الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة

Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا

Faculté de Technologie

قسم الإلكترونيك

Département d'Électronique

Mémoire de Master

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Présenté par

BENMESBAH SARAH

&

CHEKNOUN ANFEL

Analyse des scènes en vidéosurveillance : – Détection de violence –

Promoteur : Mr. KABIR Yacine

President: Mr. Guessoum Abderrazak

Examinateur : Mr. NAMANE Abderrahmane

Année Universitaire 2023-2024.

Remerciements

Nous remercions Dieu de nous avoir donné la force et la santé pour pouvoir achever ce projet.

Nous exprimons notre profonde gratitude à notre encadrant, M. Kebir Yacine, pour avoir supervisé ce mémoire. Sous sa bienveillante direction, nous avons eu l'honneur de mener à bien ce travail. Nous lui adressons nos plus vifs remerciements pour sa gentillesse, sa spontanéité, sa disponibilité et ses précieux conseils qui nous ont permis d'atteindre notre objectif. Sa confiance en nous a été un moteur de motivation, et nous tenons à lui exprimer notre reconnaissance pour avoir contribué à la réussite de cette recherche.

Nous remercions également les membres du jury, M. Namane Abderrahmane et M. Guessoum Abderrazak, pour avoir accepté d'évaluer notre travail.

Nos sincères remerciements vont également à Mme Naceur Djamila, responsable de la spécialité électronique des systèmes embarqués, pour son accueil bienveillant et son soutien indéfectible. Sa foi en notre potentiel nous a poussés à nous surpasser et à atteindre des résultats remarquables dans ce domaine.

Nous souhaitons exprimer notre gratitude à l'ensemble des enseignants du cycle Licence et Master qui nous ont transmis leur savoir et leur passion tout au long de notre parcours académique.

Nous adressons également nos remerciements sincères à toutes les personnes qui nous ont soutenus et aidés de près ou de loin dans la réalisation de ce travail, Leur encouragement constant, leur confiance en nos capacités et leur bienveillance ont été une source de motivation inestimable tout au long de cette aventure.

Dédicace

Je dédie ce travail à mes chers parents, qui ont consacré leur vie à la construction de la mienne. Qu'il s'agisse d'aide aux devoirs tard dans la nuit ou d'innombrables encouragements, leur soutien a été inébranlable ! Je leur serai à jamais reconnaissante.

À mon frère Mohamed, et mes sœurs Soumia et Sara, qui j'ai partagé d'innombrables rires, des nuits d'études tardives et un soutien familial indéfectible.

À Hana et Fethia, qui m'ont toujours soutenue et ont célébré mes victoires comme si elles étaient les leurs.

À ma Binôme, Sarah, avec qui j'ai collaboré et réfléchi pour donner vie à ce projet.

Anfel.

Dédicace

Je dédie ce travail à mes chers parents, pour leur amour inconditionnel, leur soutien indéfectible, et leurs sacrifices.

À mes sœurs, Wissem, Amira, et Meriem, pour leur encouragement constant et leur compréhension durant les moments difficiles.

À mes amis, Chayma, Lydia, Madina, Loubna, Ahlem ,Rym ,Hind Nesrine,Bayane ,Kawthar… et beaucoup d'autres pour leur soutien moral, leur patience, et les moments de joie partagés qui m'ont aidé à traverser cette période.

À Fodhil et Amine, pour leur soutien inestimable et leur aide précieuse tout au long de l'année

À ma copine de chambre Hadil, à Narjess et à tous les travailleurs de la cité 4 Zoubida Hamadouche pour avoir été ma deuxième famille.

À mes professeurs et Mr Kabir notre encadrant, pour leurs précieux conseils, leur expertise, et leur dévouement à mon éducation et à ma formation.

À ma belle promo ESE et au Club CSCC – pour les bons moments que je n'oublierai jamais

À ma famille élargie, pour leur amour et leur soutien.

À ma binôme, Anfel, pour sa collaboration, son soutien et son dévouement tout au long de ce projet.

À tous ceux qui ont cru en moi, m'ont inspiré, et m'ont donné la force de persévérer pour atteindre mes objectifs.

Merci à vous tous.

SARAH.

Résumé :

L'objectif de ce projet est de mettre en place un système de vidéosurveillance d'apprentissage profond pour analyser et séparer les comportements violents des comportements normaux en temps réel. Le modèle CNN pré-entraîné de MobileNetV1 est utilisé pour extraire des fonctionnalités de la vidéo, et ces fonctionnalités sont ensuite classées en utilisant le réseau LSTM. Des résultats précis ont été obtenus grâce à MobileNetV1, qui a démontré concrètement l'efficacité de la mise en œuvre du projet.

