail et son succès à

A mon cher père, l'homme qui m'a amené ici. Tu as toujours été pour moi un exemple de père parfait ,merci pour tout ce que vous avez donné et continuez d'offrir avec amour, attention et prières. Grâce à vous, j'ai appris le sens de l'insistance sur le succès et la responsabilité. Merci beaucoup pour tout, mes mots ne vous suffiront pas. Je prie Dieu de te protéger et de prolonger ta vie.

A ma chère mère qui m'a élevé et fait de moi ce que je suis Il n'y a pas de mots pour vous exprimer mon amour et ma gratitude pour votre soutien et vos prières pour moi, vous êtes mon bien le plus précieux.

A mes frères "ahmed akram Et ma chéré Amira " et leurs familles Je vous suis reconnaissant pour les encouragements que vous m'avez donnés et pour votre confiance et votre bonheur de ma réussite. Et à tous ceux qui m'ont aidé de près ou de loin.

Je dédie ce travail à vous tous en remerciement du soutien que vous m'avez apporté tout au long de cette période

RACHA

Résumé

Au cours des dernières années, la technologie de génération de visages synthétiques s'est développée rapidement, basée sur la technologie deepfake pour créer de fausses images et vidéos ultra-réalistes.

Dans ce mémoire de master, notre objectif est de développer un système de détection des visages falsifiés sur les images et les videos en utilisant un réseau neuronal spécifique appelé Transformateur visuel (VIT).

Cette approche d'apprentissage en profondeur nous permet d'obtenir une représentation simplifiée de nos données, sur laquelle nous distinguons les images "fausses" des images "réelles".

Utilisés à l'origine dans le traitement du langage naturel ou'ils ont prouvé leur robustesse et précision , les VIT ont par la suite été adopté dans divers domaines du traitement d'images et de la vision artificielle.

Le système proposé consiste à détecter la fausse information reçue sur des images ou des vidéos par le biais de la detection des faux visages détectés.

Les tests effectués sur les images et vidéos falsifiées ont donné des résultats encourageants, mais des améliorations peuvent être apportées en poursuivant l'apprentissage.

Mots clés :
Image falsifiée ,vidéo falsifiée ,faux visages, Apprentissage profond, Faux profond, Transformateur visuel

Abstract

In recent years, synthetic face generation technology has developed rapidly, based on deepfake technology to create ultra-realistic fake images and videos.

In this master thesis, our goal is to develop a fake face detection system using a specific neural network called Visual Transformer (VIT).

This deep learning approach allows us to obtain a simplified representation of our data, on which we distinguish "fake" images from "real" images.

Originally used in natural language processing where they proved their robustness and accuracy, VITs were later adopted in various areas of image processing and machine vision.

The proposed system consists in detecting the false information received on images or videos through the detection of false faces detected.

Testing of falsified faces has yielded encouraging results, but improvements can be made by continuing to learn.

Key words : Fake News, Fake Faces, Deep Learning, Deep Fake, vision transformer.

ملخص

في السنوات الأخيرة ، تطورت تقنية إنشاء الوجوه الاصطناعية بسرعة ، بناءً على تقنية التزييف العميق لإنشاء صور ومقاطع فيديو مزيفة واقعية للغاية.

في أطروحة الماجستير هذه ، يتمثل هدفنا في تطوير نظام للكشف عن الوجوه المزيفة على الصور ومقاطع الفيديو باستخدام شبكة عصبية محددة تسمى محول بصري .

يتيح لنا نهج التعلم العميق هذا الحصول على تمثيل مبسط لبياناتنا ، حيث نميز الصور "الخاطئة" عن الصور "الحقيقية."

تُستخدم في الأصل في معالجة اللغة الطبيعية حيث أثبتت قوتها ودقتها ، ثم تم اعتمادها لاحقًا في مجالات مختلفة من معالجة الصور ورؤية الآلة.

يتمثل النظام المقترح في الكشف عن المعلومات الخاطئة الواردة على الصور أو مقاطع الفيديو من خلال الكشف عن الوجوه الزائفة المكتشفة.

أسفر اختبار الصور ومقاطع الفيديو التي تم العبث بها عن نتائج مشجعة ، ولكن يمكن إجراء تحسينات من خلال الاستمرار في التعلم.

الكلمات الدالة : صورة مزيفة ، فيديو مزيف ، وجوه مزيفة ، تعلم عميق ، مزيف عميق ، محول بصري

Table des matières

Table des figures

Liste des tableaux

Introduction générale

La détection des fausses nouvelles(fake news) est un défi majeur dans le domaine de l'information aujourd'hui. Avec la prolifération rapide des médias sociaux et des plateformes en ligne, il est devenu de plus en plus difficile de distinguer les informations fiables des fausses. Les fausses informations peuvent avoir un impact significatif sur les individus, les communautés et même les processus démocratiques.

L'utilisation des modèles de traitement du langage naturel (NLP) a été largement explorée pour détecter les fake news en analysant le contenu textuel. Cependant, les fake news peuvent également être propagées à travers des images et des vidéos manipulées. Cela a conduit à l'émergence de nouvelles approches qui intègrent des techniques de vision par ordinateur pour détecter les fausses informations visuelles.

Les transformateurs visuels, qui sont des modèles basés sur l'architecture des transformateurs utilisés dans le domaine de la vision par ordinateur, se sont révélés très performants dans la tâche de reconnaissance et de classification d'images. Ces modèles peuvent capturer les relations spatiales et contextuelles entre les différents éléments d'une image, ce qui les rend potentiellement adaptés pour détecter les fake news visuelles.

En combinant les avantages des transformateurs visuels avec des techniques de détection des fake news, il est possible de développer des modèles qui peuvent analyser les images, détecter les manipulations et les altérations, et évaluer la crédibilité des informations visuelles. Ces modèles peuvent jouer un rôle essentiel dans la lutte contre la désinformation et contribuer à promouvoir une information plus fiable et authentique.

Par conséquent, avec cet objectif principal, nous avons développé un système visant à détecter les faux visages dans les images et les vidéos basé sur un type spécifique de réseau de neurones appelé transformateur visuel(VIT). Cette approche d'apprentissage profond nous permet d'obtenir une représentation simple des données, sur laquelle nous utilisons ensuite la densité du noyau et une estimation de l'erreur de reconstruction pour faire la distinction entre les "faux" et les "vrais" visages.

Notre travail est organisé en 4 chapitres principales :

Chapitre 01 : Fausses nouvelles (Fake News) Dans ce chapitre nous avons abordé les diffirents types de fausses nouvelles(fausses informations) et nous avons discuté quelques travaux basés sur les CNN et sur lesVIT.

Chapitre 02 : Réseau convolutifs et transformateurs de visions Dans ce chapitre nous avons présenté une étude détaillée sur les réseau à convolution (CNN) et sur les transformateur de vision (VIT) ,suivi d'une comparaison entre les deux modèles

CNN et VIT.

Chapitre 03 : Conception L'architecture détaillée du système à été développé dans ce chapitre.

Chapitre 04 :Implémentation ce chapitre a été consacré à l'implémentation des tests et des résultats obtenus, ainsi que leurs interprétation.

Nous terminons ce travail avec une conclution générale et quelques point de vue pour des travaux futures.

Chapitre 1

Fausses nouvelles (Fake News)

1 Introduction

Les "fake news", ou "fausses nouvelles" en français, font référence à des informations ou des histoires qui sont délibérément fausses ou trompeuses, présentées comme si elles étaient véridiques. Ces fausses nouvelles peuvent être créées et diffusées intentionnellement pour influencer l'opinion publique, discréditer une personne ou une organisation, ou générer des clics et des vues sur des sites web.

Les fake news sont devenues un problème majeur dans le monde entier, car elles peuvent causer des dommages considérables à la société, notamment en alimentant la confusion, la méfiance, la désinformation et la polarisation. Les plateformes de médias sociaux ont également été critiquées pour leur rôle dans la diffusion de ces fausses nouvelles à grande échelle.

Il est donc essentiel de savoir repérer et éviter les fake news afin de préserver l'intégrité de l'information et de lutter contre la désinformation et la manipulation.

2 Definition Des Faux(Fake)

2.1 Définition 01[1]

Qui est contraire à ce qui est vrai, qui comporte une erreur, qui manque de justesse, de logique, ou qui n'est pas justifié par les faits.

2.2 Définition 02[2]

Qui n'est qu'une imitation, qui n'est pas original, naturel ou authentique (avec ou sans intention frauduleuse)

3 Définition Des nouvelles(news)

3.1 Définition 01[W2]

une émission de télévision ou de radio composée de reportages sur des événements récents.

3.2 notre Définition

Tout les information publiés dans les diffirent outil de communication (TV, réseaux sociaux, les journaux,...).

4 Définition Des fausses nouvelles (fakes news)

Il existe plusieurs définitions :

4.1 Définition1[3]

Les fausses nouvelles sont largement définies comme des articles de presse qui sont intentionnellement et de manière vérifiable faux et qui pourraient induire les consommateurs en erreur.

4.2 Définition2[4]

Le terme « fake news » est composé des deux mots anglais « fake » et « news », qui signifient « fausses informations ». Ces nouvelles, qui trompent délibérément l'opinion, essaient d'attirer l'attention avec quelque chose de soi-disant « authentique », de choquer ou d'influencer l'opinion des autres. Les fake news sont écrites par des individus et par des groupes agissant dans leur propre intérêt ou au nom d'autres personnes. La création de cette désinformation est principalement due à des motifs personnels, politiques ou économiques.

4.3 Definition 03[W1]

Les fausses nouvelles sont les informations erronées sous forme d'articles, de photos ou de vidéos déguisées en « vraies informations » et visant à manipuler les opinions sont appelées fakes news. Leur diffusion est assurée par les utilisateurs des réseaux sociaux et par des robots sociaux déployés secrètement qui commentent, republient et retweetent les contenus critiques.

5 Types de fausses informations

5.1 Misinformation et Désinformation

Definition 01 [W3]

On considère que la désinformation est un phénomène courant. Par exemple, imaginons qu'une soirée commence à 20h, mais que vous oubliez ou lisez mal l'invitation et que vous indiquez à vos amis que la soirée commence à 21h. Dans ce cas, vous transmettez une information erronée, mais cela ne vous met pas en cause de manière intentionnelle. Il s'agit simplement d'une erreur involontaire, car vous n'aviez pas l'intention de tromper ou de manipuler les autres, et vous pensiez peut-être même que l'information était correcte. Il n'y a donc pas lieu de prendre des mesures répressives à votre encontre.

Definition 02 [58]

La désinformation est un ensemble de techniques de communication visant à tromper des personnes ou l'opinion publique pour protéger des intérêts (privés ou non) ou influencer l'opinion publique. L'information fausse ou faussée est à la fois.

5.2 Malinformation

Definition 01 [59]

La malinformation est une information vraie et factuelle, mais elle est intentionnellement transmise afin d'infliger un préjudice réel ou de causer une menace imminente de préjudice à une personne, une organisation ou un pays.

Definition 02 [60]

la malinformation, il s'agit du fait de diffuser de l'information qui repose sur un fait, mais qui est souvent exagérée de façon à tromper ou même à causer des préjudices.

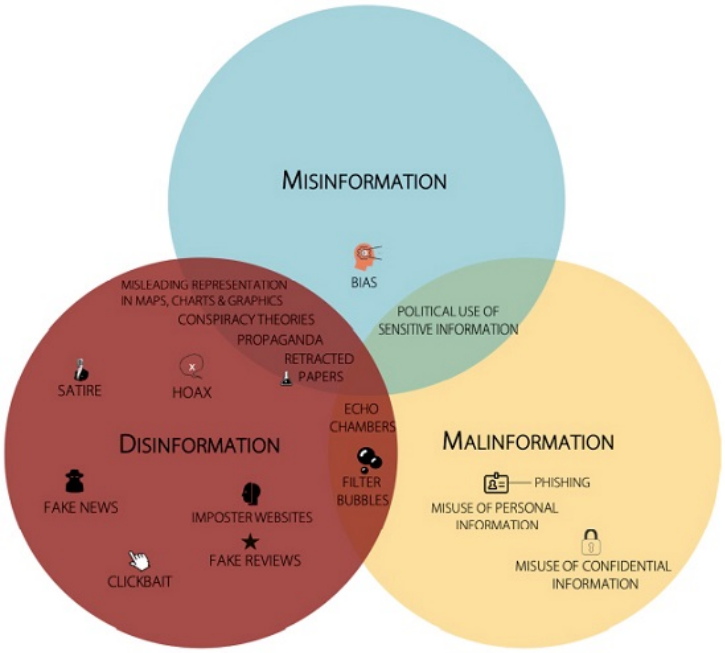


FIGURE 1.1 – Types de fausses informations[38]

6 Types De Fausses nouvelles

6.1 Audio Deepfake [5]

C'est le processus d'emprunt d'identité sonsore en manipulant les fréquences du son d'origine en ajoutant des effets externes pour obtenir une sortie aussi proche que possible du son souhaité.

6.2 fausses images

Toutes les manipulations appliquées sur une image source pour produire une autre image differente a travers une image cible (les mix entre les diffirentes caractiristiques de l'image source avec le cadrage de l'image cible).

FIGURE 1.2 – Exemple d'une fausses image

6.3 Fausses videos

C'est la manipulation des différentes caractéristiques des videos par temps(modification sous image, son). Dans la mauvaise vidéo, l'image d'une personne est combinée avec la voix d'une autre personne qui parle, une fusion complète sans montrer les détails de l'orateur.

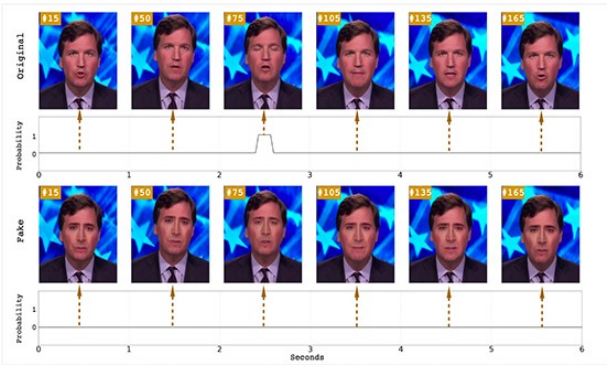


FIGURE 1.3 – Exemple de video Fake[39]

7 Méthodes utilisées pour la génération des fausses nouvelles

Les générateurs automatiques de fausses nouvelles (ou "fake news") utilisent souvent des algorithmes de machine learning pour créer des textes qui ressemblent à des articles de presse authentiques. Les deux méthodes les plus couramment utilisées pour la génération de ces textes sont les réseaux de neurones générateurs adversaires (GAN) et les auto-encodeurs.

7.1 Les Generatives Adversarials Networks (GANs) [6]

Une nouvelle méthode d'apprentissage semi-supervisé et non supervisé est appelée réseaux antagonistes génératifs (GAN). Ils réussissent à simuler implicitement la distribution des données en grande dimension. Ils peuvent être reconnus par le développement d'une paire de réseaux qui se font concurrence. Une analogie courante qui s'applique aux données visuelles consiste à considérer un réseau comme un faux artiste et l'autre comme un expert en art. Le faussaire, appelé aussi le générateur dans la littérature d'armes, invente des contradictions pour marquer des images plausibles. Le discriminateur, ou expert, reçoit à la fois des photos fausses et réelles et tente de les différencier. Les deux sont engagés simultanément et se font concurrence.

1. **Générateur (Generator)**

   Afin de produire de fausses données qui seront données au discriminateur lors de son apprentissage, le générateur est un réseau de neurones. Il accepte un vecteur arbitraire avec une taille fixe en entrée et en sortie des données générées. Le but principal du générateur est de faire classer par le discriminateur les echantillons de la fausse base de données qu'il produit en echantillons de la vraie base de données. Par conséquent, lorsque D(G(z)) est proche de 1, le générateur a atteint son but. Par conséquent, le but est de maximiser D(G(z)), ce qui revient à minimiser $D(G(z)) - 1$.

2. **Discriminateur(Discriminator)**

   Le discriminateur est un réseau de neurones qui catégorise les données qui lui sont entrées afin de faire la distinction entre les données réelles issues de la base de données réelles et les données générées.

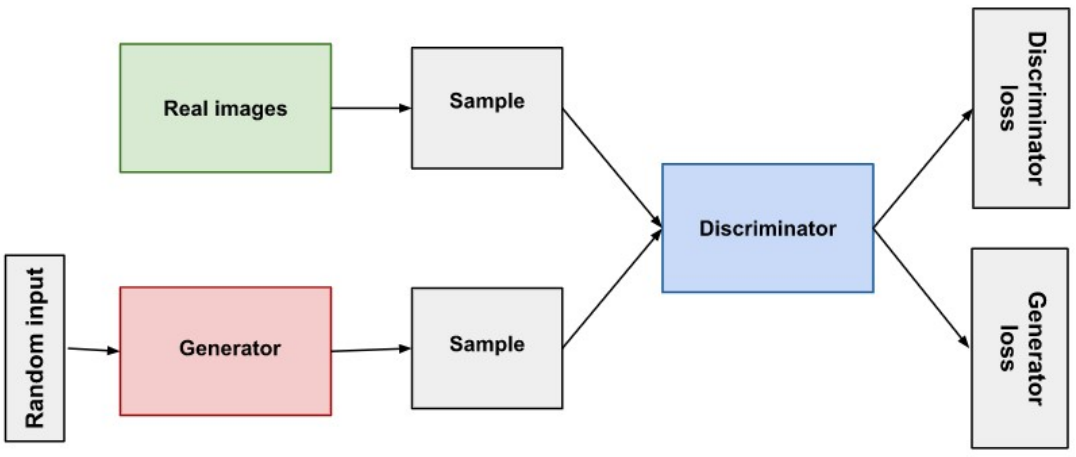


FIGURE 1.4 – Architecture du reseau GAN [40]

7.2 Les auto-encodeurs [7]

La possibilité de créer une nouvelle représentation d'un ensemble de données est rendue possible par les auto-encodeurs, qui sont des algorithmes d'apprentissage non supervisés

basés sur des réseaux de neurones artificiels. De manière générale, celui-ci est plus court et contient moins de descripteurs.
Permettant de réduire la dimension du jeu de données. L'étape suivante consiste à combiner l'encodeur de la première image avec le décodeur de la seconde pour créer une troisième fausse image à partir de deux autres images. Deux composants composent une auto-architecture :l'encodeur et le décodeur

1. **Encodeur**

   l'encodeur sert à compresser le jeu de données d'entrée en une représentation plus petite. À cette fin, il extrait les caractéristiques (features) les plus importantes à partir de données initiales. Cela aboutit à une formation compacte appelée bottleneck aussi connu comme l'espace latent.

2. **Decodeur**

   À l'inverse de l'encodeur, le décodeur décompresse le bottleneck pour reconstituer les données. Son défi consiste à utiliser les caractéristiques contenues dans le vecteur condensé pour essayer de reconstruire le plus fidèlement possible le jeu de données.

3. **L'espace latent**

   L'espace latent correspond donc aux données compressées, autrement dit à l'espace entre l'encodeur et le décodeur. Le but de la création de cet espace latent est de limiter le flux d'information entre les deux composants de l'auto-encodeur. Cette limitation se traduit par la suppression du bruit afin de ne laisser passer que les informations importantes.

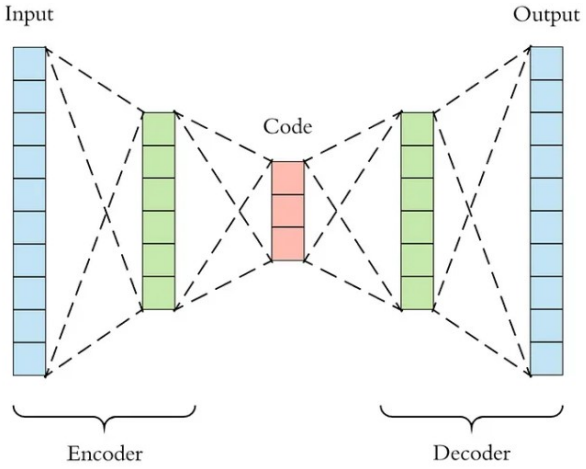


FIGURE 1.5 – Architecture de l'auto-encodeur [41]

8 Bases de donné

8.1 Celeb Df v2 Dataset [8]

L'ensemble de données Celeb-DF est composé de 590 vidéos réelles et de 5 639 vidéos DeepFake (correspondant à plus de deux millions d'images vidéo). La durée moyenne de toutes les vidéos est d'environ 13 secondes avec la fréquence d'images standard de 30 images par seconde. Les vraies vidéos sont choisies parmi des vidéos YouTube accessibles au public, correspondant à des interviews de 59 célébrités avec une distribution diversifiée dans leurs sexes, âges et groupes ethniques.
56,8% des sujets dans les vidéos réelles sont des hommes et 43,2 % sont des femmes.
8,5% ont 60 ans et plus,30,5% ont entre 50 et 60 ans, 26,6% ont la quarantaine, 28,0% ont la trentaine, et 6,4 % ont moins de 30 ans.
5,1% sont asiatiques, 6,8% sont des Afro-Américains et 88,1 % sont des Caucasiens.
De plus, les vidéos réelles présentent une large gamme de changements dans des aspects tels que la taille des visages des sujets (en pixels), les orientations, les conditions d'éclairage et les arrière-plans. Les vidéos DeepFake sont générées en échangeant des visages pour chaque paire des 59 sujets. Les vidéos finales sont au format MPEG4.0.

FIGURE 1.6 – Base de Donnée Celep DF [8]

8.2 FaceForensics [9]

FaceForensics est un ensemble de données vidéo composé de plus de 500 000 images contenant des visages de 1004 vidéos qui peuvent être utilisées pour étudier les falsifications d'images ou de vidéos. Pour créer ces vidéos, ils ont utilisé une version automatisée de l'approche de pointe Face2Face. Toutes les vidéos sont téléchargées depuis Youtube, ils ont choisi des vidéos avec une résolution supérieure à 480p qui ont été étiquetées avec "visage", "présentateur de nouvelles" ou "programme d'actualités" et sont réduites à de courts clips continus qui contiennent principalement des visages frontaux. En particulier, ils proposent deux versions de jeu de données : Source-to-Target : où ils reconstituent plus de 1000 vidéos avec de nouvelles expressions faciales extraites d'autres vidéos, qui par ex. peut être utilisé pour entraîner un classificateur à détecter de fausses images ou vidéos. Auto-reconstitution : où ils utilisent Face2Face pour reconstituer les expressions faciales des vidéos avec leurs propres expressions faciales comme entrée pour obtenir des paires de vidéos, qui par ex. peut être utilisé pour entraîner des modèles de raffinement génératif

supervisé.

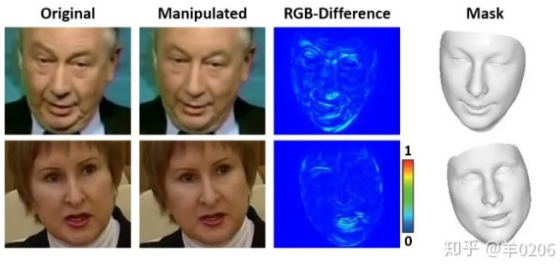


FIGURE 1.7 – Base de donneé FaceForensics [42]

8.3 FaceForensics++ [10]

Composé de 1000 vidéos originales manipulées avec des techniques de première et deuxième génération. En particulier, il existe quatre types de manipulation, deux approches basées sur l'infographie (Face2Face et FaceSwap) et deux approches basées sur l'apprentissage (DeepFakes et NeuralTextures). Au total, l'ensemble de données contient 4000 vidéos deepfake.

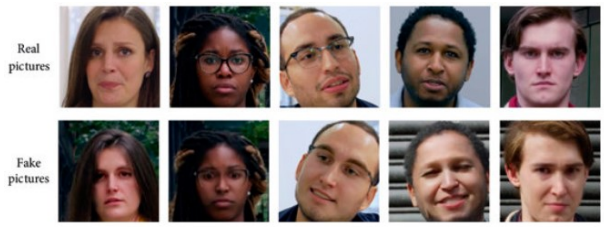


FIGURE 1.8 – Base de donneé The FaceForensics++ [43]

8.4 Face Synthesis[11]

Le jeu de données Face Synthetics est une collection de diverses images de visages synthétiques avec des étiquettes de vérité terrain.le jeu de données contient :

- 100 000 images de visages à une résolution de 512 x 512 pixels ;
- 70 annotations standard de repères faciaux ;
- Annotations de classe sémantique par pixel : Les images sont également accompagnées d'annotations de classe sémantique par pixel. Cela signifie que chaque pixel de l'image est étiqueté avec une classe sémantique correspondante. Par exemple, différents éléments du visage (yeux, nez, bouche) peuvent être étiquetés individuellement, ainsi que d'autres régions du visage et du fond ;

Il peut être utilisé pour former des systèmes d'apprentissage automatique pour des tâches liées au visage telles que la localisation de points de repère et l'analyse de visage, montrant que les données synthétiques peuvent à la fois correspondre aux données réelles en termes de précision et ouvrir de nouvelles approches où l'étiquetage manuel serait impossible.

Certaines images incluent également des faces distractrices décentrées en plus des faces primaires centrées dans l'image.
L'ensemble de données Face Synthetics peut être utilisé pour des recherches non commerciales .

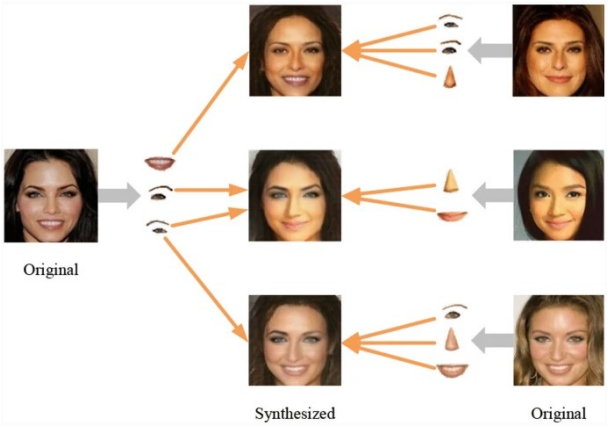


FIGURE 1.9 – Base de donneé Face Synthesis [44]

8.5 The DeepFake Detection Challenge (DFDC) Dataset [12]

C'est une ensemble de données unique,créer et partager pour le défi composé de plus de 100 000 vidéos. Le DFDC a permis à des experts du monde entier de se réunir, de comparer leurs modèles de détection de deepfake, d'essayer de nouvelles approches et d'apprendre du travail de chacun. Le jeu de données DFDC se compose de deux versions :

1. **Base de donnée"Dataset preview"**
   - 5k vidéos
   - Doté de deux algorithmes de modification faciale

   Base de donnée"Complete dataset"

2. - 124k vidéos
   - Doté de huit algorithmes de modification faciale

Cet ensemble de données complet a été utilisé pour créer de nouveaux et meilleurs modèles pour détecter les médias manipulés. L'ensemble de données a été créé par Facebook avec des acteurs qui ont conclu un accord pour l'utilisation et la manipulation de leurs ressemblances dans la création de l'ensemble de données.

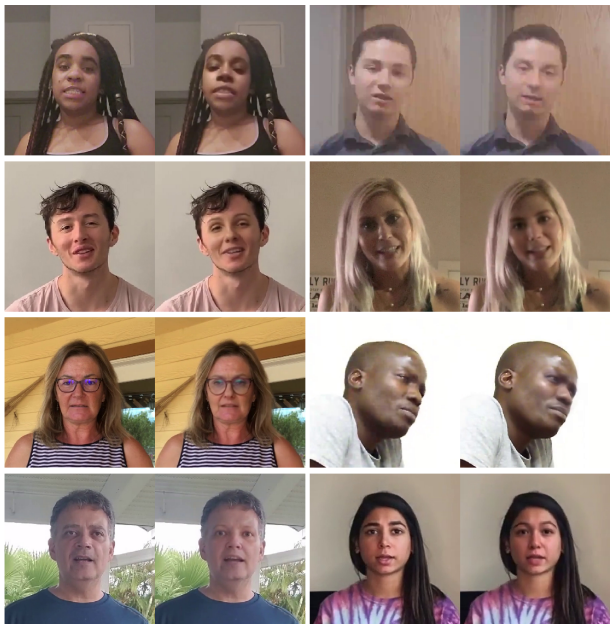


FIGURE 1.10 – Base de donneé The DeepFake Detection Challenge (DFDC)[45]

8.6 Face2Face Video Dataset [13]

Cet ensemble de données se compose des 300 vidéos originales modifiées à l'aide des méthodes de manipulation Face2Face. L'ensemble de données est divisé en 10 blocs de 30 vidéos chacun. La manipulation Face2Face consiste à transférer les expressions faciales d'un acteur source vers un acteur cible tout en préservant l'identité de l'acteur cible.

FIGURE 1.11 – Base de donneé Face2Face Video [46]

9 Travaux connexes

9.1 ARTICLE 01 : 'Deepfake Video Detection Using Convolutional Vision Transformer' [14] 'Détection vidéo Deepfake à l'aide d'un transformateur de vision convolutif '

Dans cet article les chercheurs se sont concentrés sur detection des faux visages (générer des visages, échanger des visages entre deux sujets dans une vidéo, modifier les expressions faciales, changer de sexe et modifier les traits du visage) pour minimisé leur danger dans

notre vie, ils ont développés un modèle généralisé nommé Convolutional Vison Transformer.Ils proposent une Convolution de transformateur visuel pour la détection de deep fakes dans les videos. utilisées les transformateurs visuels pour classifier les caractéristique d'image extraier par CNN.
Ce modèle a été testé sur 400 vidéos de la base de données Deep Fake Detection Challenge (DFDC) où ils ont obtenu une précision de 91,5% dans les tests, ce qui est une bonne valeur, mais pas avec d'autres bases de données où il chute à 69%.

Méthode	validation	test
CNN and RNN-GRU	92.61%	91.88%
CViT	87.25%	91.5%

TABLE 1.1 – Resultats de test de model CVit pour DFDC dataset

Méthod	validation	FaceSwap	Face2Face
MesoNet	92.61%	91.88%	92%
MesoInception	82.4%	98%	93.33%
CVit	93.75%	69%	69.39%

TABLE 1.2 – Resultats de test de model CVit pour base de donner FaceForensics++

9.2 ARTICLE 02 : Generalization Of Audio Deepfake Detection [5] : Généralisation de la détection de deep fake audio

Dans cet article, les chercheurs s'interessent à détecter les faux sons ou ce qu'on appelle l'usurpation d'identité phonétique, qui représente un grand danger dans notre vie quotidienne. Ils proposent un modèle qui utilise la technique du cosinus de perte à grande marge (LMCL) pour forcer le réseau de neurones à apprendre des fonctionnalités qui peuvent augmenter la variance inter-couches et réduire la variance intra-couche, et augmenter le masquage de fréquence en ligne pour forcer le réseau de neurones à apprendre plus de combinaisons de fonctionnalités. Pour que le modèle puisse faire la distinction entre la vraie et la fausse voix dans différents environnements.
Ce systeme a été entrainé sur la base de données "ASVspoof 2019 train set" [61] qui contient totalement 25,380 Énoncés et "T1 + Augmented train set" [61] qui contient totalement 152,280 Énoncés et "T1 + Augmented train set et logically-replayed train set" [61] qui contient totalement 177,660 Énoncés.
Le system a été tester sur un ensemble différent de la base de données "ASVspoof2019 challenge" donnent un EER (le taux d'erreur égal) de 4.04% et un t-DCF(la fonction de coût de détection en tandem) de 0.109% comme resultats préliminaire et a près l'ajout de la couche de masquage de fréquence l'erreur EER a été réduit a 1.81%. telque :

- E1 :the official evaluation set of the 'ASVspoof 2019 LA challenge'
- E2 :the official evaluation set of the 'Augmented ASVspoof 2019'
- E3 :the official evaluation set 'Logically-Replayed ASVspoof 2019'
- T1 est le protocole officiel du 'ASVspoof 2019 LA challenge'
- T2 est le protocole officiel du 'Augmented ASVspoof 2019'
- T3 est le protocole officiel du 'Logically-Replayed ASVspoof 2019'

	Benchmaks		
Training Protocols	E1	E2	E3
T1	1.81%	20.43%	8.70%
T2	1.64%	5.34%	8.21%
T3	1.26%	5.32%	2.62%

TABLE 1.3 – Resultats de test du model proposé

9.3 ARTICLE 03 :Deepfake detection in digital media forensics [15] Détection de deepfake dans la criminalistique des médias numériques

Les chercheurs dans cet article se sent concentré sur la détection des fausses vidéos créer par générateur de deepfake(gan, auto-encodeur) diffusées sur les médias sociaux, pour minimiser leur danger. Ils proposent une méthode pour détecter les faux basée sur l'utilisation des reseau CNN (Réseau de neurones convolutifs) pour la détection des différentes caractiristiques au niveau de trame, et Long Short-Term Memory (LSTM) utilisé pour la comparison entre les différentes images d'une vidéo.
Les chercheurs ont formé leur modèle sur la base de données Celab DF, en mettant l'accent sur la précision et la perte. On note que la perte de validation est supérieure à la perte d'apprentissage, ce qui signifie un bon apprentissage pour le modèle.
La précision obtenue est 91% qui est supérieur au autres modeles existants

Methods	Celeb-Df
Proposed Modèle	88.8%
MesoInception-4	53.5%
Xception-c40	65.5%
Xception-raw	48.3%
Xception-c30	65.3%
Two-stream	53.8%
Meso-4	54.8 %

TABLE 1.4 – Comparaison du modele proposé avec les modeles existants

9.4 ARTICLE 04 :Combining EfficientNet and Vision Transformers for Video Deepfake Detection [16] Combinaison de transformateurs nets et de vision efficaces pour la détection de deepfake vidéo :

Dans cet article, les chercheurs se sont concentrés sur la recherche des faux visages dans les fausses vidéos, ils ont proposé deux modèles pour la detection des faux visages dans une vidéo :
Le ViT efficace :ils utilisent les reseaux convolutifs pré-entrainé 'EfficientNet B0' pour extraire les caractiristiques visuelles au format d'image de 7*7 pixel et ensuite utilisent un encodeur de transformateur, dans une configuration très similaire au transformateur de vision (VIT) pour obtenir une description globale informative pour la tâche.
Convolutional Cross ViT, est une evolution de ViT efficace, il s'appuie à la fois sur le VIT efficace et l'architecture transformateur multi-échelle. Qui fonctionne à deux échelles différentes pour capturer les détails locaux et globaux. Le premier pour traiter des patchs

plus petits, et l'autre branche travaille sur des patchs plus grands pour avoir un champ réceptif plus large.
Des tests ont été réalisés sur les bases de données Faceforensics++ et DFDC. Bien que les résultats obtenus soient précis, le modèle CVIT fournit des résultats plus optimaux que ce du modèle pour les deepfakes dans les bases de données Faceforensics++, Selim EfficientNet B7 et ViT with distillation.

Model	AUC	F1-score	# params
ViT with distillation	0.978	91.9%	373M
Selim EfficientNet B7	0.972	90.6%	462M
Convolutional ViT	0.843	77.0%	89M
Efficient ViT (our)	0.919	83.8%	109M
Conv. Cross ViT Wodajo CNN (our)	0.925	84.5%	142M
Conv. Cross ViT Eff.Net B0 - Avg (our)	0.947	85.6%	101M
Conv. Cross ViT Eff.Net B0 - Voting (our)	0.951	88.0%	101M

TABLE 1.5 – Resultats de tests du model Conv. Cross ViT

Model	Mean	FaceSwap	DeepFakes	FaceShifter	NeuralTextures
Convolutional ViT	67%	69%	93%	46%	60%
Efficient ViT (our)	76%	78%	83%	76%	68%
Conv. Cross ViT Wodajo CNN (our)	76%	81%	83%	73%	67%
Conv. Cross ViT Eff.Net B0 (our)	80%	84%	87%	80%	69%

TABLE 1.6 – Resultants de tests du modele Conv. Cross ViT sur différentes datasets

10 conclusion

Les fausses nouvelles constituent un grave problème pour la société et la démocratie car elles conduisent à la désinformation, à la manipulation et à la polarisation. Les plateformes de médias sociaux peuvent jouer un rôle important dans la lutte contre les fausses nouvelles en mettant en place des politiques qui limitent la diffusion de fausses informations et vérifient l'origine du matériel.
Pour éviter d'être dupé par de fausses nouvelles, les sources doivent être vérifiées, des informations complémentaires doivent être recherchées et des sources fiables doivent être consultées.
La lutte contre les fausses nouvelles est un effort d'équipe qui implique la coopération des médias, des gouvernements, des plateformes de médias sociaux et du grand public. Nous pouvons travailler ensemble pour promouvoir l'exactitude et l'intégrité de l'information et contribuer ainsi à bâtir une société plus juste et informée.

Chapitre 2

Réseau convolutifs et transformateurs de visions

1 Introduction

Les réseaux de neurones sont devenus des outils clés, en particulier pour la reconnaissance d'objets et la catégorisation d'objets dans les images. Les deux types de réseaux neuronaux les plus populaires pour la vision par ordinateur sont les réseaux convolutifs (CNN) et les transformateurs de vision (ViT).
Les CNN sont des réseaux de neurones spécialisés dans le traitement des données spatiales, dans les images. Ils utilisent des couches alambiquées pour extraire les caractéristiques de l'image en examinant des zones désignées de l'image, appelées filtres. En termes de classificsation et de localisation d'objets sur les photos, CNN sont trés efficace.
Une technique plus récente de vision par ordinateur appelée "transformers for vision" (ViT) utilise des couches transformatrices similaires à celles utilisées pour traiter le langage naturel. Le ViT prend des photos en entrée, les divise en patchs, puis les traite comme des séquences vectorielles. Utilisant moins de paramètres que le CNN, cette méthode a produit des résultats impressionnants dans la catégorisation des images.

2 Définition de l'apprentissage

L'apprentissage a été defini de différentes manières dont voici les plus importantes :

2.1 Définition 01[17]

Beillerot (1989) définit l'apprentissage comme un processus qui permet à l'apprenant de créer des savoirs pour pouvoir penser et agir. C'est à dire, c'est un processus qui permet à celui qui apprend de créer à l'intérieur de lui-même des savoirs pour penser et agir.

2.2 Définition 02[17]

Pour Delevay (1992), le processus d'apprentissage est quelque peu différent. En ce sens, l'apprentissage désigne un processus qui permet à l'individu de se transformer psy-

chiquement. Donc, l'objet de ce processus est l'information. En d'autres termes, c'est une transformation qui s'effectue lorsque celui qui apprend entre en contact avec des objets qui lui sont extérieurs.

2.3 Définition 03[17]

Legendre (1993) définit l'apprentissage comme un acte de perception, d'interaction et d'intégration d'un objet par un sujet. Acquisition des connaissances et développement d'habiletés, d'attitudes et de valeurs qui s'ajoutent à la structure cognitive d'une personne."Processus qui permet l'évolution de la synthèse des savoirs, des habiletés, des attitudes et des valeurs d'une personne."

3 Apprentissage automatique [18]

L'apprentissage automatique, ou machine learning en anglais, est un domaine de recherche en intelligence artificielle qui vise à donner aux machines la capacité "d'apprendre" à partir de données à l'aide de modèles mathématiques. Plus précisément, il s'agit d'un processus par lequel des données importantes sont extraites d'un ensemble de données d'apprentissage.
L'objectif de cette phase est d'identifier les paramètres du modèle qui donneront les meilleurs résultats, y compris lorsque le modèle termine la tâche qui lui a été assignée. Une fois l'apprentissage terminé, le modèle peut ensuite être utilisé en production.

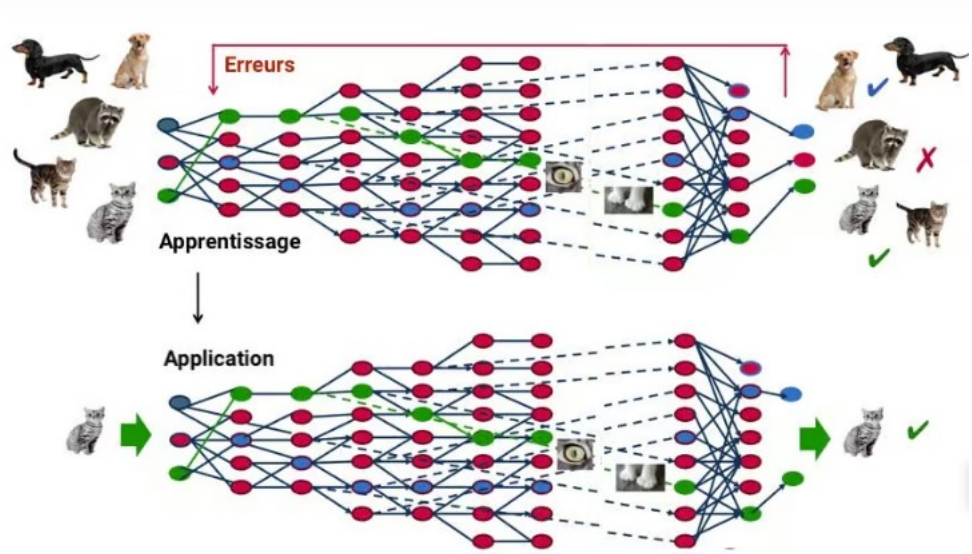


FIGURE 2.1 – schema reprisentatif de l'apprentissage automatique

4 Apprentissage en profondeur [19]

Un type d'intelligence artificielle appelé "deep learning" ou "apprentissage en profondeur" est dérivé du machine learning (apprentissage automatique), où la machine est capable d'apprendre par elle-même, par opposition à la programmation, où elle se contente

de suivre exactement des règles. car ils ont été prédéterminés. Il a permis des avancées majeures et rapides dans les domaines de l'analyse des signaux sonores ou visuels, en particulier dans les domaines du traitement automatisé du langage, de la vision par ordinateur, de l'identification faciale et de la reconnaissance vocale.

5 Types d'apprentissages

5.1 Apprentissage supervisé[20]

L'apprentissage supervisé : Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante).
Un expert (ou oracle) doit préalablement correctement étiqueter des exemples. L' « apprenant » peut alors trouver ou approximer la fonction qui permet d'affecter la bonne « étiquette » à ces exemples. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées (on parle alors d'apprentissage supervisé probabiliste).
Le but d'apprentissage supervisé est de prédire la valeur de sortie pour de nouvelles données.

5.2 Apprentissage non supervisé [18]

L'apprentissage non supervisé est une branche du machine learning, caractérisée par l'analyse et le regroupement de données non-étiquetées. Pour cela, ces algorithmes apprennent à trouver des schémas ou des groupes dans les données, avec très peu d'intervention humaine. En termes mathématiques, l'apprentissage non supervisé implique l'observation de plusieures occurrences d'un vecteur X et l'apprentissage de la probabilité de distribution p(X) pour ces occurrences.
Le but de l'apprentissage non supervisé est d'obtenir un aperçu à partir d'une grande quantité de donnée.

5.3 Apprentissage semi supervisé [21]

L'apprentissage » semi-supervisé « se situe entre les deux et offre un compromis entre apprentissage supervisé et non-supervisé. Pendant l'entraînement, un ensemble de données étiqueté de moindre envergure est utilisé pour guider la classification et l'extraction de caractéristiques à partir d'un ensemble plus large de données non étiquetées.
Cette approche s'avère utile dans les situations où le nombre de données étiquetées est insuffisant pour l'entraînement d'un algorithme supervisé. Elle permet de contourner le problème.

5.4 Apprentissage par renforcement

L'apprentissage par renforcement consiste à laisser un algorithme apprendre de ses erreurs pour atteindre un objectif. L'algorithme essayera de nombreuses approches différentes pour tenter d'atteindre son but.

En fonction de ses performances, il sera récompensé ou pénalisé pour l'inciter à poursuivre dans une voie ou à changer d'approche. Cette technique est notamment utilisée pour permettre à une IA de surpasser les humains dans les jeux.

5.5 Apprentissage par transfert[W4]

Ce type d'apprentissage applique les connaissances acquises lors de nos expériences passées lorsque nous sommes confrontés à un nouveau problème ou à une nouvelle tâche, et c'est la base de l'apprentissage par transfert. Par exemple, si nous savons faire de la bicyclette et que l'on nous demande de faire de la moto, ce que nous n'avons jamais fait auparavant, nous appliquerons toujours notre expérience de la bicyclette à la moto, notamment en ce qui concerne la direction du guidon et l'équilibre de la moto. Ce concept simple constitue la base de l'apprentissage par transfert.

6 Réseaux conventionnel(classique)[W5]

Les réseaux de neurones artificiels sont un type spécial d'algorithmes d'apprentissage automatique qui sont calqués sur le cerveau humain. Autrement dit, tout comme la façon dont les neurones de notre système nerveux sont capables d'apprendre des données passées, de même, l'ANN est capable d'apprendre des données et de fournir des réponses sous forme de prédictions ou de classifications.
Les ANN sont des modèles statistiques non linéaires qui affichent une relation complexe entre les entrées et les sorties pour découvrir un nouveau modèle. Une variété de tâches telles que la reconnaissance d'images, la reconnaissance vocale, la traduction automatique ainsi que le diagnostic médical utilisent ces réseaux de neurones artificiels.

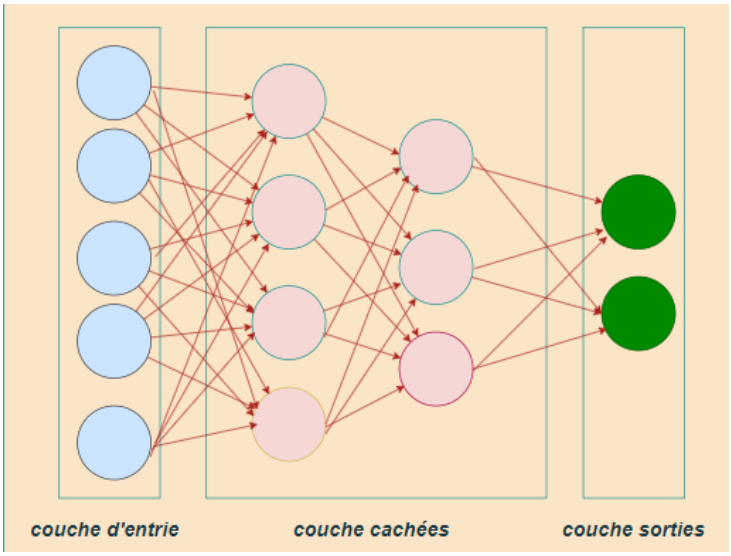


FIGURE 2.2 – Representation d'un réseau conventionel

7 Définition de réseaux convolutionnel(CNN)[22]

Un réseau de neurones convolutionnel (CNN) est un type de réseau de neurones artificiels qui est souvent utilisé pour la reconnaissance d'images et d'autres tâches de traitement de données qui impliquent des entrées structurées en forme de grille, telles que des images, des vidéos ou des signaux sonores.
Le principe de base du CNN est d'utiliser des filtres convolutifs pour extraire des caractéristiques significatives de l'image en entrée. Ces filtres sont appliqués de manière répétée sur l'image en entrée pour produire une carte de caractéristiques qui capture les motifs locaux de l'image.