Mots clés : apprentissage profond, CNN, pré-entraîné, MobileNetV1, LSTM.

Abstract:

This project aims to establish a deep learning video surveillance system to analyze and separate violent behaviors from normal behaviors in real time. MobileNetV1's pre-trained CNN model is used to extract features from the video, and these features are then classified using the LSTM network. Accurate results were achieved using MobileNetV1, which demonstrated the project's implementation effectiveness in a concrete manner.

Keywords: deep learning, CNN, pre-trained, MobileNetV1, LSTM.

الملخص:

هذا المشروع يهدف إلى إنشاء نظام مراقبة بالفيديو قائم على التعلم العميق لتحليل السلوكيات العنيفة والفصل بينها وبين السلوكيات العادية في الوقت الفعلي. يتم استخراج الميزات من الفيديو باستخدام نموذج CNN مدرب مسبقاً متمثل في MobileNetV1 ، ومن ثم يتم تصنيف هذه الميزات باستخدام شبكة LSTM .حققنا نتائج دقيقة باستخدام MobileNetV1 ، مما يظهر فعالية تنفيذ المشروع بشكل ملموس.

الكلمات المفتاحية: التعلم العميقCNN، مدرب مسبقاً,MobileNetV1, LSTM .

Abréviations

IA	Intelligence Artificielle
DVR	Digital Video Recorder
NVR	Network Video Recorder
IP	Internet Protocol
CMS	Content Management System
IRA	Irish Republican Army
IoT	Internet of Things
HD	Haute Définition
SD	Standard Définition
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ML	Machine Learning
AI	Artificial Intelligence
FT	Forget Gate
IG	Input Gate
OG	Output Gate
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
NSL	Neural Structured Learning
DTW	Dynamic Time Warping
KNN	K-Nearest Neighbors
RGB	Red Green Blue
VID	Violent Interaction Detection
VIF	Violent Interaction Feature
HOMO	Histogram of Optical flow Magnitude and Orientation
ViT	Vision Transformer
OViF	Optical Flow Variability
RELU	Rectified Linear Unit
MLP	Multi-Layer Perceptron
IBM	International Business Machines Corporation
AVG	Average pooling
RVB	Rouge Vert Bleu
OPENCV	Open Source Computer Vision Library

Table des matières

Table des figures

Introduction générale

Aujourd'hui, nous constatons que les systèmes de sécurité, tels que les caméras de surveillance, sont devenus omniprésents et indispensables pour assurer la sécurité des personnes et des biens. Cependant, ces systèmes traditionnels commencent à montrer leurs limites et deviennent de plus en plus obsolètes et il devient alors impératif de développer de nouvelles méthodes de détection pour répondre efficacement à ces défis.

La détection de la violence est l'un des aspects les plus critiques des systèmes de sécurité modernes. Alors que les méthodes de surveillance traditionnelles deviennent obsolètes, le besoin de solutions plus avancées et intelligentes n'a jamais été aussi grand. Ce projet répond à cette nécessité en développant un système de détection de la violence en temps réel utilisant des technologies d'apprentissage profond de pointe.

L'objectif principal de ce projet est de développer un système de détection de la violence en temps réel efficace et fiable qui utilise des techniques avancées d'apprentissage profond pour améliorer les mesures de sécurité dans divers environnements. En intégrant MobilenetV1 pour l'extraction de fonctionnalités et les réseaux LSTM (Long Short-Term Memory) pour la reconnaissance de formes temporelles, le système vise à détecter avec précision les comportements violents, à améliorer les systèmes de surveillance et à améliorer la sécurité communautaire.

En combinant les dernières avancées en matière d'apprentissage profond avec une approche méthodique de développement et d'évaluation, ce projet vise à établir une référence en matière de détection de la violence en temps réel, contribuant ainsi à un monde plus sûr et plus sécurisé.

Pour cela, nous avons structuré notre travail en plusieurs chapitres. Le premier chapitre présente un aperçu de la vidéosurveillance et de l'IA, abordant notre problématique de détection de la violence. Le deuxième chapitre se concentre sur les technologies de l'IA, notamment l'apprentissage profond et les réseaux LSTM. Le troisième chapitre explore les différentes méthodes de détection de la violence, incluant le traitement d'image et l'apprentissage profond, ainsi que les travaux relatifs. Enfin, le quatrième chapitre décrit notre propre travail, y compris la configuration de l'environnement, la création de datasets, l'entraînement des modèles, l'évaluation des performances, et les défis d'intégration sur des dispositifs embarqués comme le Raspberry Pi.