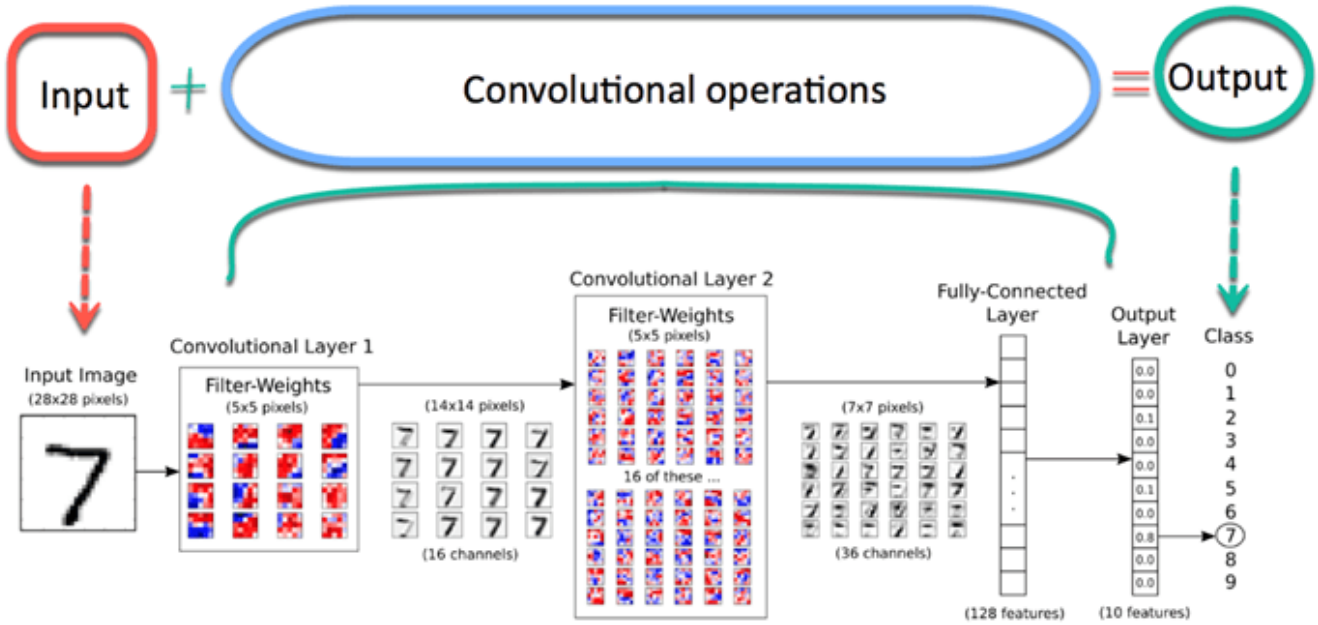


FIGURE 2.3 – Representation d'un réseau convolutionnel

8 Architecture d'un réseau convolutif :

Un réseau CNN est constituer de quatre couches principales :
— La couche d'entrée
— La couche convolutive
— La couche pooling
— La couche d'activation
— La couche entièrement connectée ou Fully Connected (FC)

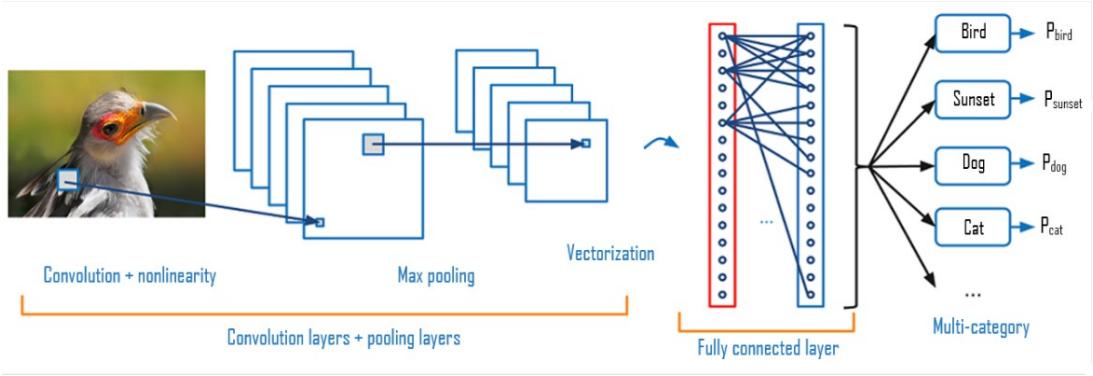


FIGURE 2.4 – Architecture générale d'un CNN [47]

8.1 Couche convolutive[W6]

La convolution, est le fait d'appliquer un filtre mathématique à une image. D'un point de vue plus technique, il s'agit de faire glisser une matrice par-dessus une image, et pour chaque pixel, utiliser la somme de la multiplication de ce pixel par la valeur de la matrice. Cette technique nous permet de trouver des parties de l'image qui pourraient nous être intéressantes.
Un réseau de neurones à convolution contient de multiples filtres et ces filtres sont appliqués sur l'image d'origine. Après la première étape nous avons donc autant de nouvelles images que de filtres.
On'a : X est une matrice filtre et Y matrice d'image l'opération de convolution est definie par :

$$\sum_{i=0}^{k}\prod_{j=0}^{k}(X[i][j]*Y[i][j]) \tag{2.1}$$

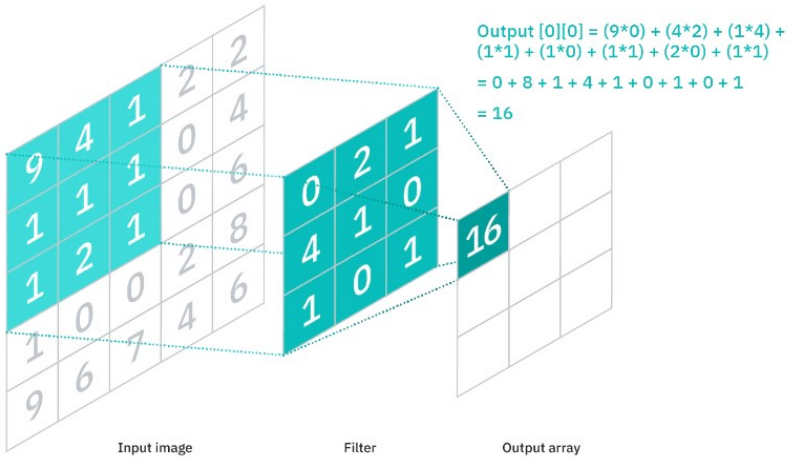


FIGURE 2.5 – Exemple de convolution [48]

Les déffirents parametres définissent le volume de la couche[W7] :

— La profondeur de la couche : c'est le nombre de noyaux de convolution voire le nombre de neurones assemblés a un même champ récepteur.
— Le pas (stride) : représente le nombre de pixel de déplacent aprés chaque opération .

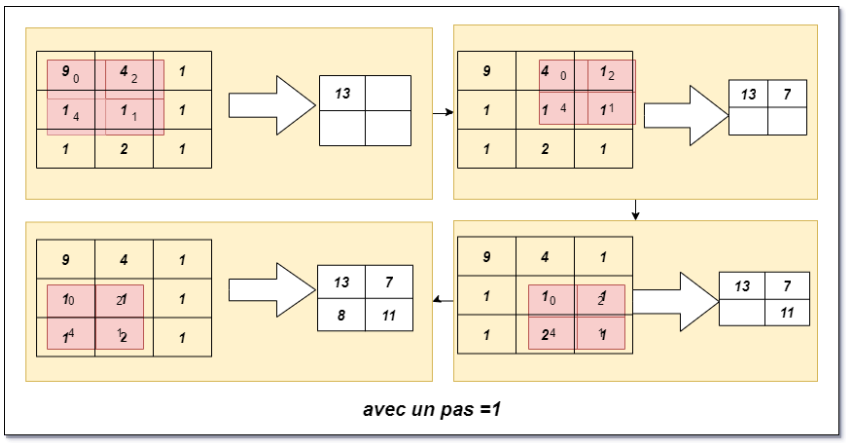


FIGURE 2.6 – Illustration de pas dans la couche convolutive

— La marge a zéro (zéro padding) : est une technique consistant à ajouter P zeros à chaque côté des frontières de l'entrée. Cette valeur peut être spécifiée soit manuellement, soit automatiquement.

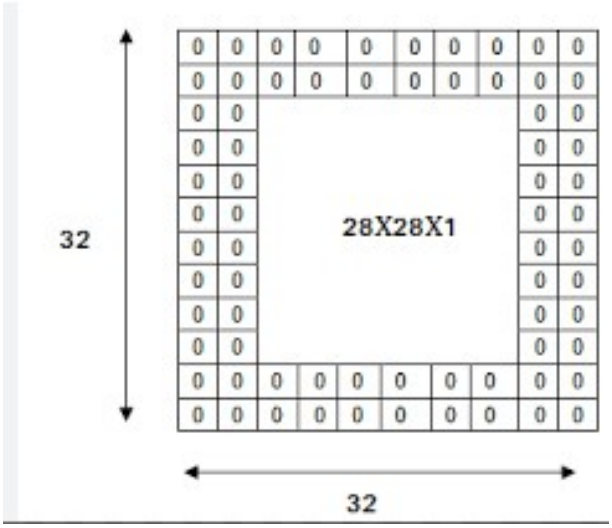


FIGURE 2.7 – Marge zéro de couche convolutive [49]

8.2 La couche pooling[19]

C'est une couche intermédiaire entre les couches convolutives utilisées pour minimiser la taille de matrice donner et éviter la perte d'information,Il existe plusieurs types de pooling mais les plus utilisés sont le pooling max et le pooling moyen.

— Pooling max : consiste à couper l'image en petites cellules de 2*2 pixels ou 3*3 pixels et choisir les valeurs max adjacentes.

— Pooling moyen : on prend la moyenne des valeurs selectionées dans le cellule.

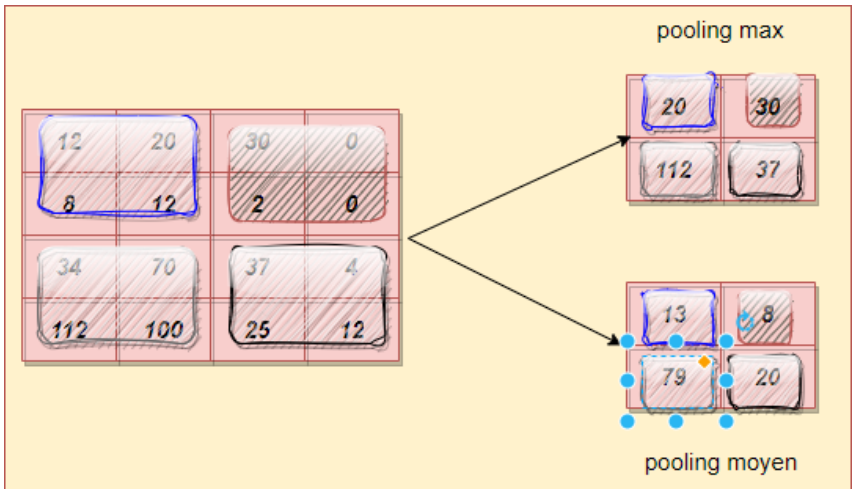


FIGURE 2.8 – Exemple de pooling max et pooling moyen de 2*2

8.3 Couche d'activation [23]

Les couches d'activation, sont des couches non linéaires qui suivent généralement les couches de convolution, et jouent le role de sélection de neurone qui va se déclencher. L'entrée de la couche d'activation est un nombre réel qui est transféré par l'application d'une fonction non linéaire.
La couche d'activation est importante car elle permet au réseau d'apprendre des mappages non linéaires pour le rendre plus robuste face à des fonctions complexes. Les couches d'activation les plus courantes utilisées dans les CNN sont sigmoïde, Tanh, ReLU, LeakyReLU et softmax.
Les couches d'activation peuvent être classées en couches d'activation saturées et en couches d'activation non saturées. Si la sortie de la couche d'activation se situe entre des limites finies, elle est alors classée comme saturée ; sinon, si elle tend vers l'infini, elle est considérée comme une fonction d'activation non saturée.

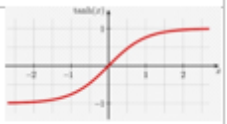
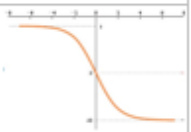
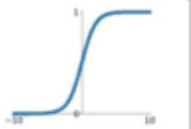
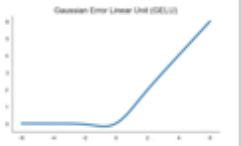
Nom de fonction	*Formule*	*Graphe*
la couche Rectified Linear Units (Relu)	relu (x)=max(0,y)	
correction par Tangente Hyperbolique (TanH)	Tanh(x)= $((e^x - e^{-x})/(e^x + e^{-x})$	
La Fonction d'activation Softmax(Softmax)	softmax(x) = exp(x) / sum(exp(x)).	
La correction par la fonction Sigmoïde	sigmoid(x) = 1 / (1 + exp(-x))	
activation des unités linéaires d'erreur gaussienne (GeLu)	gelu(x) = x. 1 2 [1 + erf (x √2/)]	

FIGURE 2.9 – Exemple de quelques fonctions d'activation

8.4 La couche entièrement connectée ou Fully Connected (FC)[W8] :

Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme fully-connected). Après avoir reçu un vecteur en entrée, la couche FC applique successivement une combinaison linéaire puis une fonction d'activation dans le but final de classifier l'image d'entrée. Elle renvoie enfin en sortie un vecteur de taille 'd' correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'image d'entrée d'appartenir à une classe.
Par exemple, s'il s'agit bien d'un problème de classification de pommes et d'orange, le vecteur final sera de taille '2' : chaque élément donne la probabilité d'appartenir soit à la classe pomme, soit à la classe orange. Ainsi, le vecteur [0.8 , 0.2] signifie que l'image à 80% de chances de représenter une pomme.

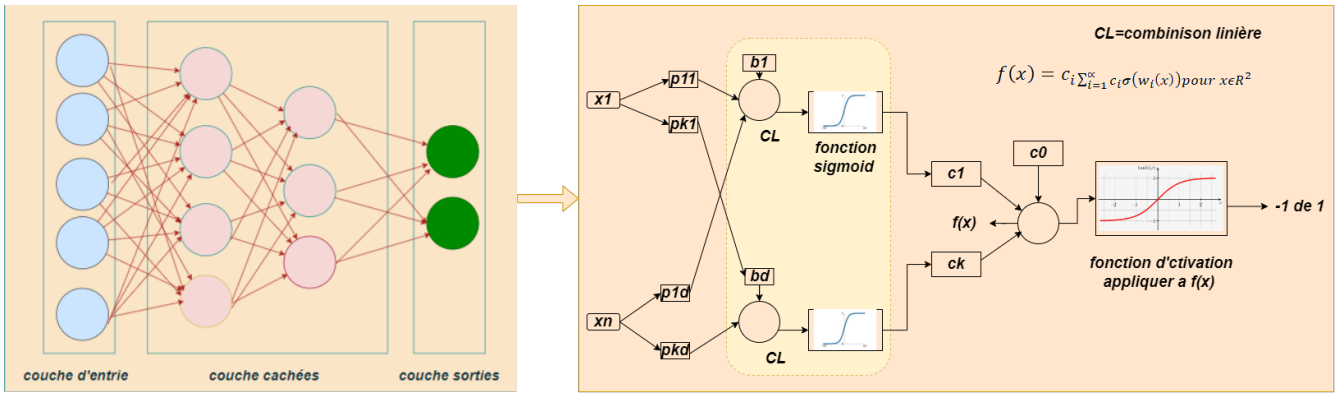


FIGURE 2.10 – Illustration de la couche entierement connectées

9 La difference entre les résaux ANN et CNN [W9]

la différence entre l'apprentissage automatique et le modèle d'apprentissage en profondeur se situe dans la zone d'extraction des fonctionnalités. L'extraction de caractéristiques est effectuée par l'homme dans l'apprentissage automatique, tandis que dans les modèles d'apprentissage en profondeur l'extraction des fonctionnalités est par eux-mêmes.

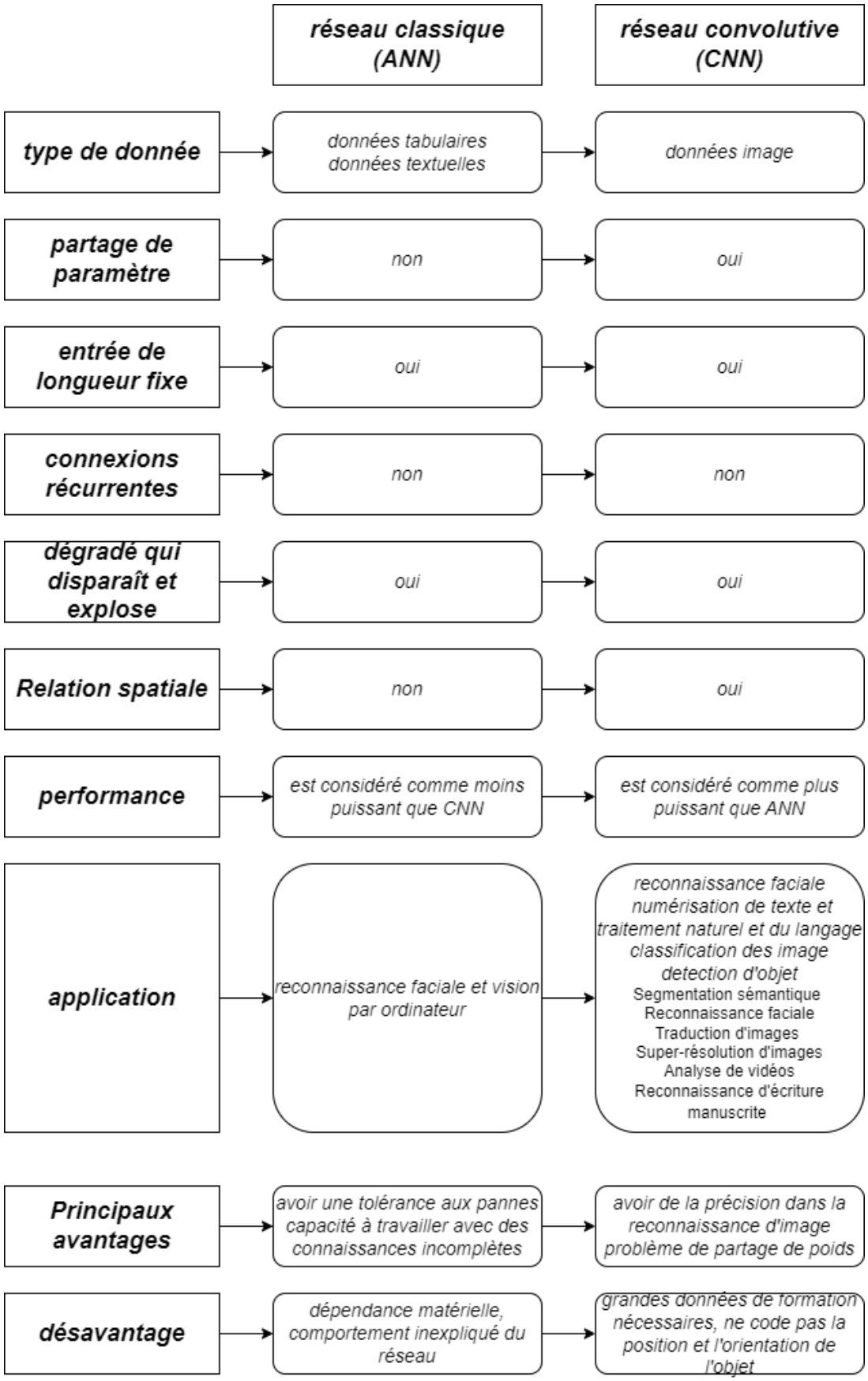


FIGURE 2.11 – Comparaison entre les reseaux ANN et CNN

10 Les Hypers paramètres[19]

Un hyper paramètre est un paramètre qui est défini avant le début de l'apprentissage d'un modèle d'apprentissage automatique et qui contrôle le comportement de l'algorithme d'apprentissage. Contrairement aux paramètres du modèle, qui sont appris au cours de l'apprentissage, les hyper paramètres doivent être définis avant l'apprentissage et peuvent avoir une influence significative sur la qualité des prédictions du modèle.
Les exemples courants d'hyper paramètres dans l'apprentissage automatique comprennent le taux d'apprentissage, le nombre d'itérations, la taille du lot, la profondeur du réseau de neurones, la régularisation et les fonctions d'activation. Le choix optimal des hyper paramètres peut être crucial pour obtenir des résultats de prédiction précis et généralisables.

11 Méthodes de recherche d'hyper paramètres [24]

11.1 La recherche aléatoire

La recherche aléatoire (Random Search en anglais) est une méthode couramment utilisée pour la recherche d'hyper paramètres dans les modèles de machine Learning. Cette méthode consiste à échantillonner aléatoirement les valeurs des hyper paramètres à partir d'une distribution spécifiée et à entraîner le modèle avec chaque ensemble d'hyper paramètres.
La recherche aléatoire peut être mise en œuvre en spécifiant une distribution de probabilité pour chaque hyper paramètre, puis en échantillonnant des valeurs à partir de ces distributions pour chaque essai. Par exemple, une distribution uniforme peut être utilisée pour l'échantillonnage de la taille du lot, tandis qu'une distribution log-uniforme peut être utilisée pour l'échantillonnage du taux d'apprentissage.
Une fois que les essais sont effectués, les résultats peuvent être analysés pour identifier les ensembles d'hyper paramètres qui ont donné les meilleurs résultats en termes de performance de validation. Ces ensembles peuvent alors être utilisés pour entraîner le modèle final.

11.2 La recherche en grille

La recherche en grille (Grid Search en anglais) est une méthode couramment utilisée pour la recherche d'hyper paramètres dans les modèles de machine Learning. Cette méthode consiste à définir une grille d'hyper paramètres, où chaque combinaison possible est essayée de manière exhaustive, puis à entraîner le modèle avec chaque combinaison d'hyper paramètres.
La recherche en grille est une méthode simple mais coûteuse en termes de temps et de ressources, car elle peut nécessiter un grand nombre d'essais pour couvrir efficacement l'espace des hyper paramètres. Cependant, cette méthode a l'avantage de garantir que toutes les combinaisons d'hyper paramètres possibles sont explorées.

11.3 L'optimisation bayésienne [25]

L'optimisation bayésienne est une approche pour rechercher les hyperparamètres des modèles de machine learning. Elle repose sur la théorie des probabilités afin de trouver les combinaisons optimales d'hyperparamètres en minimisant le nombre d'essais nécessaires. Contrairement aux méthodes de recherche en grille ou aléatoires qui effectuent des essais systématiques ou aléatoires, l'optimisation bayésienne explore sélectivement l'espace des hyperparamètres en fonction des informations acquises des essais précédents. Ainsi, elle cherche à trouver la meilleure combinaison d'hyperparamètres en utilisant les connaissances accumulées au fur et à mesure des essais.
Le principe de base de l'optimisation bayésienne est de modéliser l'espace des hyper paramètres comme une distribution de probabilité et de sélectionner les hyper paramètres les plus prometteurs pour les essais suivants. La distribution de probabilité est souvent modélisée à l'aide d'un processus stochastique appelé processus de régression gaussien (Gaussian Process Regression en anglais), qui est capable de fournir une estimation de la fonction objective pour les combinaisons d'hyper paramètres non encore essayées.
L'optimisation bayésienne peut être plus efficace que la recherche en grille ou la recherche aléatoire car elle utilise les résultats des essais précédents pour mettre à jour la distribution de probabilité et ainsi concentrer la recherche sur les régions de l'espace des hyper paramètres qui sont plus susceptibles de donner de bons résultats. Cela peut être particulièrement utile lorsque l'espace des hyper paramètres est grand ou lorsque la fonction objectif est complexe et présente de nombreux optima locaux.
Cependant, l'optimisation bayésienne peut être plus complexe à mettre en œuvre que les autres méthodes de recherche d'hyper paramètres et nécessite souvent des connaissances préalables sur les distributions de probabilité appropriées pour les hyper paramètres. Elle est également plus coûteuse en temps de calcul que la recherche en grille ou la recherche aléatoire, en particulier lorsque la fonction objective nécessite des calculs informatiques lourds.
Malgré ces limitations, l'optimisation bayésienne est largement utilisée dans le domaine de l'apprentissage automatique et a donné des résultats prometteurs dans de nombreux domaines.

12 Types d'hyperparamet́res

12.1 Les hyperparamètres des layers (Les hyperparamètres de la structure du réseau) [26]

Les hyperparamètres des layers sont les paramètres qui sont définis avant l'entraînement du modèle de deep learning et qui influencent la façon dont le modèle apprend et généralise les données. Les hyperparamètres varient selon le type de layer utilisé, mais voici quelques exemples courants :

1. Le nombre de neurones : cela détermine le nombre de sorties que le layer produira.
2. La fonction d'activation : cela détermine comment les sorties du layer seront transformées en entrées pour le layer suivant. Les fonctions courantes incluent la fonction sigmoïde, la fonction ReLU et la fonction tangente hyperbolique.

3. le dropout : cela permet de régulariser le modèle en supprimant aléatoirement certains neurones du layer pendant l'entraînement prent la valeur [0.1,0.5].
4. La régularisation L1 ou L2 : cela ajoute une pénalité sur les poids du layer pour empêcher le modèle de surapprendre.
5. Le stride : cela détermine le nombre de pixels que le filtre se déplace à chaque passage.
6. La taille du filtre : cela détermine la taille de la fenêtre qui glisse sur l'entrée.
7. Le padding : cela permet de conserver la taille de l'entrée en ajoutant des zéros autour des bords.

12.2 Les hyperparamètres de compilation du modèle(Les hyperparamètres de l'algorithme d'apprentissage)[27]

Les hyperparamètres de compilation du modèle sont des paramètres qui sont définis lors de la compilation d'un modèle de deep learning. Ces paramètres affectent la façon dont le modèle est entraîné, évalué et optimisé lors de la phase d'apprentissage. Les hyperparamètres courants de compilation incluent :

1. La fonction de perte (ou loss en anglais) : cela détermine la façon dont l'erreur est calculée entre les prédictions du modèle et les valeurs réelles de la sortie.
2. L'optimiseur : cela détermine l'algorithme d'optimisation utilisé pour ajuster les poids du modèle pendant l'entraînement. Les optimiseurs courants incluent l'optimiseur stochastique par descente de gradient (SGD), l'optimiseur Adam et l'optimiseur RMSprop.
3. Les métriques d'évaluation : cela détermine les mesures de performance du modèle qui sont affichées pendant l'entraînement et l'évaluation. Les métriques courantes incluent la précision, le rappel et la F1-score.
4. Le taux d'apprentissage : cela détermine le taux auquel les poids du modèle sont ajustés pendant l'entraînement.
5. Le batch size : cela détermine le nombre d'exemples d'entraînement utilisés pour mettre à jour les poids du modèle à chaque étape.

12.3 Les hyperparamètres d'exécution du modèle(Les hyperparamètres du processus d'entraînement)[27]

Les hyperparamètres d'exécution du modèle font référence aux paramètres qui sont définis lors de l'exécution du modèle de deep learning et qui influencent la façon dont le modèle est entraîné et évalué. Les hyperparamètres courants d'exécution incluent :

1. Le nombre d'epochs : cela détermine le nombre de fois que le modèle passe par l'ensemble complet de données d'entraînement lors de l'apprentissage.
2. La taille du lot de validation : cela détermine le nombre d'échantillons utilisés pour évaluer le modèle à chaque epoch.
3. Le mode d'entraînement : cela peut être défini comme "entraînement" pour entraîner le modèle, "évaluation" pour évaluer le modèle sur un ensemble de données et "inférence" pour utiliser le modèle pour faire des prédictions.

4. L'ensemble de données d'entraînement : cela détermine l'ensemble de données qui est utilisé pour entraîner le modèle.
5. L'ensemble de données de validation : cela détermine l'ensemble de données qui est utilisé pour évaluer le modèle pendant l'entraînement.
6. L'ensemble de données de test : cela détermine l'ensemble de données qui est utilisé pour évaluer le modèle après l'entraînement.

12.4 Les hyperparamètres de prétraitement [19]

Les hyperparamètres de prétraitement font référence aux choix que l'on peut faire lors de la préparation des données avant de les utiliser pour entraîner un modèle de machine learning. Voici quelques exemples d'hyperparamètres de prétraitement courants :

— La normalisation des données : cela peut inclure la normalisation des valeurs des caractéristiques pour qu'elles soient dans une plage spécifique, ou la normalisation des données pour qu'elles aient une moyenne nulle et une variance unitaire.
— La sélection des caractéristiques : cela consiste à sélectionner un sous-ensemble des caractéristiques qui sont les plus pertinentes pour la tâche à accomplir. La sélection peut être effectuée manuellement ou à l'aide d'algorithmes automatisés tels que la régression Lasso ou les arbres de décision.
— L'encodage des variables catégorielles : cela peut inclure la transformation des variables catégorielles en variables binaires ou l'encodage en nombre entier.
— La gestion des données manquantes : cela peut inclure l'imputation des données manquantes à l'aide de diverses méthodes, telles que la moyenne ou la médiane, ou en utilisant des algorithmes d'apprentissage automatique pour prédire les valeurs manquantes.
— L'augmentation des données : cela consiste à générer de nouvelles données synthétiques à partir des données existantes afin d'augmenter la taille de l'ensemble de données d'entraînement et de renforcer la capacité de généralisation du modèle.

13 Les Transformateurs[28]

Un transformateur dans l'apprentissage automatique est un modèle d'apprentissage en profondeur qui utilise les mécanismes de l'attention, en pesant différemment l'importance de chaque partie des données d'entrée. Les transformateurs de l'apprentissage automatique sont composés de plusieurs couches d'auto-attention. Ils sont principalement utilisés dans les sous-domaines de l'IA du traitement du langage naturel (TAL) et de la vision par ordinateur (CV).

14 Transformateur de vision (VIT) [29]

Le transformateur de vision (ViT) est un transformateur utilisé dans le domaine de la vision par ordinateur qui fonctionne sur la base de la nature fonctionnelle des transformateurs utilisés dans le domaine du traitement du langage naturel. En interne, le transformateur apprend en mesurant la relation entre les paires de jetons d'entrée. En vision par ordinateur, nous pouvons utiliser les patchs d'images comme jeton. Cette

relation peut être apprise en fournissant une attention dans le réseau. Cela peut être fait soit en conjonction avec un réseau convolutif, soit en remplaçant certains composants des réseaux convolutifs. Ces structures du réseau peuvent être appliquées aux tâches de classification d'images.

14.1 Type de tranformateur de vision

Il existe quatre types de transformateur de vision :
- ViT à échelle uniforme.
- ViT à plusieurs échelles.
- ViT hybrides avec convolutions.
- ViT auto-supervisés.

15 Mecanisme d'attention[W10]

Le mécanisme d'attention dans l'apprentissage en profondeur est une technique utilisée pour améliorer les performances d'un réseau de neurones en permettant au modèle de se concentrer sur les données d'entrée les plus importantes tout en générant des prédictions. Ceci est accompli en pondérant les données d'entrée afin que le modèle donne la priorité à certaines propriétés d'entrée par rapport à d'autres. Par conséquent, le modèle peut produire des prédictions plus précises en ne considérant que les variables d'entrée les plus significatives.

15.1 Types d'attentions[W10]

1. **Attention Généralisée**
   Utiliser dans les réseaux neuronaux permettant à un modèle de se focaliser sur différentes zones de son entrée, de manière similaire à la façon dont les êtres humains portent leur attention sur différents éléments de leur environnement. Cette technique est utilisée dans divers domaines tels que la reconnaissance d'images, le traitement du langage naturel et la traduction automatique.
   En effet, un modèle utilisant l'attention généralisée apprend à identifier automatiquement les parties les plus importantes de l'entrée pour une tâche donnée, en allouant davantage de ressources informatiques à ces parties spécifiques. Cette approche peut améliorer l'efficacité du modèle et optimiser sa performance pour une variété de tâches.
2. **Auto-Attention**
   L'auto-attention, également appelée intra-attention, est un mécanisme d'attention utilisé dans les modèles de réseaux de neurones qui permet à un modèle de se concentrer sur différents aspects de son entrée sans avoir besoin de supervision ou d'entrées extérieures. Cette technique est particulièrement utile pour les tâches de traitement du langage naturel, où le modèle doit comprendre les liens entre les différents mots dans une phrase pour produire des résultats précis. Dans l'auto-attention, le modèle évalue la similarité entre chaque paire de vecteurs d'entrée, puis pondère la contribution de chaque vecteur d'entrée à la sortie en fonction de

ces scores de similarité. Ainsi, le modèle peut se concentrer automatiquement sur les parties les plus pertinentes de l'entrée sans nécessiter de surveillance extérieure.

3. **Attention Multi-Tête**
   L'attention multi-tête est une sorte de mécanisme d'attention utilisé dans certains modèles de réseaux de neurones. L'utilisation de plusieurs « têtes » ou processus d'attention permet au modèle de se concentrer sur plusieurs aspects de ses informations à la fois.
   Ceci est bénéfique pour des tâches telles que le traitement du langage naturel où le modèle doit comprendre les liens entre différents mots dans une phrase.
   Un modèle d'attention multi-tête transforme l'entrée en plusieurs espaces de représentation distincts avant d'appliquer un mécanisme d'attention séparé à chaque espace de représentation.

$$Attention(Q, V, K) = softmax(Q * K^T/\sqrt{d_k}) * V \tag{2.2}$$

Les sorties de chaque mécanisme d'attention sont ensuite intégrées, permettant au modèle de traiter l'information à partir de nombreux points de vue. Cela peut améliorer les performances sur une variété de tâches tout en rendant le modèle plus résilient et efficace.

$$multi\_head(Q, V, K) = concat(head_1, head_2, head_3, ..., head_n) \cdot W^o \tag{2.3}$$

$$head_i = Attention(Q, V, K) \tag{2.4}$$

On à :
Q : La matrice de requête. Il représente l'entrée pour laquelle nous voulons calculer les poids d'attention. K : La matrice clé. Il contient les informations utilisées pour calculer les poids d'attention. V : La matrice de valeurs. Il contient les valeurs qui seront pondérées et combinées en fonction des scores d'attention. d_k :Représente la dimensionnalité des vecteurs de requête, de clé et de valeur.
W^o Matrice de poids.

15.2 Multi head Attention de vidéo

On trouve 4 methodes pour calculer l'attention d'une video :

Modèle Attention 01 :L'attention spatio-temporelle (Attention Spatio-Temporal)[62]

L'attention spatio-temporelle est un type d'attention utilisé pour les tâches impliquant des données spatiales et temporelles, telles que la reconnaissance d'actions dans des vidéos ou la prédiction de la trajectoire d'objets dans des séquences temporelles.
Dans l'attention spatio-temporelle, les informations spatiales et temporelles sont combinées pour produire des représentations plus riches en informations pour chaque élément de la séquence. Cette attention fonctionne en deux étapes :
L'attention spatiale : dans cette étape, l'attention est appliquée sur les informations spatiales pour déterminer les parties importantes de l'image. Cela peut être réalisé en utilisant une carte de chaleur qui met en évidence les régions les plus pertinentes de l'image

pour la tâche en question. L'attention spatiale peut également être réalisée en utilisant les réseaux de neurones convolutionnels (CNN) pour extraire les caractéristiques importantes de l'image.

L'attention temporelle : dans cette étape, l'attention est appliquée sur les informations temporelles pour déterminer les moments clés de la séquence temporelle. Cela peut être réalisé en utilisant des réseaux de neurones récurrents (RNN) pour modéliser les relations temporelles entre les différents éléments de la séquence.

Une fois que l'attention spatiale et temporelle ont été appliquées, les représentations contextuelles finales sont calculées en prenant une combinaison pondérée des représentations spatiales et temporelles pour chaque élément de la séquence.

L'attention spatiotemporelle permet de capturer des informations riches et complexes à partir de données spatiales et temporelles, ce qui peut améliorer la performance de nombreux modèles de reconnaissance d'actions et de prédiction de trajectoires d'objets dans des vidéos ou des séquences temporelles.

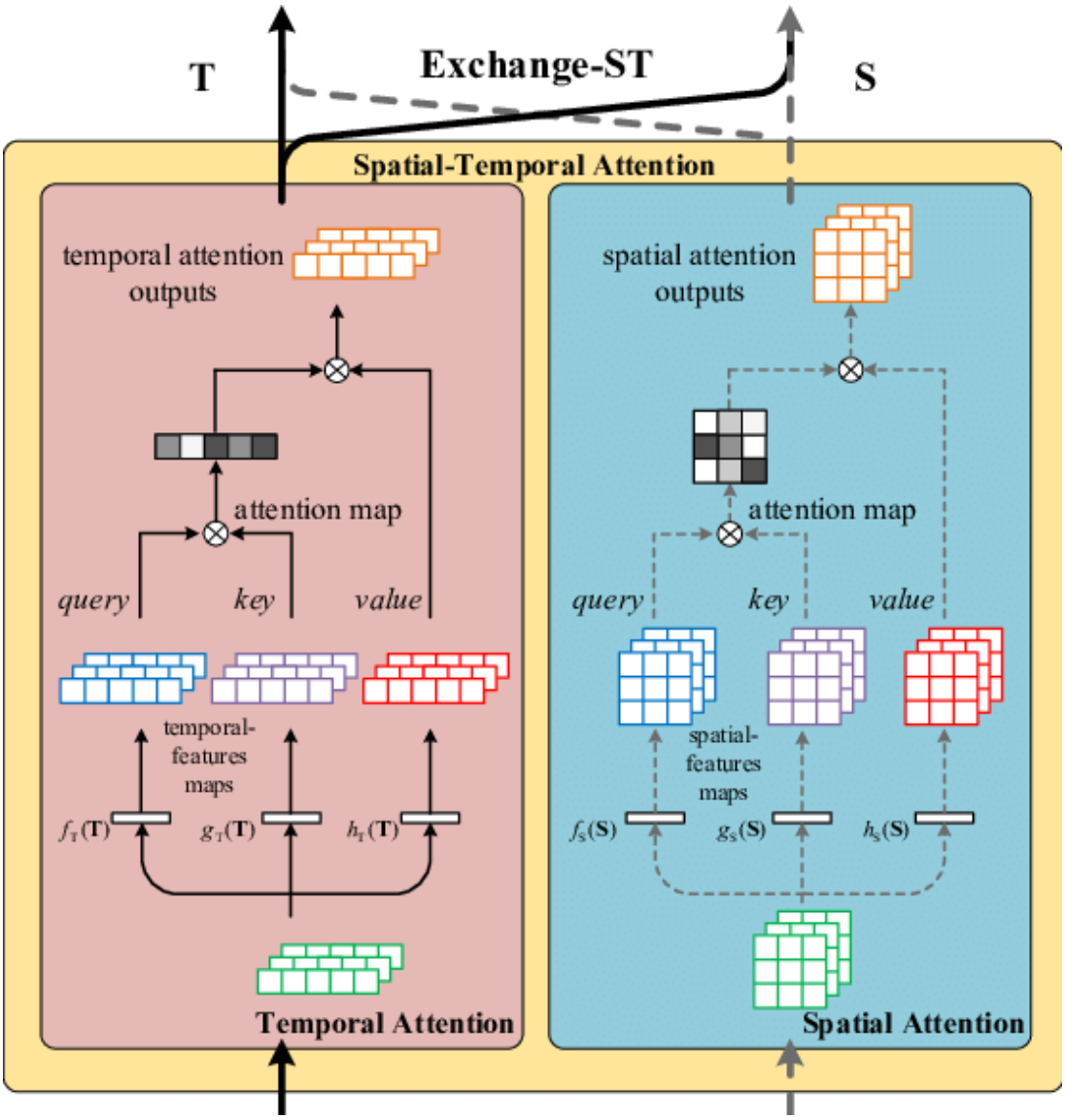


FIGURE 2.12 – Attention Spatio-Temporal[49]

Modèle Attention 02 : Encodeur factorisé [63]

Le codeur factorisé est une variante du codeur standard utilisé dans les transformateurs pour traiter les séquences d'entrée. Cette variante est plus rapide et nécessite moins de mémoire que l'encodeur standard.
Dans le codeur standard, chaque couche de codeur est composée d'une couche d'attention à plusieurs têtes, suivie d'une couche d'anticipation de position. La couche d'attention à plusieurs têtes prend les intégrations de séquences en entrée et calcule les représentations contextuelles de chaque élément en utilisant l'auto-attention. La couche d'anticipation positionnelle est ensuite appliquée à chaque représentation contextuelle pour produire des représentations plus riches en informations.
Dans le codeur factorisé, la couche d'attention multi-têtes est remplacée par deux couches : une couche d'attention factorisée et une couche d'anticipation positionnelle. La couche d'attention pondérée calcule les représentations contextuelles de chaque élément à l'aide de l'auto-attention factorisée, qui est une variante de l'auto-attention standard qui utilise des vecteurs clé et valeur factorisés pour réduire la complexité de calcul. La couche d'anticipation positionnelle est ensuite appliquée à chaque représentation contextuelle pour Produire des représentations plus riches en informations.
L'utilisation de l'auto-attention factorisée permet de réduire la complexité de calcul de l'encodeur, car les produits scalaires ne sont plus calculés entre des vecteurs de grande dimension. Cela réduit également la quantité de mémoire nécessaire pour stocker les vecteurs de clé et de valeur. De plus, l'encodeur factorisé nécessite moins de calculs que l'encodeur standard, ce qui le rend plus rapide.
Cependant, l'encodeur factorisé peut être moins performant que l'encodeur standard si la taille des vecteurs clé et valeur factorisés est choisie trop petite, car cela réduit l'expressivité du mécanisme d'attention.

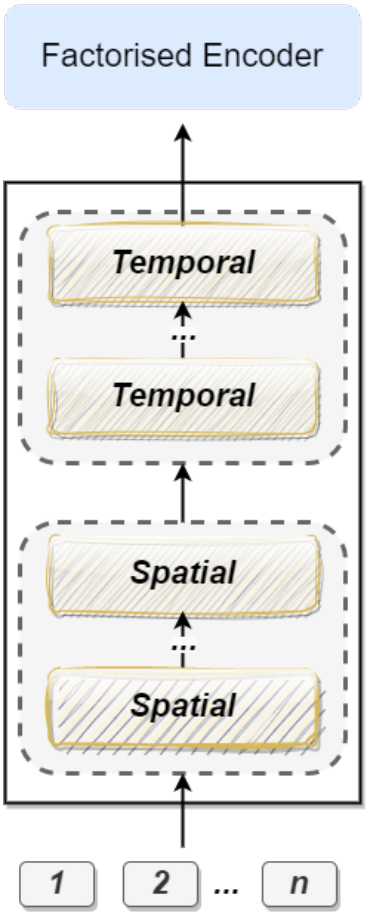


FIGURE 2.13 – Représentation de l'encodeur factorisé

Modèle Attention 03 : Auto_attention factorisée [64]

L'auto-attention factorisée (ou factorized self-attention en anglais) est une variante du mécanisme d'attention utilisé dans les transformateurs pour calculer les représentations contextuelles de chaque élément d'une séquence. Cette variante est plus rapide et nécessite moins de mémoire que l'auto-attention standard.
Dans l'auto-attention standard, chaque élément de la séquence est considéré à la fois comme une requête, une clé et une valeur. Les scores d'attention sont calculés en prenant le produit scalaire entre chaque paire de vecteurs requête-clé, puis en appliquant une fonction softmax pour obtenir les poids d'attention. Enfin, les poids d'attention sont utilisés pour calculer la somme pondérée des vecteurs valeur, qui sont la représentation contextuelle finale de chaque élément.
Dans l'auto-attention factorisée, les vecteurs clé et valeur sont factorisés en plusieurs vecteurs plus petits, chacun de taille r. Par exemple, si la dimension des vecteurs clé et valeur est de 512 et que r est fixé à 8, chaque vecteur clé et valeur est divisé en 8 vecteurs de taille 64.
Les scores d'attention sont calculés en prenant le produit scalaire entre chaque vecteur requête et les vecteurs clé plus petits correspondants, pour chaque groupe de vecteurs clé et valeur. Ensuite, les scores d'attention obtenus sont concaténés et passés dans une fonction softmax pour obtenir les poids d'attention. Les poids d'attention sont finalement utilisés pour calculer la somme pondérée des vecteurs valeur, qui sont la représentation contextuelle finale de chaque élément.

La factorisation des vecteurs clé et valeur en vecteurs plus petits permet de réduire la complexité calculatoire de l'auto-attention, car les produits scalaires ne sont plus calculés entre des vecteurs de grande dimension. Cela réduit également la quantité de mémoire nécessaire pour stocker les vecteurs clé et valeur. Cependant, la performance de l'auto-attention factorisée peut être moins bonne que celle de l'auto-attention standard si la taille des vecteurs plus petits est choisie de manière trop petite, car cela réduit l'expressivité du mécanisme d'attention.

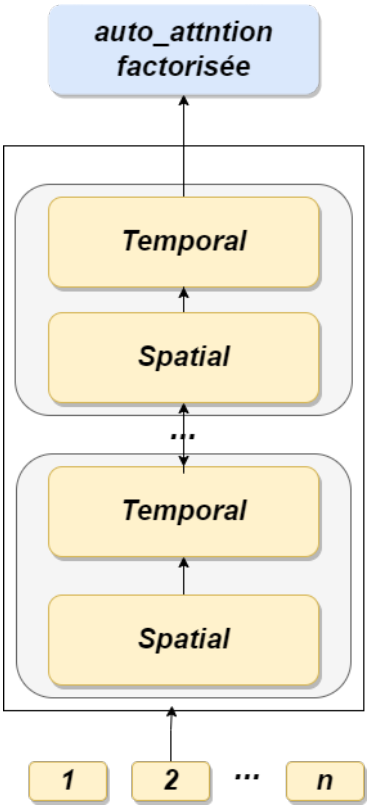


FIGURE 2.14 – Autoattention factorisée

Modèle Attention 04 :L'attention factorisée du produit scalaire (Factorised dot-product attention)[65]

L'attention factorisée du produit scalaire est une variante du mécanisme d'auto-attention utilisé dans les transformateurs. Ce mécanisme d'attention est plus rapide et nécessite moins de mémoire par rapport à l'attention standard du produit scalaire.
Dans l'attention standard du produit scalaire, les scores d'attention sont calculés en prenant le produit scalaire entre le vecteur de requête et le vecteur clé, puis en divisant le résultat par la racine carrée de la dimension du vecteur clé. Ceci est suivi par l'application d'une fonction softmax pour obtenir les poids d'attention, qui sont ensuite utilisés pour calculer la somme pondérée des vecteurs de valeur.
Dans l'attention factorisée du produit scalaire, les vecteurs clé et valeur sont divisés en vecteurs plus petits. Cela se fait en factorisant les matrices de clé et de valeur en deux matrices plus petites, où les lignes sont divisées en plus petits morceaux. Le nombre de morceaux est généralement désigné par l'hyperparamètre r. Par exemple, si les vecteurs clé et valeur ont une dimension de 512 et que r est défini sur 8, alors chaque vecteur clé et valeur est divisé en 8 vecteurs plus petits de dimension 64.