Chapitre 1

Analyse des scènes en video-surveillance

1.1 Introduction

L'augmentation de la criminalité et de l'insécurité dans les lieux publics, tels que les stations de métro, les gares routières et les écoles, a entraîné une hausse de l'utilisation de la vidéosurveillance. Les progrès technologiques ont fait évoluer la vidéosurveillance traditionnelle vers des systèmes de vidéosurveillance intelligents. Ces systèmes sont capables de capturer des images, de les analyser, de les interpréter et de réagir aux événements en temps réel. Dans ce chapitre, nous allons explorer les principes fondamentaux de la vidéosurveillance et examine les composants et les caractéristiques de base des systèmes de vidéosurveillance intelligents et discute de leurs applications dans divers secteurs et environnements.

1.2 La vidéosurveillance

La vidéosurveillance est un ensemble composé d'une ou de plusieurs caméras et de tout ce qui est nécessaire pour pouvoir transmettre, enregistrer et exploiter les images dans le but de surveiller un lieu, tel qu'une entreprise, un entrepôt, une usine ou un espace public [1].

FIGURE 1.1 – Image d'illustration d'une caméra de surveillance. [1]

1.3 L'histoire de la vidéo surveillance

Il existe en effet des preuves suggérant que la première caméra de surveillance a été inventée en Union soviétique sous le règne de Joseph Staline, vers 1927. Le physicien

russe Leon Theremin a inventé le premier instrument de musique électronique au monde, le Theremin, cet instrument de musique consistait en un système en circuit fermé utilisant à la fois des technologies de caméra et de télévision. Cependant, le Kremlin a rapidement classé cette invention. [2]

Quinze ans plus tard, en 1942, Siemens a conçu ce que le monde considère comme le premier système de vidéosurveillance en Allemagne pour observer les lancements de fusées pendant la Seconde Guerre mondiale.

Dans les années 1980, le Royaume-Uni est devenu un pionnier dans le développement de la vidéosurveillance pour prévenir et combattre les attaques de l'IRA (Armée républicaine irlandaise). Depuis, ces systèmes ont proliféré dans les villes du monde entier, Londres étant l'une des plus équipées. Les caméras de vidéosurveillance ont été largement acceptées par le public en raison de leur rôle dans l'arrestation des auteurs des attentats de 2005.

Les États-Unis ont également connu une augmentation de l'utilisation des caméras de surveillance depuis les attaques terroristes du 11 septembre 2001, car elles sont devenues un aspect important des mesures de sécurité.[1]

Le marché de la vidéosurveillance est actuellement en plein essor en raison de la disponibilité de différents modèles de systèmes de surveillance avec une large gamme de qualité et de prix. Cela a entraîné une augmentation de la demande de la part du secteur public et des particuliers.

1.4 L'évaluation de la vidéosurveillance

Les premières caméras étaient perçues comme encombrantes et produisaient des images de moindre qualité. Elles étaient difficiles à installer et à entretenir, et souvent trop chères pour les entreprises et les particuliers.

Au fil du temps, les caméras sont devenues plus petites et la qualité vidéo s'est considérablement améliorée. Les caméras modernes sont désormais équipées d'un mode infrarouge, qui leur permet d'enregistrer des séquences même dans des environnements sombres ou non éclairés.

Les progrès en matière de technologie de stockage ont également été remarquables. Les fabricants de caméras ont amélioré les options de stockage, et de nombreuses entreprises utilisent désormais le stockage en nuage. En outre, les séquences enregistrées par votre caméra peuvent être directement téléchargées sur Internet. Cette fonctionnalité permet aux utilisateurs d'accéder à leurs images depuis n'importe quel endroit dans le monde, ce qui leur donne la possibilité de surveiller leur entreprise et de rester informés des activités en cours, quel que soit leur emplacement physique.[3] [4]

Cette caractéristique a contribué à leur popularité croissante et à leur utilisation généralisée. Actuellement, on les trouve couramment dans les foyers, les établissements commerciaux et les espaces publics du monde entier.