Les scores d'attention sont ensuite calculés en prenant le produit scalaire entre le vecteur de requête et chaque vecteur clé plus petit. Ceci est suivi en concaténant les scores d'attention résultants de tous les vecteurs plus petits, puis en appliquant la fonction softmax. Les poids d'attention obtenus à partir de la fonction softmax sont ensuite utilisés pour calculer la somme pondérée des vecteurs de valeur, où chaque vecteur de valeur est divisé en le même nombre de vecteurs plus petits que les vecteurs clés.
L'utilisation de vecteurs plus petits réduit la complexité de calcul du mécanisme d'attention, puisque le produit scalaire n'est effectué qu'entre des vecteurs plus petits plutôt qu'avec les vecteurs de clé et de requête pleine grandeur. De plus, les vecteurs plus petits nécessitent moins de mémoire, ce qui rend le mécanisme d'attention plus efficace en mémoire. Cependant, les performances de l'attention du produit scalaire factorisé peuvent ne pas être aussi bonnes que l'attention du produit scalaire standard lorsque l'hyperparamètre r est défini sur une petite valeur, car cela réduit l'expressivité du mécanisme d'attention.

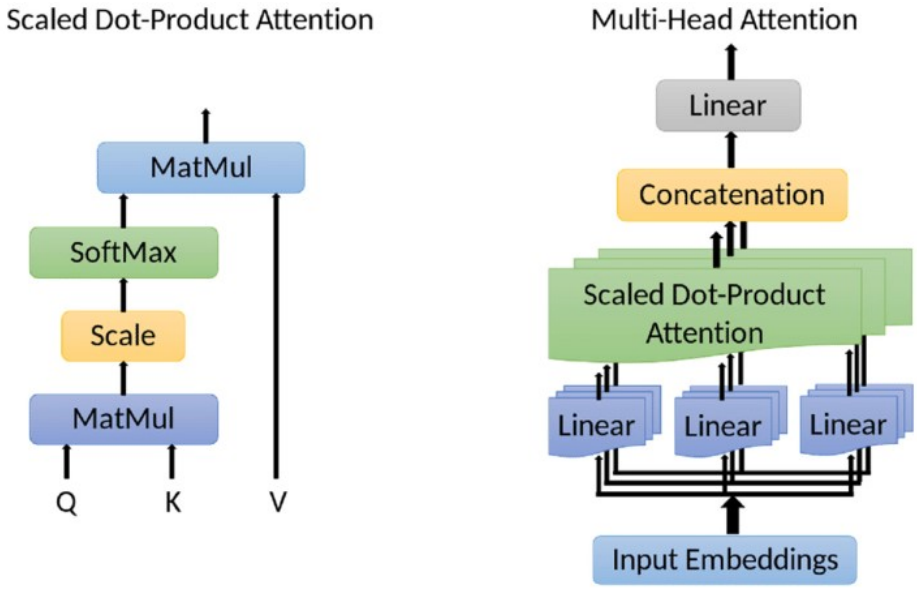


FIGURE 2.15 – L'attention factorisée de produit scalaire Model-4 [50]

16 Architecture de transformateur

Le transformateur utilise une architecture codeur-décodeur. L'encodeur extrait des caractéristiques d'une phrase d'entrée, et le décodeur utilise les caractéristiques pour produire une phrase de sortie (traduction).

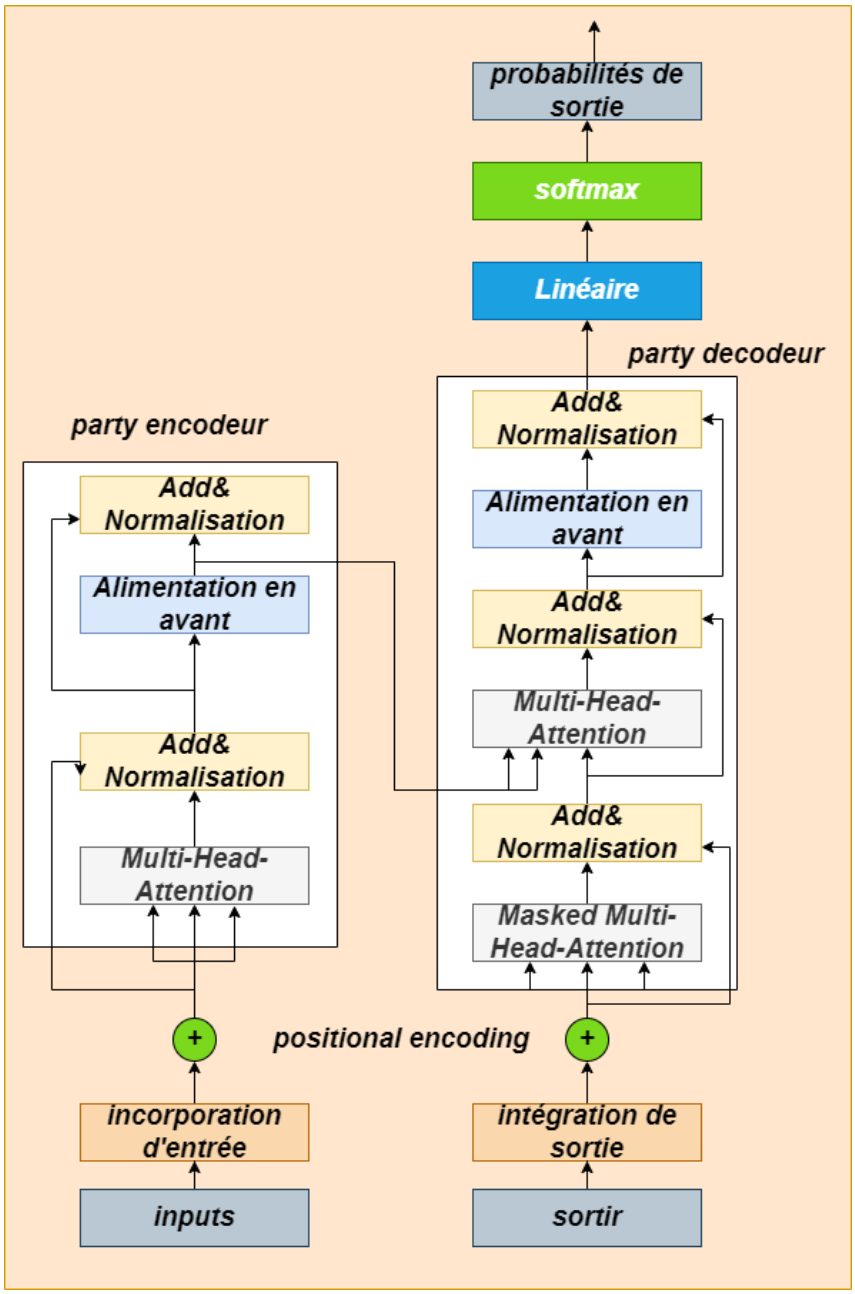


FIGURE 2.16 – Representation de l'architecture d'un transformateur

16.1 Encodeur(codeur)[30]

L'encodeur du transformateur se compose de plusieurs blocs d'encodeur. Une phrase d'entrée passe par les blocs d'encodeur et la sortie du dernier bloc d'encodeur devient les caractéristiques d'entrée du décodeur.

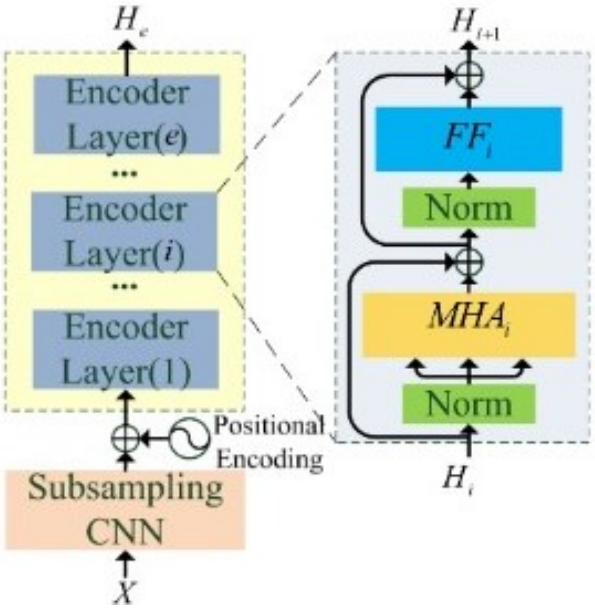


FIGURE 2.17 – Architecture de l'encodeur [51]

1. **L'attention multi-têtes (MHA)**
   C'est un réseau chargé de générer des cartes d'attention à partir des jetons visuels intégrés donnés. Ces cartes d'attention aident le réseau à se concentrer sur les régions les plus critiques de l'image, telles que les objets.

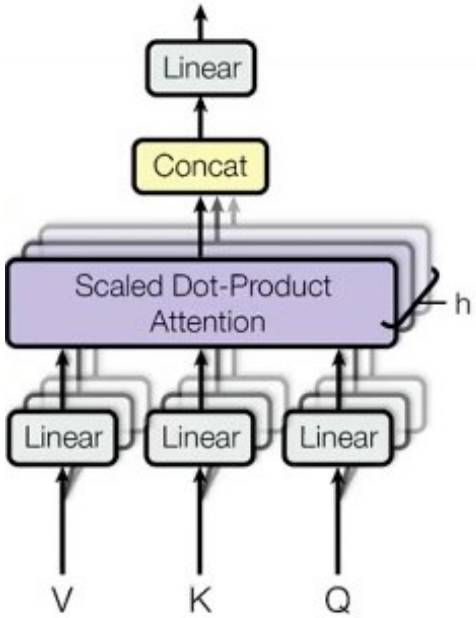


FIGURE 2.18 – represention de L'attention multitêtes (MHA) [52]

2. **Normalisation et résiduel connection**
   La couche normalisation maintient le processus de formation sur la bonne voie et permet au modèle de s'adapter aux variations entre les images de formation.
3. **Couche de perceptrons multicouches (MLP) :**
   MLP est un réseau de classification à deux couches avec GELU (Gaussian Error

Linear Unit) à la fin. Le bloc MLP final, également appelé tête MLP, est utilisé comme sortie du transformateur. Une application de softmax sur cette sortie peut fournir des étiquettes de classification (c'est-à-dire si l'application est Classification d'image).

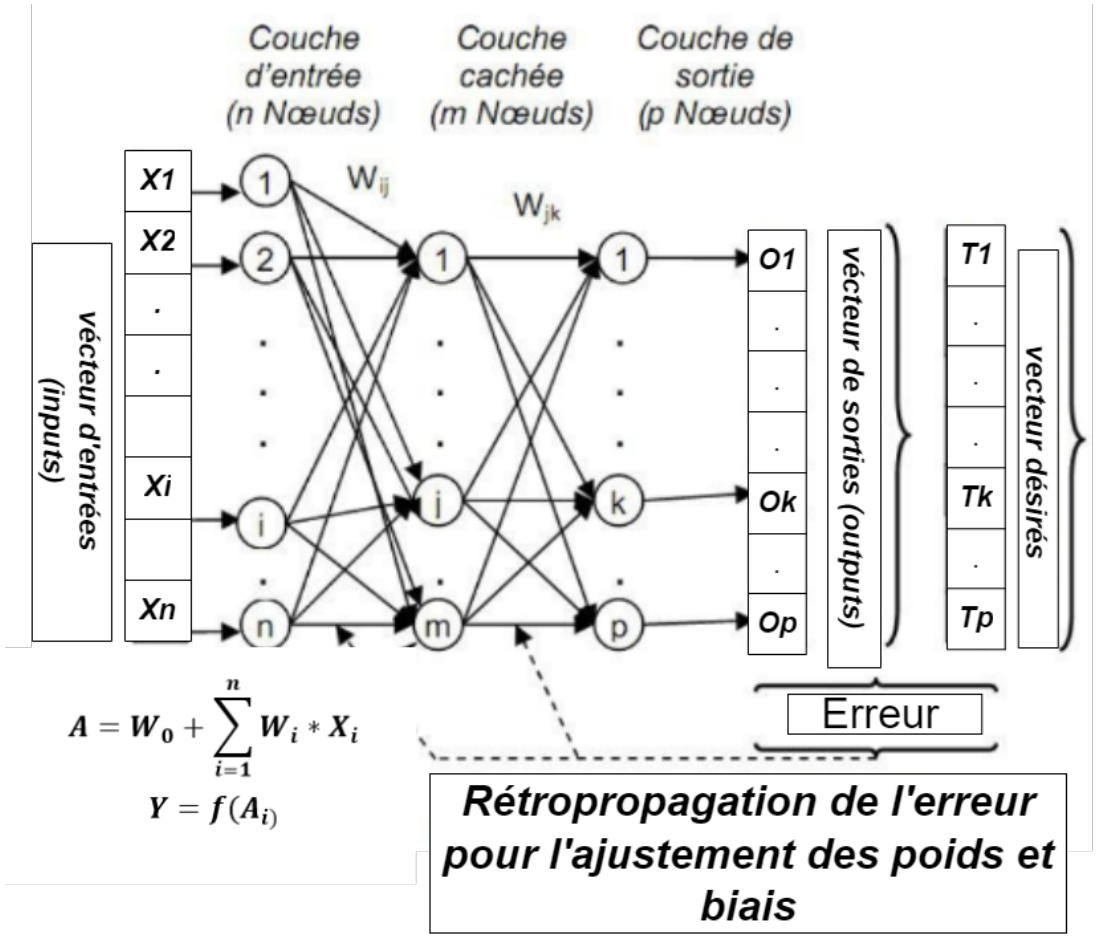


FIGURE 2.19 – Architecture du perceptrons multicouches (MLP)[53]

16.2 Décodeur[30]

Le côté décodeur a beaucoup de composants partagés avec le côté codeur. Par conséquent, cette section ne sera pas aussi détaillée que la précédente. Les principales différences entre le décodeur et l'encodeur sont que le décodeur prend deux entrées et applique une attention multi-tête deux fois, l'une d'entre elles étant "masquée". De plus, la couche linéaire finale dans le décodeur a la taille (c'est-à-dire le nombre d'unités) égale au nombre de mots dans le dictionnaire cible (dans ce cas, le dictionnaire de langue française). Chaque unité se verra attribuer un score ; le softmax est appliqué pour convertir ces scores en probabilités indiquant la probabilité que chaque mot soit présent dans la sortie.

1. **Attention masquée à plusieurs têtes [30]**
   Le processus de l'attention multi-tête masquée est similaire à celui de l'attention multi-tête régulière. La seule différence est qu'après avoir multiplié les matrices Q et K, et les avoir mises à l'échelle, un masque spécial est appliqué sur la matrice résultante avant d'appliquer le softmax . L'objectif est d'avoir chaque mot à une

position spécifique "i" dans le texte pour ne s'occuper que de toutes les autres positions dans le texte jusqu'à sa position actuelle incluse (position 0 jusqu'à la position i). Ceci est important dans la phase d'apprentissage, car lors de la prédiction du mot à la position i+1, le modèle ne prêtera attention qu'à tous les mots avant cette position. Par conséquent, toutes les positions après i, sont masquées et définies sur l'infini négatif avant de les passer à l'opération softmax, ce qui se traduit par des 0 dans le filtre d'attention.

	je	*suis*	*un*	*étudiant*
je	*1*	*0*	*0*	*0*
suis	*0.02*	*0.98*	*0*	*0*
un	*0.05*	*0.20*	*0.75*	*0*
étudiant	*0.38*	*0.02*	*0.05*	*0.55*

FIGURE 2.20 – Exemple de filtre de masque d'attention

16.3 Avantages des transformateurs

— Les transformateurs peuvent apprendre efficacement les dépendances sémantiques à longue portée.
— La formation des transformateurs est parallélisable (au niveau du jeton), alors que les modèles basés sur RNN sont intrinsèquement séquentiels

17 Vidéo Vision transformateur[31]

Modèle basé sur un transformateur pur pour la classification vidéo. en trouve Deux façons d'intégrer des clips vidéo pour le calcul de l'attention :
— Echantillonnage de trame uniforme.
— Intégration de Tubelet.

17.1 Échantillonnage uniforme de trames(Uniform frame sampling)

une méthode simple de tokenisation(La tokenisation d'une vidéo peut être un processus complexe, car il implique la transformation des données audiovisuelles en une séquence de tokens ou d'unités discrètes de sens.) de la vidéo d'entrée consiste à échantillonner uniformément nt trames du clip vidéo d'entrée, à intégrer chaque trame 2D indépendamment en utilisant la même méthode que ViT, et à concaténer tous ces tokens ensemble.

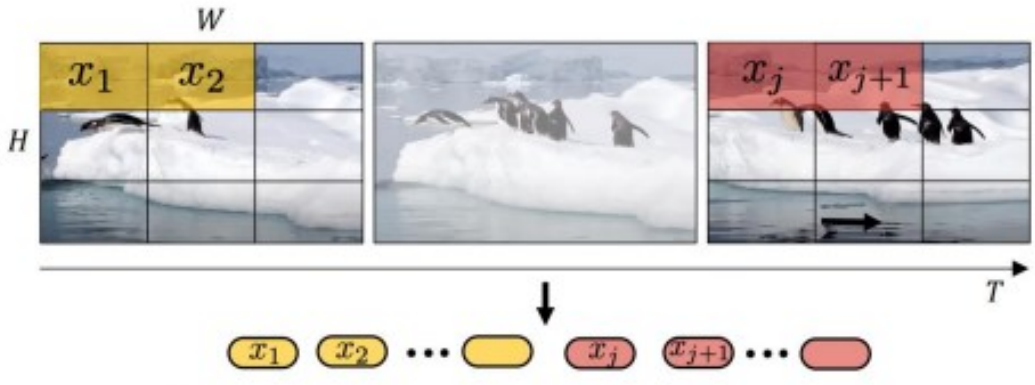


FIGURE 2.21 – Echantillonnage de trame uniforme [54]

17.2 Incorporation de tubelettes(Tubelet embedding)

Incorporation de tubelettes Une autre méthode, consiste à extraire du volume d'entrée des "tubes" spatio-temporels qui ne se chevauchent pas et à les projeter linéairement dans Rd. Cette méthode est une extension de l'intégration de ViT à la 3D, et correspond à une convolution 3D. Pour un tubelet de dimension t × h*w, les tokens sont extraits respectivement des dimensions temporelle, de hauteur et de largeur. Des dimensions de tubelet plus petites entraînent donc un plus grand nombre de tokens, ce qui augmente le calcul. Intuitivement, cette méthode fusionne les informations spatio-temporelles pendant la tokénisation, contrairement à l'"échantillonnage de trame uniforme" où les informations temporelles de différentes trames sont fusionnées par le transformateur.

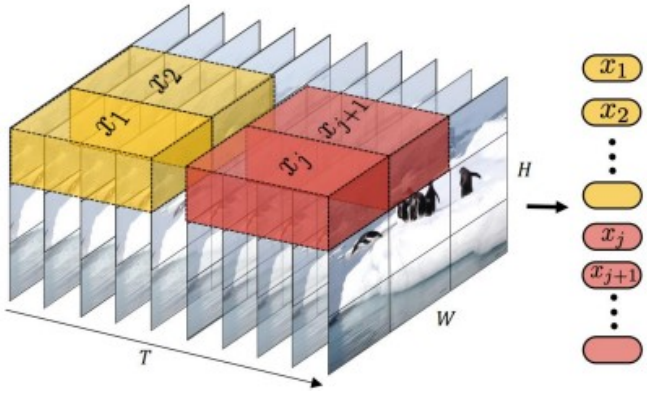


FIGURE 2.22 – représenter Incorporation de tubelettes[54]

17.3 L'architecture de transformateur de Vision

L'architecture globale de ViT peut être présentée en six points principaux :
- Diviser l'image en patchs sans chevauchement (16 x 16, 32 x 32, etc.)
- Aplatir les patchs et produire des plongements linéaires de dimension inférieure à partir des patchs aplatis référencés (Patch Embedding).
- Ajoutez une intégration positionnelle et une classe jeton.

— Les plongements linéaires des patchs, y compris le jeton de classe, sont alimentés dans un réseau de transformateur. qui permettent de capturer les relations spatiales et les dépendances entre les patchs.
— À la sortie du réseau de transformateur, les informations des patchs sont agrégées en utilisant des opérations telles que la moyenne ou le max-pooling. Cela permet de réduire la dimension de la représentation globale de l'image.
— Transmettre les sortie du réseau de transformateur à la tête MLP pour obtenir la prédiction de sortie finale.

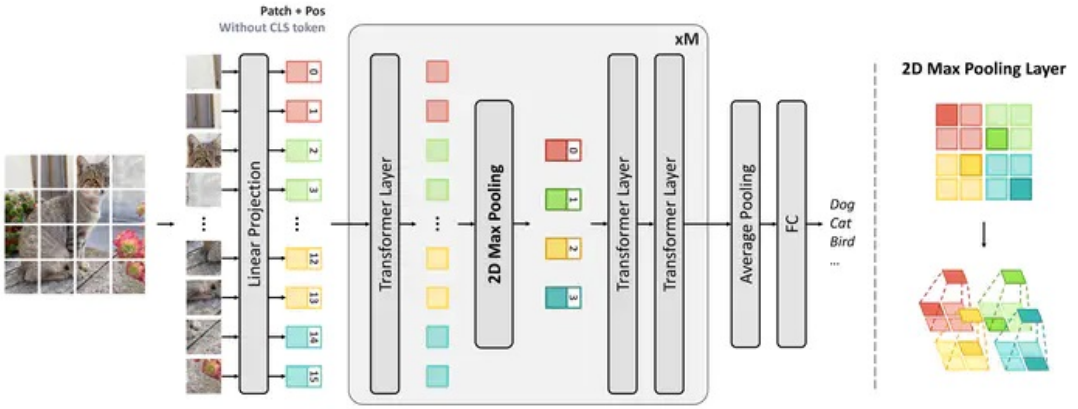


FIGURE 2.23 – rarchitecture de VIT[55]

18 Application de Transformateur de vision

1. Classification
2. Légende des images
3. La reconnaissance faciale
4. Segmentation
5. Réidentification de la personne
6. Détection

19 Mesures de performances

Pour évaluer un apprentissage ,il faut caculer un certain nombre de paramètre telque :

1. Vrai positif 'True Positive Tp' : prédiction positive correcte.
2. Vrai négatif 'True Negative TN' : prédiction negative correcte.
3. Faux positif 'False Positive FP' : prédiction positive incorrecte.
4. Faux négatif 'False Négative FN' : prédiction negative incorrecte.

19.1 Matrice de confusion

La matrice de confusion est un outil de mesure couramment utilisé dans la résolution de problèmes de classification, qu'il s'agisse de problèmes de classification binaire ou multiclasses. Elle permet de représenter et d'analyser les performances d'un modèle de classification en comparant les prédictions du modèle aux valeurs réelles des données de test.

	Classe réelle possitive	Classe réelle négative
Classe réelle possitive	Vrai positive VP	faux positive FP
Classe réelle négative	faux négative FN	Vrai négative VN

TABLE 2.1 – Matrice de confusion

19.2 Précision(Accuracy)[18]

L'accuracy (exactitude) est une métrique largement employée pour évaluer les performances d'un modèle de classification. Elle mesure la proportion de prédictions correctes effectuées par le modèle sur un ensemble de données de test. L'accuracy est exprimée en pourcentage, où une valeur plus élevée indique une meilleure performance du modèle en termes de précision de ses prédictions.
La accuracy peut être calculée comme suit :

$$Accuracy = \frac{(VP + VN)}{(VP + VN + FP + FN)} \tag{2.5}$$

19.3 Taux de vrais positifs/Recall(Sensitivity)

Recall, également connu sous le nom de sensibilité, est une mesure qui représente le rapport entre les vrais positifs et tous les éléments réellement positifs présents dans les données.

$$TVP = \frac{TP}{TP + FN} \tag{2.6}$$

19.4 Taux de vrais négatifs

Probabilité d'une prédiction négative dans un cas négatif.

$$TVN = 1 - \frac{TN}{TN + FN} \tag{2.7}$$

19.5 Taux de faux positifs

Le taux de faux positifs (False Positive Rate, FPR en anglais) est une mesure qui évalue la proportion d'échantillons négatifs incorrectement classés comme positifs par un modèle de classification. Il est calculé par :

$$TFP = 1 - \frac{TN}{TN + FP} \tag{2.8}$$

19.6 Taux_de_faux négatifs

Le taux de faux négatifs (False Negative Rate, FNR en anglais) est une mesure qui évalue la proportion d'échantillons positifs incorrectement classés comme négatifs par un modèle de classification. Il est calculé par :

$$TFN = 1 - \frac{TP}{TN + FP} \tag{2.9}$$

19.7 F-measure

La mesure F représente la moyenne harmonique de la précision (P) et du rappel (R).

$$F1 - Mesure = 2(\frac{precision * recall}{precision + recall}) \tag{2.10}$$

19.8 La courbe ROC

La courbe ROC (Receiver Operating Characteristic Curve) est un graphique qui illustre les performances d'un modèle de classification pour tous les seuils de classification [w6]. Cette courbe représente deux paramètres.

— Sensitivité (fréquence des vraies présence)
— Spécificité (fréquence des fausses présence)

Courbe ROC représentant le pire des cas avec un fort effet du hasard si AUC de 0,5 associée à une capacité de discrimination très faible et un meilleur des cas si AUC de 1 associée à une capacité de discrimination très forte, le modèle donne des prédictions exactes et une capacité de discrimination d'un modèle étudié (AUC compris entre 1 et 0,5)

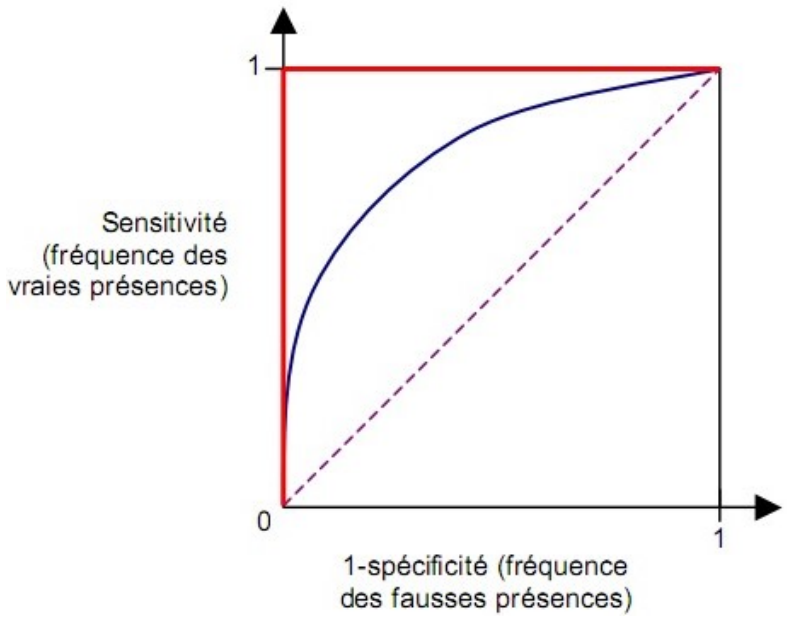


FIGURE 2.24 – La courbe de ROC [56]

20 Traveaux Connexes sur les transformateurs de vision

20.1 ARTICLE 01 "CvT : Introducing Convolutions to Vision Transformers" [32] CvT : Présentation des convolutions dans les transformateurs de vision

Dans cet article, les chercheurs abordent le domaine de la reconnaissance d'images ouíls proposent une architecture appelée Convolutional vision transformer (CvT) qui utilise les transformateur de vision (vit) et les réseaux de neurones convolutifs (CNN), a partir de l'ajout d'une partie convolutive à l'architecture VIT. dans cet architecture, ils ont prend les caractéristiques du CNN avec une conservation d'avantages de vit, pour donner de meilleures performances et une efficacité modulaire. Les auteurs ont comparé les résultats d'entraînement de trois méthodes : CNN, VIT et CVT où ils ont obtenu le meilleur résultat avec CVT, soit 99,39% obtenus en « CIFAR 10 ».

Model	Param (M)	CIFAR 10	CIFAR 100	Pets	Flowers 102
BiT-M	928	98.91	92.17	94.46	99.30
Vit-B/16	86	98.95	91.67	94.43	99.38
ViT-L/16	307	99.16	93.44	94.73	99.61
Vit-H/16	632	99.27	93.82	94.82	99.51
CvT-13	20	98.83	91.11	93.25	99.50
CvT-21	32	99.16	92.88	94.03	99.62
CVT-W24	277	99.39	94.09	94.73	99.72

FIGURE 2.25 – Illustration des resultats de CVT [32]

20.2 ARTICLE 02 "ViViT : A Video Vision Transformer" [31]ViViT : un transformateur de vision vidéo

Dans cet article les chercheurs se sont concentré sur le domaine de classification des videos, ils proposent 3 models de classification de videos(ViT-Base ,vit-large, vit-huge) basées sur les transformateur purs, ce model extrait les caractéristiques temporelles et spacials qui sont utilisées comme des entrées pour les différentes couches de transformateurs.
Il montre comment ajuster le modèle pendant la formation et tirer parti des modèles

d'image pré-formés pour former efficacement le modèle sur de petits ensembles de données.
Ils ont testé ce model sur les bases de données suivantes Kinetics 400 et 600, Epic Kitchens, Something-Something v2 et Moments in Time, les resultat obtenus étaient meilleurs que les autres modèles avec 84.9% d'accuracy pour 3*4 views(3 cultures temporelles et 4 cultures spatiales) et 47.77 Tflops(Tflops indication du nombre de billions d'opérations en virgule flottante qu'un appareil peut effectuer en une seconde).

Method	Top 1	Top 5	Views	TFLOPs
b1VNet	73.5	91.2	-	-
STM	73.7	91.6	-	-
TEA	76.1	92.5	10x3	2.10
TSM-ResNeXt-101	76.3	-	-	-
I3D NL	77.7	93.3	10x3	10.77
CorrNet-101	79.2	-	10x3	6.72
Ip-CSN-152	79.2	93.8	10x3	3.27
LGD-3D R101	79.4	94.4	-	-
Slow Fast R101	79.8	93.9	10x3	7.02
X3D-XXL	80.4	94.6	10x3	5.82
TimeSformer-L	80.7	94.7	1x3	7.14
ViVit-L/16x2FE	80.6	92.7	1x1	3.98
ViVit-L/16x2FE	81.7	93.8	1x3	11.94

TABLE 2.2 – Illustration des resultats de VIVIT [31]

20.3 ARTICLE 03 "CMT : Convolutional Neural Networks Meet Vision Transformers" [33]CMT : les réseaux de neurones convolutifs rencontrent les transformateurs de vision

Dans Cette article les chercheures proposent une nouvel architecture nommées Convolutional Neural Networks Meet Vision Transformers qui utilise les transformateur pour capturer les dépendances à longue portée et des CNN pour extraire des informations locales pour la reconnaissance d'images.
L'objectif de cette Architecture est d'augmeter les performances et le coût de calcul entre CNNs et vit.
L'entrainement de l'architecture CMTs a donné un accuracy élevé par rapport à d'autres méthodes 95.84% pour 1.04B flops (un billion d'opérations en virgule flottante par seconde)et 25.1M paramètres (Les paramètres sont les variables internes ou les poids qu'un modèle apprend au cours du processus de formation) et 51.4% pour 249B flops et 44.5M paramètres

Backbone	#Params	#FLOPs	mAP	AP50	AP75	APs	APM	APL
ConT-M	27.0M	217B	37.9	58.1	40.2	23.0	40.6	50.4
ResNet-101	56.7M	315B	38.5	57.6	41.0	21.7	42.8	50.4
RelationNet++	39.0M	266B	39.4	58.2	42.5	-	-	-
ResNeXt-101-32x4d	56.4M	319B	39.9	59.6	42.7	22.3	44.2	52.5
PVT-S	34.2M	226B	40.4	61.3	43.0	25.0	42.9	55.7
Swin-T	38.5M	245B	41.5	62.1	44.2	25.1	44.9	55.5
Twins-SVT-S	34.3M	209B	42.3	63.4	45.2	26.0	45.5	56.5
Twins-PCPVT-S	34.4M	226B	43.0	64.1	46.0	27.5	46.3	57.3
CMT-S	44.3M	231B	44.3	65.5	47.5	27.1	48.3	59.1

TABLE 2.3 – Illustration des résultats du CMT [33]

20.4 ARTICLE 04 :Estimation de l'âge et de la taille du locuteur à partir du signal vocal à l'aide d'un modèle de mélange de transformateurs bi-encodeur [34]Estimation of speaker age and height from speech signal using bi-encoder transformer mixture model

Dans cet article les chercheurs se sont concentré sur le domaine de détection des audio pour faire une estimation des caractéristiques du locuteur telles que l'âge et la taille, ils proposent un modèle de mélange de transformateurs bi-codeur, utilisant wav2vec 2.0 comme un niveau commun pour extraire les caractéristiques de la forme d'onde audio brute et deux transformateurs encodeurs identiquent utilisés comme des experts une pour l'homme et l'autre pour femme.
Ils ont testé le model sur la base de donnée TIMIT qui a donné une taux d'erreur 5.54 pour les hommes et 6.49 pour les femmes dans le cas de l'estimation de "l'age" par rapport du"Son".considère meilleurs que les modeles existants. mais dans le cas d'estimation d'age par rapport à la taille le taux d'erreur été 7.3 pour les hommes et 6.43 pour les femmes.

Method	Height RMSE		Height MAE		Age RMSE		Age MAE	
	Male	Female	Male	Female	Male	Female	Male	Female
Singh et al	6.7	6.1	5.0	5.0	7.8	8.9	5.5	6.5
Kalluri et al	6.85	6.29	-	-	7.60	8.63	-	-
Kwasny et al	-	-	-	-	7.24	8.12	5.12	5.29
Williams et al	-	-	5.37	5.49	-	-	-	-
Mporas et al	6.8	6.3	5.3	5.1	-	-	-	-
Shangeth et all(single-task model)	8.1	6.0	5.9	4.9	6.96	7.6	4.8	5.1
Shangeth et al (multi-task model)	7.5	6.5	5.8	5.1	6.8	7.4	4.8	5.0
Manav et al(single-task model)	6.92	6.24	5.20	4.95	7.20	7.10	5.04	5.02
Manav et al(multi -task model)	6.95	6.44	5.26	5.15	7.81	8.60	5.50	5.89
Wav2vec2.0bi-encoder	7.3	6.43	5.58	5.07	5.54	6.49	3.96	4.4

TABLE 2.4 – Resultat du wav2vec

20.5 ARTICLE 05 :Oriented Object Detection with Transformer[35]Détection d'objets orientés avec transformateur

les chercheurs dans cet article proposent un modele basée sur l'architecture de transformateur nommé Détection d'objets orientés avec transformateur (O^2DETR),ou'ils utilisent un encodeur transformateur pour coder tous les emplacements d'échelles différentes dans des cartes d'entités multi-échelles en propageant et en agrégeant les informations entre les pixels d'échelles différentes, et des cartes de caractéristiques multi-échelles pour enrichir la présentation visuelle des caractéristiques.
Ils ont testé ce model sur la base de donnée DOTA. les resultats obtenus étaient meilleurs que les modeles traditionel RetinaNet et R-CNN.

Method	backbone	MS.downsample ratios 64	32	16	8	4	epochs	params	mAP
Faster R-CNN	ResNet-50						50	39M	60.32
	ResNet-50	√	√	√	√		50	42M	64.17
	ResNet-50	√	√	√	√	√	50	43M	66.25
	ResNet-101						50	60M	62.44
	ResNet-101	√	√	√	√	√	50	63M	66.03
	ResNet-101	√	√	√	√	√	50	64M	67.71
RetinaNet	ResNet-50						50	34M	58.54
	ResNet-50	√	√	√	√		50	37M	62.78
	ResNet-50	√	√	√	√	√	50	38M	65.77
	ResNet-101						50	55M	60.47
	ResNet-101	√	√	√	√		50	58M	64.11
	ResNet-101	√	√	√	√	√	50	59M	66.53
O^2 DETR	ResNet-50						50	38M	62.22
	ResNet-50	√	√	√	√		50	41M	66.10
	ResNet-50	√	√	√	√	√	50	42M	68.65
	ResNet-101						50	59M	64.32
	ResNet-101	√	√	√	√		50	62M	67.66
	ResNet-101	√	√	√	√	√	50	63M	70.02

TABLE 2.5 – Illustration des résultats de O^2DETR et comparaisons [35]

20.6 ARTICLE 06"Bispectral Pedestrian Detection Augmented with Saliency Maps using Transformer [36]Détection bispectrale des piétons augmentée avec des cartes de saillance à l'aide de Transformer

Dans cet article, les chercheurs proposent une application pour la détection automatique des piétons dans temps réel, Ils ont proposé une nouvelle architecture de fusion basée sur des images bispectrales et augmentée de cartes de saillance à l'aide d'un transformateur, fusionner avec yolo-v3 comme une architecture base.les test ont été éffectués sur la base de données multispectral KAIST dataset , ou'ils ont obtenus des résultats plus efficace par rapport aux entrées simple et par rapport a autre méthode, avec l'avantage de calcul simple important pour les applications en temps réel.

Methods	Day	Night	All
Visible input	64.4	42.3	58.1
Thermal input	62.7	72.6	65.9
Input-fusion	69.2	46.0	62.8
Early-fusion	67.8	42.1	60.4
Halfway fusion-2	68.5	39.2	59.4
Halfway fusion-3	69.6	44.0	62.1
Halfway fusion-4	69.3	49.5	63.4
Late-fusion	67.7	48.8	62.4
Ours	72.6	79.0	**75.8**

TABLE 2.6 – Resultat de tests de detection des piétons

21 Conclusion

Les réseaux convolutifs (CNN) et les transformateurs pour la vision (ViT) sont deux approches importantes pour la vision par ordinateur. Les CNN sont des réseaux spécialisés dans l'analyse de données spatiales, comme les images, et sont souvent utilisés pour la classification et la détection d'objets. Les ViT, quant à eux, sont une méthode plus récente qui utilise des couches de transformateur pour traiter les images comme des séquences de vecteurs, permettant une classification précise avec moins de paramètres.
Ces deux approches ont permis des avancées significatives dans la reconnaissance d'objets et la compréhension des images, et continuent de faire l'objet de recherches pour améliorer leur efficacité et leur fiabilité.

Chapitre 3

Conception

1 Introduction

Le domaine des deepfakes a été grandement enrichi par l'intelligence artificielle. Avec l'aide de l'IA, les images numériques peuvent désormais être rendues incroyablement réalistes en superposant de faux visages sur de vrais, conduisant à des illusions presque homogènes qui sont difficiles à différencier de la réalité.
Notre projet se concentre spécifiquement sur la détection des fausses informations par la detection des faux visages à l'aide des transformateurs de vision, car cette technologie a montré des résultats prometteurs dans l'identification des images manipulées. En tirant parti de la puissance du transformateur de vision (VIT), nous visons à développer une solution capable de détecter avec précision les deepfakes et d'aider à lutter contre la menace croissante de désinformation dans l'espace numérique.

2 Architecture du system

L'architecture détaillée de notre système est présentée dans la figure 3.1 :

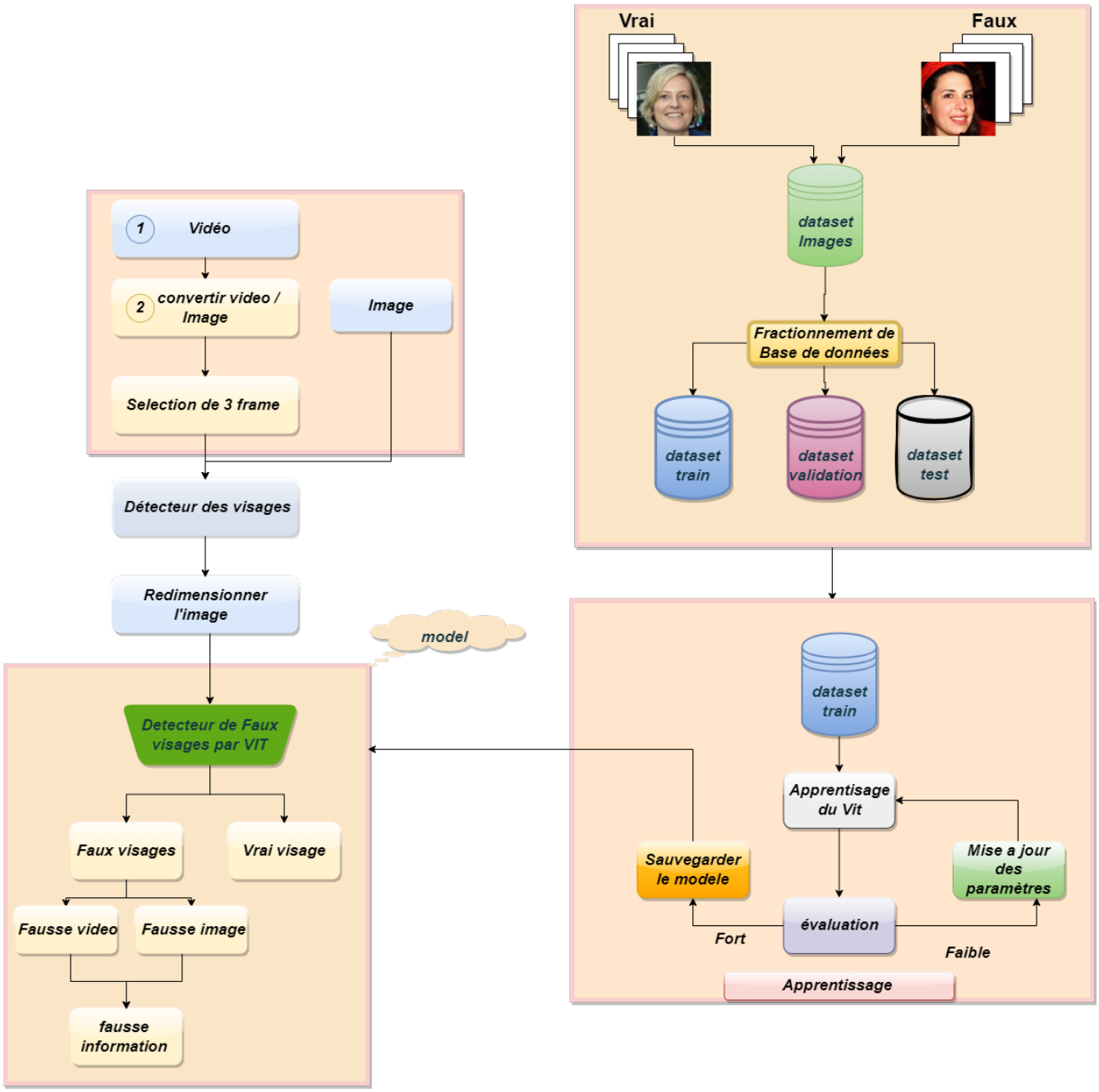


FIGURE 3.1 – Architecture génerale du systeme

2.1 Prétraitement des videos

Les videos téléchargées du youtube sous format mp4 sont convertis en une série d'images(frames) en utilisent le bibliothèque : opencv.

2.2 Bloc détection de visages

Le bloc de detection de visages que nous avons utilisé est basé sur la méthode de classification en cascade de Haar utilisant des fonctionnalités visuelles appelées "Fonctions Haar" et un algorithme d'apprentissage basé sur AdaBoost.

L'avantage de la cascade de classificateurs de Haar est sa capacité à effectuer une détection rapide d'objets en éliminant rapidement les régions d'image non pertinentes à l'aide de niveaux de rejet. Cela économise des ressources de calcul et accélère le processus de détection. Cependant, cette méthode est sensible aux changements d'éclairage, de pose et d'arrière-plan, et nécessite généralement un ensemble de données d'entraînement représentatif pour obtenir de bons résultats.[66]

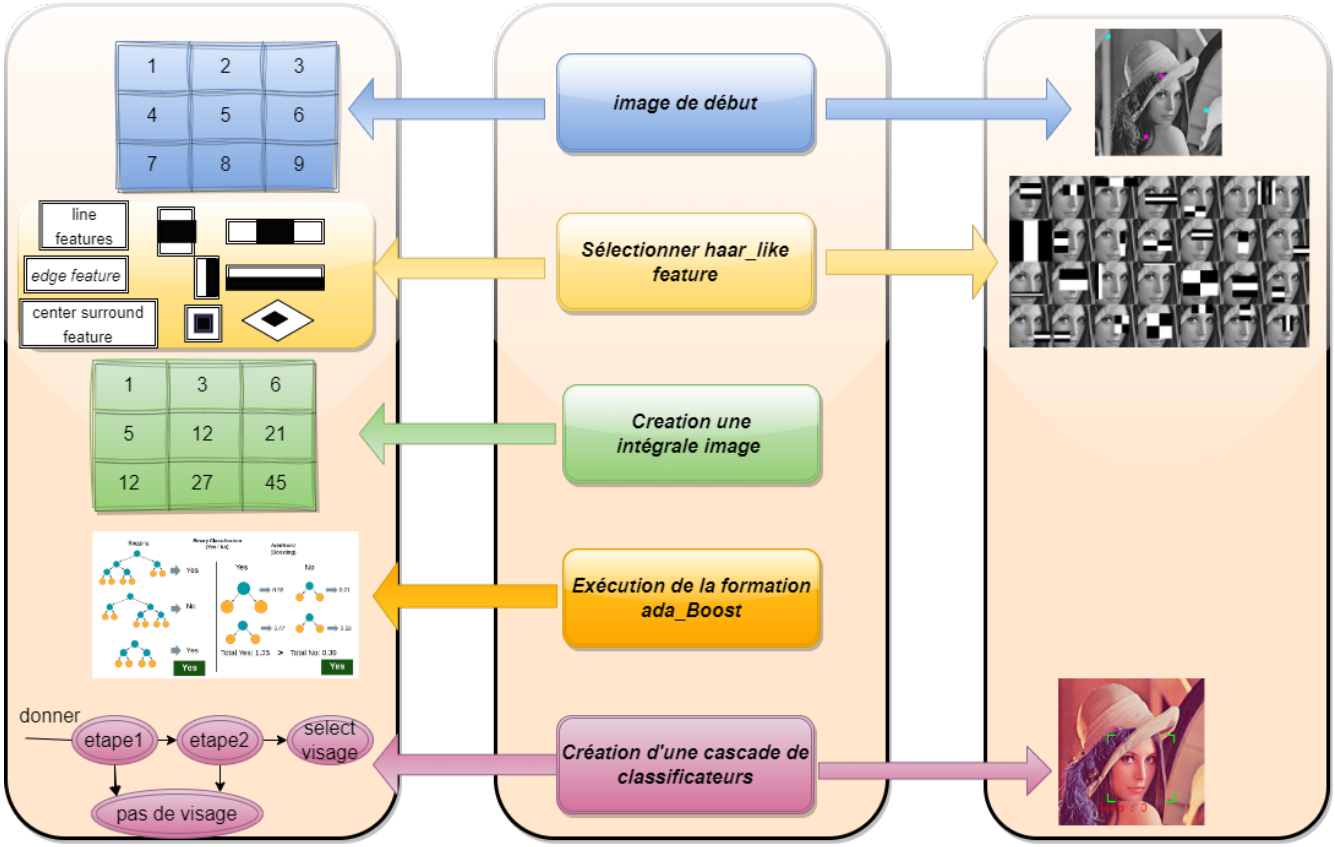


FIGURE 3.2 – Architecture du detecteur de visages utilisé

2.3 Bloc redimentionnement d'image

Afin d'utiliser les images d'entraînement comme entrées pour notre modèle, elles doivent être redimensionnées car elles ont des tailles variables. Nous avons redimensionné toutes les images à une résolution fixe de 200 X 200 pour assurer une dimensionnalité d'entrée cohérente pour notre système.