1.5 Architecture d'un système de vidéosurveillance

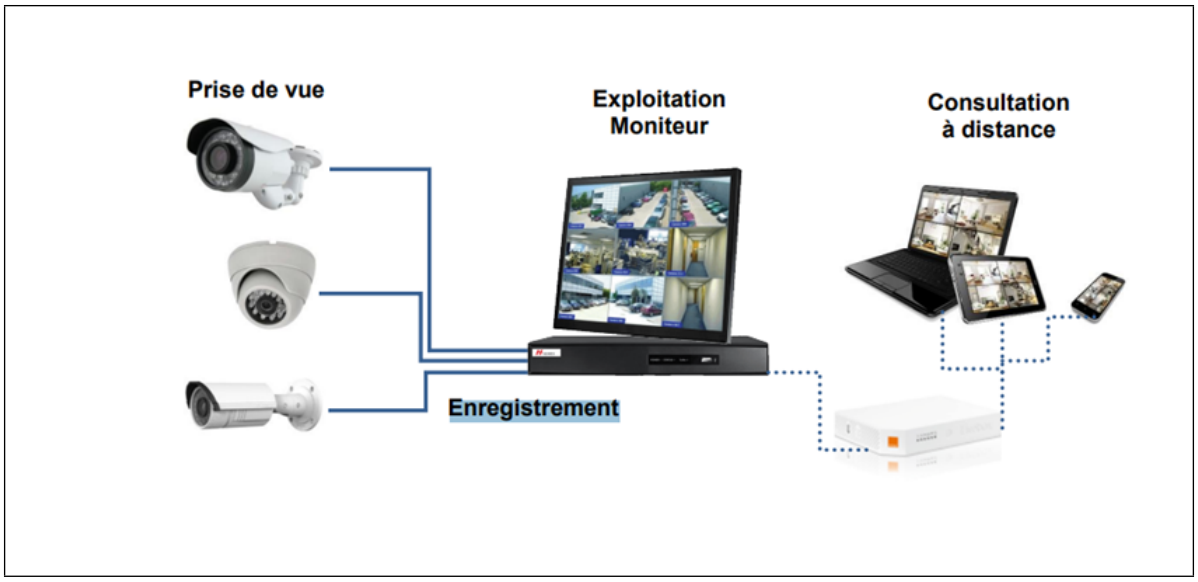

FIGURE 1.2 – Architecture d'un système de vidéosurveillance. [5]

— Exploitation : Enregistreur, logiciel CMS, clavier, souris, moniteur.

— Enregistrement : Disque dur, carte SD, serveur distant, Cloud.

— Prise de vue : Caméra HD, caméra IP, caméra Wifi, caméra mobile, Web cam.

• La caméra :

La caméra est la partie la plus importante du système de vidéosurveillance.Lors du choix d'une caméra pour un système de vidéosurveillance, il est important de prendre en compte plusieurs facteurs clés, tels que la luminosité, la qualité de l'image et la taille de l'objectif. En outre, il est essentiel de définir la zone à filmer, y compris sa largeur, sa profondeur et sa distance par rapport à la caméra. [5]

• Les moniteurs :

Les moniteurs sont des périphériques informatiques qui permettent aux utilisateurs d'interagir avec un écran. Ils sont chargés de présenter visuellement toutes les informations capturées par le réseau de caméras.[5]

• Enregistreurs vidéo (DVR/NVR) :

Ces appareils sont utilisés pour enregistrer et stocker les données vidéo capturées par les caméras. Les DVR sont généralement utilisés avec des caméras analogiques, tandis que les NVR sont conçus pour les caméras IP. Les images peuvent être visualisées en direct, enregistrées, lues et transmises via le réseau.[5]

• La compression :

La compression des données vidéo peut être un outil utile pour réduire la taille des fichiers et les besoins en mémoire, ainsi que pour améliorer les vitesses de transmission des données. C'est particulièrement vrai lorsqu'on la compare aux formats vidéo non compressés. [5]

1.6 Vidéosurveillance intelligente

La surveillance intelligente est une méthode qui implique l'intégration de technologies telles que l'intelligence artificielle (IA) et l'analyse de données. En utilisant des capteurs, des caméras, des connexions réseau et des algorithmes d'analyse intelligents, elle améliore les systèmes de surveillance traditionnels pour créer des solutions plus innovantes et plus efficaces. Le concept de vidéosurveillance intelligente implique l'observation d'une scène ou de plusieurs scènes afin d'identifier les actions qui peuvent être inappropriées ou indiquer l'occurrence d'un comportement inapproprié. Les données collectées sont utilisées pour effectuer diverses opérations, notamment la classification des individus en fonction de leurs mouvements et la génération d'une alarme en fonction de facteurs de déclenchement prédéfinis. [6]