2.4 Bloc VIT

ViT (Vision Transformer) est un modèle de classification d'images de pointe qui utilise l'architecture de transformateur, développée à l'origine pour les tâches de traitement du langage naturel, pour traiter les images. Il vise à minimiser une perte d'entropie croisée entre les probabilités de classe prédites et les véritables étiquettes de classe. Le modèle comprend les composants suivants[29] :

1. **Incorporations d'entrée (Input embeddings)**

   L'image d'entrée est d'abord divisée en patchs de taille fixe, et chaque patch est intégré linéairement pour obtenir une représentation vectorielle de taille fixe.

2. **Imbrications positionnelles(Positional embeddings)**

   La position de chaque patch est codée à l'aide d'embeddings positionnels appris, qui sont ajoutés aux embeddings de patch pour conserver les informations spatiales.

3. **Encodeur de transformateur(Transformer encoder)**

   Les incorporations de patch, ainsi que les incorporations de position, sont introduites dans un encodeur de transformateur. L'encodeur se compose de plusieurs couches d'auto-attention multi-têtes et de réseaux de neurones à anticipation, qui permettent au modèle de capturer des relations complexes entre les patchs. elle calcule cet relation comme suit :

   — Pour calculer l'attention entre les patchs dans l'architecture du Vision Transformer, on utilise généralement l'attention multi-têtes (multi-head attention) pour capturer les relations spatiales entre les différentes parties de l'image.
   Voici les formules mathématiques pour calculer l'attention entre les patchs :

$$attention = (\frac{softmax(Q * K^t)}{\sqrt{D_k}}) * V \tag{3.1}$$

   Q : les vecteurs requêtes.
   K : les vecteurs clés.
   V : les vecteurs valeurs.
   D_k : dimensions après projection.

   ou

$$Q = X * W_q \tag{3.2}$$
$$K = X * W_k \tag{3.3}$$
$$V = X * W_v \tag{3.4}$$

   W_q, W_k, W_v : les matrices de poids.
   X :matrice de patch.

   — Les connexions résiduelles résolvent le problème de pert d'information en ajoutant des chemins directs (skip connections) qui permettent aux informations de contourner une ou plusieurs couches. Plus précisément, au lieu de modifier directement les sorties d'une couche pour les entrées de la couche suivante, les connexions résiduelles ajoutent les sorties de la couche précédente aux entrées de la couche suivante. Cela crée une "route" alternative pour les informations à travers le réseau.
   en suite on calcul :

$$(x) : y = F(x) + x \tag{3.5}$$

ou
F(x) : est le résultat de la couche transformée.
x : est l'entrée originale non transformée.

— La normalisation est utilisée pour stabiliser et accélérer l'apprentissage d'un réseau de neurones. Elle vise à réduire les effets indésirables tels que le déséquilibre de l'échelle des activations et le problème de dégradation qui peut survenir lorsque le réseau devient plus profond. La formule générale de la normalisation est la suivante :

$$x' = \frac{x - \mu}{\sigma} \tag{3.6}$$

x : est la valeur d'entrée à normaliser.
μ est la moyenne des valeurs de x dans l'ensemble des données.
σ est l'écart type des valeurs de x dans l'ensemble des données.

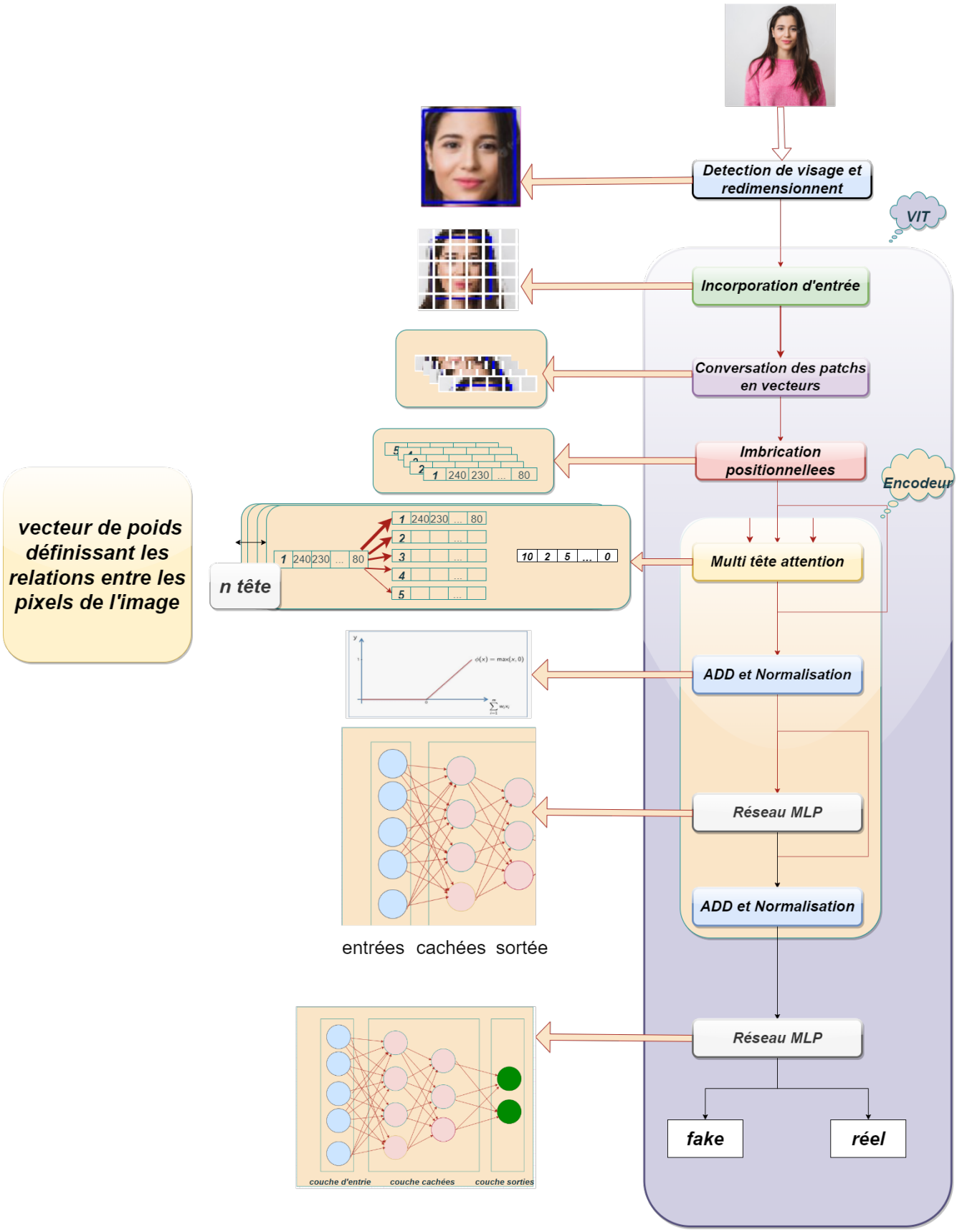


FIGURE 3.3 – Architecture détaillée du VIT

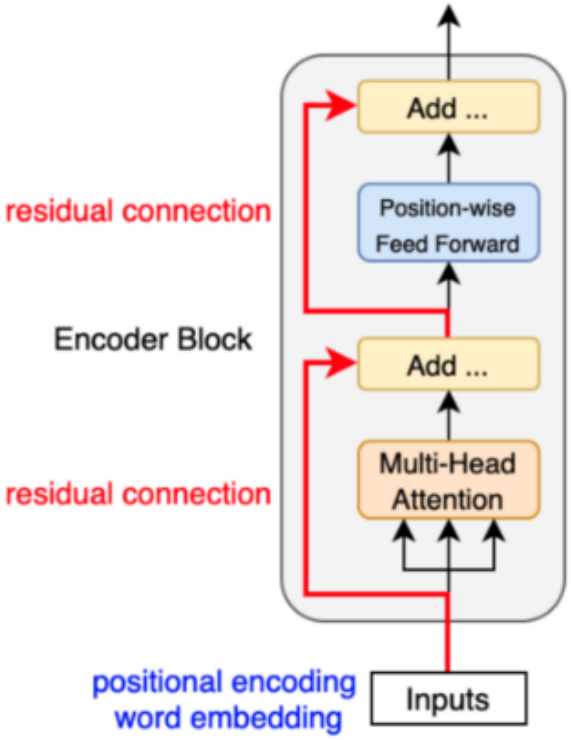


FIGURE 3.4 – Architecture de l'encodeur du transformareur [57]

4. **Tête de classification(Classification head)**
   La tête de classification est la partie du modèle de réseau neuronal qui est chargée de prendre les caractéristiques extraites par les premières couches du modèle et de les utiliser pour faire des prédictions pour différentes catégories du problème de classification.
   Dans le contexte du modèle Vision Transformer, la tête de classification se compose généralement de plusieurs couches entièrement connectées qui transforment les caractéristiques extraites en probabilités pour chaque classe. Ces probabilités sont calculées à l'aide d'une fonction d'activation, généralement une fonction softmax, qui normalise les valeurs de sortie pour représenter des probabilités valides. elle calcule la probabiliter par :

$$p(i) = \exp(z(i))/\sum(\exp(z(j)))pour j = 1\text{à}N \tag{3.7}$$

   Ou :
   z(i) : représente la sortie (ou l'activation) associée à la classe i. Cette valeur est généralement obtenue à partir des caractéristiques extraites par les couches précédentes du modèle.

3 Configuration du modèle

La configuration du modèle est illustrée par la figure 3.5 :

```python
def ViT(cf):
    """ Inputs """
    input_shape = (cf["num_patches"], cf["patch_size"]*cf["patch_size"]*cf["num_channels"])
    inputs = Input(input_shape)     ## (None, 256, 3072)

    """ Patch + Position Embeddings """
    patch_embed = Dense(cf["hidden_dim"])(inputs)   ## (None, 256, 768)

    positions = tf.range(start=0, limit=cf["num_patches"], delta=1)
    pos_embed = Embedding(input_dim=cf["num_patches"], output_dim=cf["hidden_dim"])(positions) ## (256, 768)
    embed = patch_embed + pos_embed ## (None, 256, 768)

    """ Adding Class Token """
    token = ClassToken()(embed)
    x = Concatenate(axis=1)([token, embed]) ## (None, 257, 768)

    for _ in range(cf["num_layers"]):
        x = transformer_encoder(x, cf)

    """ Classification Head """
    x = LayerNormalization()(x)     ## (None, 257, 768)
    x = x[:, 0, :]
    x = Dense(cf["num_classes"], activation="softmax")(x)
    model = Model(inputs, x)
    return model
```

FIGURE 3.5 – Configuration du model VIT

3.1 Input shape

Dans cette partie de code en vas extraire le forme d'une image qui prend le nombre de patch dans une image et la taille de chaque patch et le nombre de canals(colour utiliser dans l'image)

3.2 Couche d'entrée

A une dimension de 200*200*3, où 3 représente le nombre de matrices de couleurs (rouge, vert et bleu).

3.3 Couche patchs et la encoded patchs

Ce module est responsable de la divisions de l'image donnée en patches de tailles equivallentes a 32 X 32 px et donner a chaque patch un positionnement par rapport a l'image initiale.

1. **La couche patchs**

   Dans cet couche nous définissons en entrée un lot d'images sous la forme d'un tenseur 4D de forme [batch_size, height, width, channels],tel que :
   — **batch_size** : le nombre d'image dans chaque groupe.

— **height & width** : les dimensions des images.
— **channels** : représente le nombre de matrices de couleurs.
En va diviser l'image en petites images de 32 pixels sans les superposer.
elle donne comme sortie un tenseur 3D de forme [batch_size,num_patches, patch_dims] où :
— **batch_size** le nombre d'image dans chaque groupe.
— **num_patchs** est le nombre de patchs extraits de chaque image.
— **patch_dims** est le nombre d'éléments dans chaque patch.

2. **Couche encoded patchs**
Dans Cette couche on prend deux paramètre :
num_patchs qui designe le nombre de patch extrait de l'image principale.
hidden_dim : est la dimension de l'espace de projection dans lequel les patches seront projetés.

3.4 Couche ClassToken

Cet couche est utilisé pour définir les poid de facon aléatoire et entrainable (elle est modifiable dans chaque itération d'entrainnement d'une model)

3.5 Couche normalization et couche résiduelle connection(Add)

La couche résiduel connection utilisée pour réaliser une connexion de saut (skip connection) entre la sortie de deux couche comme la couche MultiHeadAttention (attention_output) et la couche inputs(positional encoding), ou la couche MLP et les résultat de couche qui précède elle (réséduel encoding), en utilisant une opération d'addition.
on applique la normalisation sur les résultat de couche réséduelle connection avant de les transmettre à la couche suivante pour éviter les valeur négative.

3.6 La couche transformer encoder

Cette couche represents un bloc codeur dans un modèle de transformateur. elle prend la configuration du model et les donnée utilisées comme paramètre .

3.7 La couche MLP(Perceptron multicouche)

Le perceptron multicouche est constitué de plusieurs couches de neurones. La première couche est la couche d'entrée, qui reçoit les données d'entrée avec taille égale a 3072 neurones. La couche cachée, effectuent des transformations non linéaires des données d'entrée avec taille égale a 768 neurones. Enfin, la couche de sortie génère les prédictions ou les résultats du modèle égale a 2 neurones.
hidden_dim définit le nombre de neurones cachés dans chaque couche.
dropout_rate définit le taux de dropout, qui est la probabilité que chaque neurone soit désactivé pendant l'entraînement pour éviter le surapprentissage.

4 Configuration détaillée du VIT

Sur la table 3.1 on donne la configuration détaillée du model utilisé.

Layer	Output Shape	Param
input	[(None, 256, 3072)]	0
couche patches et patches codés	(None, 256, 768)	2360064
class_token (ClassToken)	(N one, 1, 768)	768
concatenate (Concatenate)	(None, 257, 768)	0
layer_normalization	(None, 257, 768)	1536
multi_head_attention	(None, 257, 768)	28339968
add (Add)	(None, 257, 768)	0
dense(Dense)	(None, 257, 768)	2360064
dropout (Dropout)	(None, 257, 768)	0
mlp	(None, 257, 768)	0
dense_25	(None, 2)	1538

TABLE 3.1 – configuration du ViT

5 Conclusion

Le système que nous proposons pour la détection de fausses informations par la recherche des faux visages repose en grande partie sur la formation du réseau VIT dont les performances dépendent principalement du calcul de l'attention entre les différents patchs de l'image.
Lors de la phase de construction d'attention entre les différents patch d'image, il est indispensable de fixer les seuils des paramètres des fonctions loss et accuracy , que le système utilisera ensuite pour décider si une image arbitraire appartient à la classe 'faux visage' ou "vrai visage".

Chapitre 4

Implementation

1 Introduction

L'architecture de notre système est fixée ; nous procédons dans ce chapitre aux détails concernant le matériel et les logiciels utilisés pour sa mise en œuvre. Il est indispensable de valider notre projet en évaluant le système mis en place après une formation de 2 mois, les tests ont été réalisés sur des images est des vidéo contenant de fausses et de réelles informations.

2 Environnement

2.1 Matériel

Station de calcul(Computing station

l'apprentissage de notre modèle a été réalisé dans une machine à calcul haute performance (HPC) du LAIG (laboratoire d'automatique et informatique de guelma). avec une édition du système Windows 10 2021 et une RAM égale à 32 Go, le processeur est un processeur intel (r) core TMi7 et Nvidia Gpu

2.2 Logiciel

2.3 Python[37]

Python est un langage de programmation largement utilisé avec des fonctionnalités de haut niveau, interprétées et orientées objet. Sa grande communauté de développeurs et de programmeurs en fait un choix populaire. Python est connu pour sa simplicité et sa facilité d'apprentissage. De plus, la bibliothèque Python est accessible sur la plupart des plates-formes et peut être redistribuée gratuitement.

2.4 PyCharm[W11]

PyCharm est un environnement de développement intégré utilisé pour programmer en Python2.
Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.
Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache.

2.5 Anaconda[W11]

Dans Anaconda, l'éditeur de texte intégré par défaut est appelé "Spyder". Spyder est un environnement de développement intégré (IDE) spécialement conçu pour les scientifiques des données et les chercheurs utilisant Python. Il offre des fonctionnalités avancées telles qu'un éditeur de code, une console IPython, un explorateur de variables, un gestionnaire de fichiers et bien d'autres.

Tensorflow[67]

TensorFlow est une bibliothèque de logiciels open source pour le flux de données et la programmation différentiable sur une gamme de tâches. Il est développé par l'équipe Google Brain et est largement utilisé pour la recherche et les applications d'apprentissage automatique et d'apprentissage profond. TensorFlow permet aux développeurs de créer et de former des réseaux de neurones profonds, y compris des réseaux de neurones convolutifs (CNN), des réseaux de neurones récurrents (RNN) et d'autres types de modèles.
TensorFlow fonctionne en construisant un graphique de calcul qui décrit les opérations mathématiques à effectuer sur les données. Ce graphique est ensuite exécuté sur un dispositif informatique, tel qu'un CPU ou un GPU, pour produire la sortie souhaitée. TensorFlow fournit également une API de haut niveau appelée Keras, qui facilite la création et la formation de réseaux de neurones.
TensorFlow est devenu un outil populaire dans le domaine de l'intelligence artificielle, utilisé pour un large éventail d'applications, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel et la robotique. Il est également utilisé dans la recherche universitaire et les applications commerciales.

Keras[68]

La bibliothèque Keras est une bibliothèque Python de haut niveau pour l'apprentissage en profondeur, compacte et facile à apprendre, conçue pour fonctionner sur TensorFlow. Elle offre une collection de blocs de construction puissants et abstraits pour la construction de modèles d'apprentissage profond, ce qui simplifie le processus de développement et permet aux développeurs de se concentrer sur les concepts fondamentaux de l'apprentissage en profondeur. Avec Keras, les utilisateurs peuvent créer des applications sophistiquées d'apprentissage en profondeur sans avoir à interagir directement avec les complexités de TensorFlow.

PIL[69]

PIL signifie Python Imaging Library, et c'est une bibliothèque dans le langage de programmation Python qui prend en charge l'ouverture, la manipulation et l'enregistrement de nombreux formats de fichiers image différents. PIL permet aux développeurs d'effectuer diverses tâches de traitement d'images telles que le redimensionnement, le recadrage, le filtrage et la transformation d'images.

CV2[70]

cv2 est l'abréviation d'OpenCV (Open Source Computer Vision Library) version 2, une bibliothèque de logiciels de vision par ordinateur et d'apprentissage automatique open source populaire qui est principalement utilisée pour la vision par ordinateur et le traitement d'images en temps réel. Il fournit un large éventail de fonctions et d'algorithmes pouvant être utilisés pour effectuer diverses tâches, telles que la détection d'objets, la reconnaissance faciale, le traitement d'images et de vidéos, etc. OpenCV est écrit en C++ et possède des liaisons pour divers langages de programmation, notamment Python, Java et MATLAB.

Matplotlib[71]

Matplotlib est une bibliothèque Python spécialisée dans la création de graphiques en 2D de qualité professionnelle. Elle permet le tracé interactif ou non interactif et offre la possibilité de sauvegarder les images dans différents formats de sortie (PNG, PS, etc.). Elle est compatible avec plusieurs boîtes à outils de fenêtres (GTK+, wxWidgets, Qt, etc.) et propose une large gamme de types de graphiques (lignes, barres, camemberts, histogrammes et bien d'autres). En outre, elle est facilement personnalisable, souple et simple d'utilisation.

Tkinter [W12]

Tkinter est une bibliothèque standard de Python pour la création d'interfaces graphiques utilisateur (GUI). Elle permet aux développeurs de créer des fenêtres, des boutons, des listes, des boîtes de dialogue et d'autres éléments d'interface utilisateur en utilisant des widgets pré-construits. Tkinter est basé sur Tk, une bibliothèque de création d'interfaces graphiques en langage Tcl/Tk. Tkinter est inclus dans la distribution standard de Python et est disponible pour les systèmes d'exploitation Unix, Windows et Macintosh.

3 Base de données

Nous avons utilisé un ensemble de données nommées '140k Real and Fake Faces' [W13] de 140000 images couleur contenant 70000 images réelles de l'ensemble de données flickre collectées par Nvidia, ainsi que 70000 faux visages échantillonnés à partir des 1 million de visages FAUX (générés par StyleGAN) fournis par Bojan. Toutes les images fausses et réelles que nous avons utilisées dans notre projet ont été redimensionnées à 200 X 200 px avant de les soumettre en réseau.

1. **Fractionnement de la base de données**

   L'entrainement de notre système à été effectué sur 12010 images de la base.'140k Real and Fake Faces' fractionner en trois parties :
   • Les images d'entraînement (training set), seront utilisées pour entraîner le réseau. Cela représente 9610 images de visages.
   • Les images de test (test set), sont utilisées pour évaluer la progression de notre modèle. Ils présentent 1200 Images réparties entre 600 vraies images et 600 fausses images.
   • Les images de validation (validation set), présentent 1080 images réparties entre 540 images réelles et 540 images fausses.

	train	validation	test
real	4805	600	600
fake	4805	600	600

TABLE 4.1 – Fractionnent de la base de donner

FIGURE 4.1 – Exemples de faux et vrais visages de'140k Real and Fake Faces'

4 Formation et Test

Lors de la phase d'apprentissage, nous expérimentons plusieurs configurations en modifiant les paramètres du réseau comme l'optimiseur, le nombre d'itérations, le nombre d'époques, et la taille du lot.
Pour former notre réseau, nous avons choisi la configuration suivante :

— Optimiseur : "Adam".
— Taille du lot : 32.
— Nombre d'époques : 10.

— La fonction de Loss utilisée : "categorical_crossentropy".
— Toutes les couches denses utilisent la fonction "GeLu" sauf la couche de sortie utilise la fonction "Softmax".

L'évaluation d'apprentissage est représenté sur les figures 4.2 et 4.3 par les courbes de précision(accuracy) et de pertes(Loss) obtenues sur les données d'apprentissage et de validation. Sur ces courbes nous constatons que la perte minimale obtenue aprés 10 'époch' est de 59% pour les données d'entrainement (train dataset) et de 57% pour les données de validation (validation dataset).
Les précisions obtenues lors de l'entrainement et la validation sont respectivement 69% et 68%.considirées comme acceptable et suffisanes pour rependre a l'objectif de notre application.

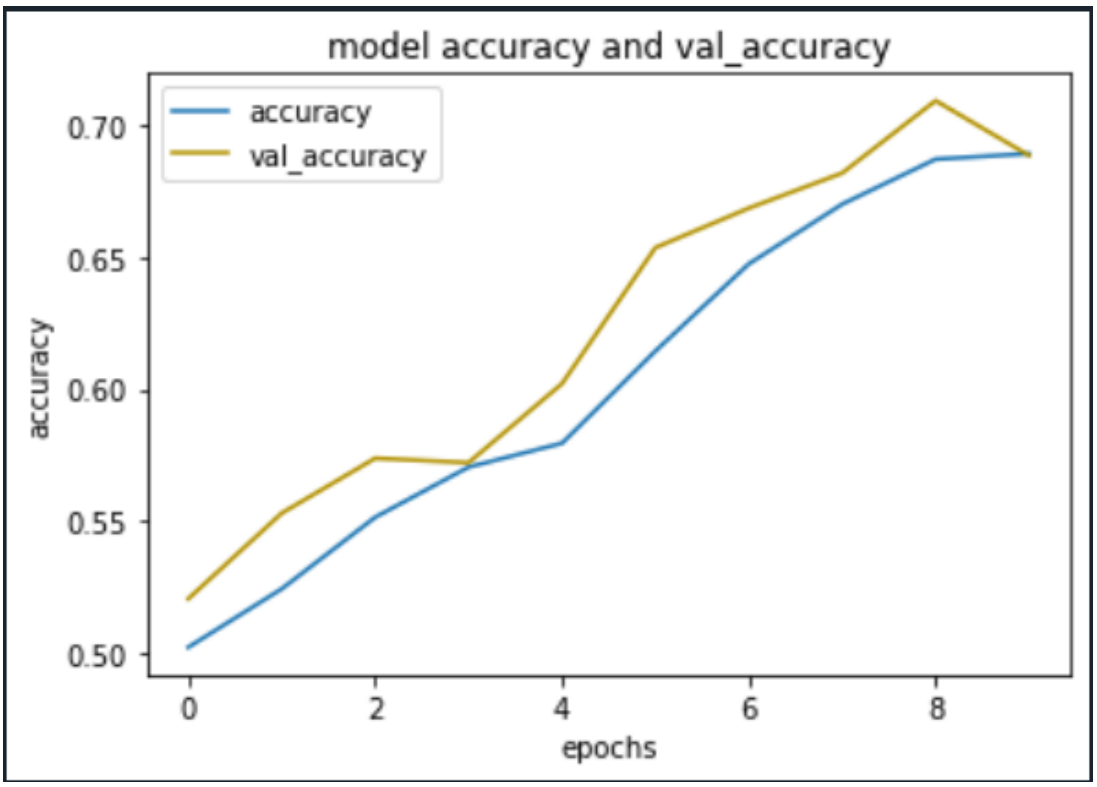


FIGURE 4.2 – Train et validation accuracy

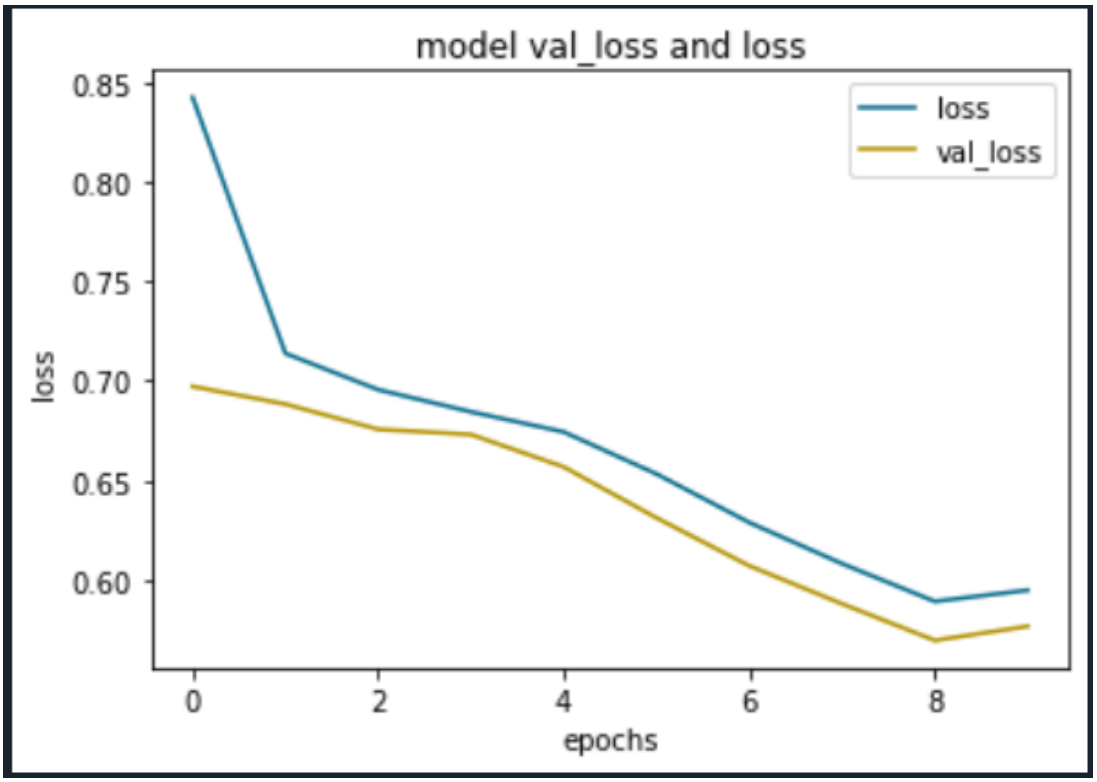


FIGURE 4.3 – Train et validation Loss

Les tests effectués sur la base de test de 10000 images(fake,real) ont donnés un taux de reconnaissance de 60.47 % sur les deux classes confandus,considéré comme accéptable vu la complexité des classes Indiscernables à l'oeil nu.

5 Interface graphique

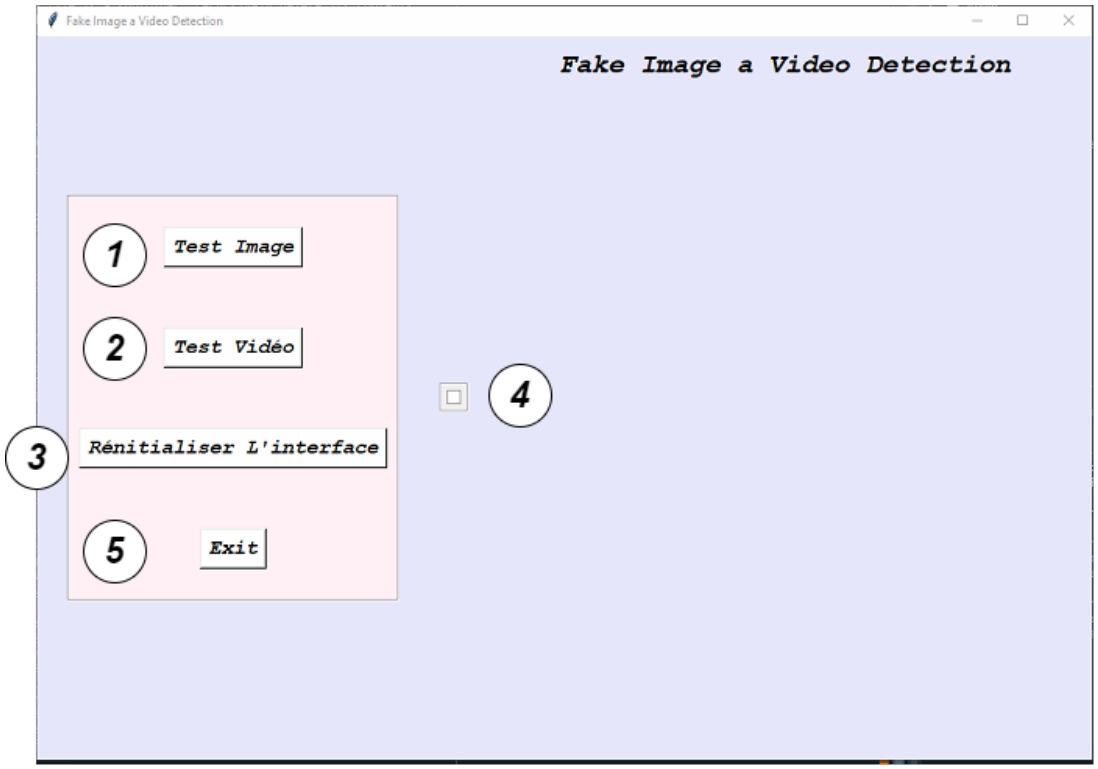


FIGURE 4.4 – interface grafique de l'application

1. Ce bouton permet de sélectionner un fichier image. Une fois sélectionné, le programme effectuera une détection de visage sur l'image suivie d'une classification pour déterminer si le visage est réel ou faux.
2. Ce bouton vous permet de sélectionner un fichier vidéo. Une fois sélectionné, le programme extraira 3 images de la vidéo. Ces images subiront une détection de visage, suivie d'une classification pour déterminer si les visages sont réels ou faux.
3. Ce bouton est utilisé pour réinitialiser la fenêtre et effacer tous les paramètres ou réglages qui ont été utilisés.
4. Cette partie et pour afficher les image initiale et les image après détection et classification
5. Bouton pour fermer le programme

6 Test et Interprétations

L'objectif principal étant de detecter la fausses information sur les images et sur les vidéos nous avons donc éffectués des tests pour detecter les fausses images(Fake Images) et les fausses vidéos (Fake videos).

6.1 Test sur les images

La table 3.2 illustre quelques tests effectués sur des images fausses et réelles où la majorité des images ont été reconnues correctement(voir figure 1 a 5).
Des erreurs de détections ont été également signalés dans quelques images testées tel que sur les images n : 6,7,8 .
Il faut noter que si le visage est detecté correctement faux, ceci implique que l'image est fausse et donc l'information portée par cette image est également fausse.
D'une autre part si le visage est detecté correctement réel cela n'implique pas obligatoirement que l'image est réelle et l'information aussi car la modification de l'image peut être au niveau de l'arrière plan et c'est souvent le cas dans les images diffusées sur internet et sur les sites sociaux.

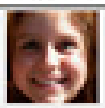
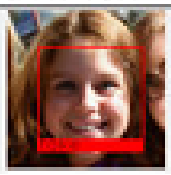
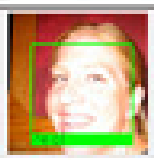
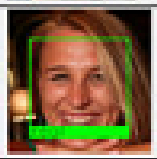
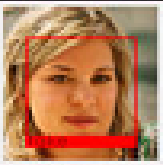
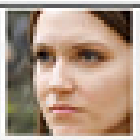
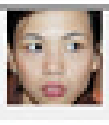
N	Image	Détection	Décision	Remarque
1	 Faux visage			Faux visage →Fausse Information
2	 Vrai visage			Vrai visage
3	 Faux visage			Faux visage →Fausse Information
4	 Vrai visage			Vrai visage
5	 Faux visage			Faux visage →Fausse Information
6	 Vrai visage		 Fausse visage	Erreur de détection
7	 Faux visage		 Vrai visage	Erreur de détection
8				Erreur de détection

TABLE 4.2 – Resultat de test image

6.2 Test sur les videos

La table 3.3 illustre des tests effectués sur des vidéos réelles et des vidéos fausses oúle système procède a un test sur les 3 premières frames et la decision finale est prise sur les résultats majoritaire obtenus sur les trois frames.

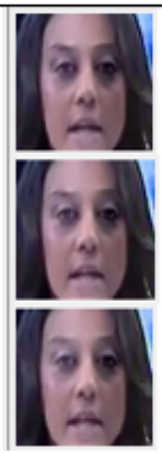
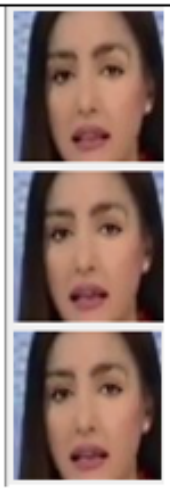
Vidéo	Frame test	Détection	Décision
Vidéo1 005_010.mp4 (Fausse)		 Fausse	Fausse visage → Fausse information
Vidéo2 012_026.mp4 (Fausse)		 Fausse	Fausse visage → Fausse information
Vidéo3 009.mp4 (Real)		 Vrai visage	Vrai visage

TABLE 4.3 – Resultat de test video

7 Conclusion

Les résultats de notre système sont prometteurs, et les tests de détection de faux visages dans le domaine de la détection deepfake sont encourageants. Les performances de ce système peuvent être optimisées en prolongeant la phase d'apprentissage avec un autre type de transformateur visuel ou une combinison d'un réseau transformateur avec un CNN.

Conclusion génerale

La détection des fausses informations (fake news) est un défi majeur dans le contexte de la propagation rapide de l'information sur les plateformes en ligne. L'utilisation de l'architecture de transformateur visuel peut offrir une approche prometteuse pour aborder ce problème.

L'architecture de transformateur visuel combine les avantages des réseaux de neurones Transformers et de la vision par ordinateur pour analyser les images et les textes associés afin de détecter les fausses informations. Cependant, il convient de noter que la détection des fausses informations est un problème complexe et évolutif.

L'architecture de transformateur visuel peut être un outil puissant, mais elle ne constitue pas une solution définitive.

Notre recherche a démontré avec succès l'efficacité du modèle VIT pour la détection des faux visages sur les images et vidéos. nous avons obtenu des résultat d'une précision et efficacité considirable qais qui peuvent être améliorer dans le future en suivant les démarches suivantes :

- Utilisez d'autres modèles de transformateurs de vision.
- Permettre aux réseaux de s'entraîner plus longtemps.
- Utiliser un ensemble de données d'entraînement plus volumineux.

Bibliographie

[1] Mathieu-Robert Sauvé. Les fake news dans les médias du québec : perceptions des journalistes. *Mémoire présenté en vue de l'obtention du grade de maîtrise en communication), Québec (Canada) : Faculté des lettres et sciences humaines, Université de Sherbrooke*, page 15, 2019.

[2] David Boyle. Authenticity : Brands, fakes, spin and the lust for real life. 2004.

[3] Juan Cao, Peng Qi, Qiang Sheng, Tianyun Yang, Junbo Guo, and Jintao Li. Exploring the role of visual content in fake news detection. *Disinformation, Misinformation, and Fake News in Social Media : Emerging Research Challenges and Opportunities*, pages 141–161, 2020.

[4] Stéphane Foucart. *La fabrique du mensonge. Comment les industriels manipulent la science et nous mettent en danger*. Editions Gallimard, 2014.

[5] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Odyssey*, pages 132–137, 2020.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11) :139–144, 2020.

[7] Fouad Jabiri. *Applications de méthodes de classification non supervisées à la détection d'anomalies*. PhD thesis, Université Laval, 2020.

[8] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF : A Large-scale Challenging Dataset for DeepFake Forensics. In *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, Seattle, WA, United States, 2020.

[9] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics : A large-scale video dataset for forgery detection in human faces. *arXiv*, 2018.

[10] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++ : Learning to detect manipulated facial images. In *International Conference on Computer Vision (ICCV)*, 2019.

[11] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it : Face analysis in the wild using synthetic data alone, 2021.

[12] Ben Pflaum Nicole Baram Cristian Canton Ferrer Brian Dolhansky, Russ Howes. The deepfake detection challenge (dfdc) preview dataset, 2019.

[13] Prabhat Kumar, Mayank Vatsa, and Richa Singh. Detecting face2face facial reenactment in videos. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2589–2597, 2020.

[14] Deressa Wodajo and Solomon Atnafu. Deepfake video detection using convolutional vision transformer. *arXiv preprint arXiv :2102.11126*, 2021.

[15] Vurimi Veera Venkata Naga Sai Vamsi, Sukanya S Shet, Sodum Sai Mohan Reddy, Sharon S Rose, Sona R Shetty, S Sathvika, MS Supriya, and Sahana P Shankar. Deepfake detection in digital media forensics. *Global Transitions Proceedings*, 3(1) :74–79, 2022.

[16] Davide Alessandro Coccomini, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. Combining efficientnet and vision transformers for video deepfake detection. In *Image Analysis and Processing–ICIAP 2022 : 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III*, pages 219–229. Springer, 2022.

[17] Marie-Josée Roch. Les conceptions de l'apprentissage chez les futur (e) s enseignant (e) s. 2016.

[18] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[20] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer, 2009.

[21] Valentin Bisson. Algorithmes d'apprentissage pour la recommandation. 2013.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553) :436–444, 2015.

[23] Ibrahem Kandel and Mauro Castelli. Transfer learning with convolutional neural networks for diabetic retinopathy image classification. a review. *Applied Sciences*, 10(6) :2021, 2020.

[24] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[25] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop : A review of bayesian optimization. *Proceedings of the IEEE*, 104(1) :148–175, 2015.

[26] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106 :249–259, 2018.

[27] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[28] Liya Wang and Alex Tien. Remote sensing scene classification with masked image modeling (mim). *arXiv preprint arXiv :2302.14256*, 2023.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*, 2020.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit : A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[32] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt : Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[33] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt : Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022.

[34] Tarun Gupta, Duc-Tuan Truong, Tran The Anh, and Chng Eng Siong. Estimation of speaker age and height from speech signal using bi-encoder transformer mixture model. *arXiv preprint arXiv :2203.11774*, 2022.

[35] Teli Ma, Mingyuan Mao, Honghui Zheng, Peng Gao, Xiaodi Wang, Shumin Han, Errui Ding, Baochang Zhang, and David Doermann. Oriented object detection with transformer. *arXiv preprint arXiv :2106.03146*, 2021.

[36] Anis Sahbani, Mohamed Amine Marnissi, Ikram Hattab, Hajer Fradi, and Najoua Essoukri Ben Amara. Bispectral pedestrian detection augmented with saliency maps using transformer. 2022.

[37] Guido Van Rossum and Fred L Drake. *An introduction to Python*. Network Theory Ltd. Bristol, 2003.

[38] Karen Santos-d'Amorim and Májory Miranda. Misinformation, disinformation, and malinformation : clarifying the definitions and examples in disinfodemic times. *Encontros Bibli Revista Eletrônica de Biblioteconomia e Ciência da Informação*, 03 2021.

[39] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi : Exposing ai generated fake face videos by detecting eye blinking. *ArXiv*, abs/1806.02877, 2018.

[40] Furkan Lüleci, Necati Catbas, and Onur Avci. A literature review : Generative adversarial networks for civil structural health monitoring. *Frontiers in Built Environment*, 11 2022.

[41] Maha Alkhayrat, Mohamad Aljnidi, and Kadan Aljoumaa. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and pca. *Journal of Big Data*, 7 :9, 02 2020.

[42] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics : A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv :1803.09179*, 2018.

[43] Liwei Deng, Hongfei Suo, and Dongjie Li. Deepfake video detection based on efficientnet-v2 network. *Computational Intelligence and Neuroscience*, 2022 :1–13, 04 2022.

[44] Qiushi Sun, Jingtao Guo, and Yi Liu. Face image synthesis from facial parts. *EURASIP Journal on Image and Video Processing*, 2022, 05 2022.

[45] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv :1910.08854*, 2019.

[46] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and I. Echizen. Mesonet : a compact facial video forgery detection network, 09 2018.

[47] Pantelis Kaplanoglou. *Content-Based Image Retrieval using Deep Learning.* PhD thesis, 06 2017.

[48] Ramadan Ramo. Detection and diagnosis of skin diseases by using snake algorithm and neural networks. *Technium : Romanian Journal of Applied Sciences and Technology*, 4 :104–114, 12 2022.

[49] Vijaya Kumar .R, B. Rao, and K. Raju. Handwritten hindi digits recognition using convolutional neural network with rmsprop optimization. pages 45–51, 06 2018.

[50] Canlin Zhang, Daniel Bis, Xiuwen Liu, and Zhe He. Biomedical word sense disambiguation with bidirectional long short-term memory and attention-based neural networks. *BMC Bioinformatics*, 20 :502, 12 2019.

[51] Zheying Huang, Pei Wang, Jian Wang, Haoran Miao, Ji Xu, and Pengyuan Zhang. Improving transformer based end-to-end code-switching speech recognition using language identification. *Applied Sciences*, 11(19) :9106, 2021.

[52] Qingtian Ke and Peng Zhang. Hybrid-transcd : A hybrid transformer remote sensing image change detection network via token aggregation. *ISPRS International Journal of Geo-Information*, 11 :263, 04 2022.

[53] Salim Heddam, Abdelmalek Bermad, and Noureddine Dechemi. Modélisation de la dose de coagulant par les systèmes à base d'inférence floue (anfis) application à la station de traitement des eaux de boudouaou (algérie). *Revue des sciences de l'eau*, 25(1) :1–17, 2012.

[54] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit : A video vision transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, 2021.

[55] Paolo Bruno, Roberto Amoroso, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Investigating bidimensional downsampling in vision transformer models. In *International Conference on Image Analysis and Processing*, pages 287–299. Springer, 2022.

[56] Mathilde IRSTEA, Marc Isenmann, Thomas Sanz, and Sandra Luque. Prédiction de la distribution d'alliances de végétation des milieux ouverts d'altitude à l'aide de l'approche dite du maximum d'entropie, 10 2012.

[57] Hamidreza Ghader. An empirical analysis of phrase-based and neural machine translation, 03 2021.

[58] Inderpal Singh and Shailey Singh. The hype machine : How social media disrupts our elections, our economy and our health-and how we must adapt, 2021.

[59] Alexis Kochel. Misinformation variation ? looking through the gendered lens. 2023.

[60] Sami Khoury. Évaluation des cybermenaces nationales 2023-2024.

[61] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. Asvspoof 2019 : Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv :1904.05441*, 2019.

[62] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[63] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1) :5485–5551, 2020.

[64] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv :1904.10509*, 2019.

[65] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv :1703.03130*, 2017.

[66] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57 :137–154, 2004.

[67] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow : Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv :1603.04467*, 2016.

[68] Navin Kumar Manaswi, Navin Kumar Manaswi, and Suresh John. *Deep learning with applications using python*. Springer, 2018.

[69] Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, and Valentino Zocca. *Python Deep Learning : Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Packt Publishing Ltd, 2019.

[70] Gary Bradski. The opencv library. *Dr. Dobb's Journal : Software Tools for the Professional Programmer*, 25(11) :120–123, 2000.

[71] Sandro Tosi. *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.

Webographie

[W1] definition de fake news

https://www.ionos.fr/digitalguide/web-marketing/les-media-sociaux/que-sont-les-fake-news/

[W2] definition de news \url{https://dictionary.cambridge.org/fr/dictionnaire/anglais/news

[W3] définition de misinformation et désinformation

https://www.dictionary.com/e/misinformation-vs-disinformation-get-informed-on-the-difference

[W4] définition d'apprentissage par transfert

https://mobiskill.fr/blog/conseils-emploi-tech/quest-ce-que-lapprentissage-par-transfert-transfer-learning/

[W5] définition de réseau classique

https://www.analyticsvidhya.com/blog/2021/07/understanding-the-basics-of-artificial-neural-network-ann/

[W6] Couche convolutive

https://blent.ai/cnn-comment-ca-marche/

[W7] réseau CNN

https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels

[W8] la couche entièrement connectée ou Full Connecte
https://datascientest.com/convolutional-neural-network

[W9] la déférence entre ANN et CNN

https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/

[W10] \url{https://hashdork.com/fr/attention-mechanism-in-deep-learning

[W11] pycharm éditeur

https://blog.jetbrains.com/pycharm/2023/04/2023-1-1/

[W12] bibliothèque tkinter

https://docs.python.org/fr/3/library/tkinter.html

[W13] Dataset utilisée https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces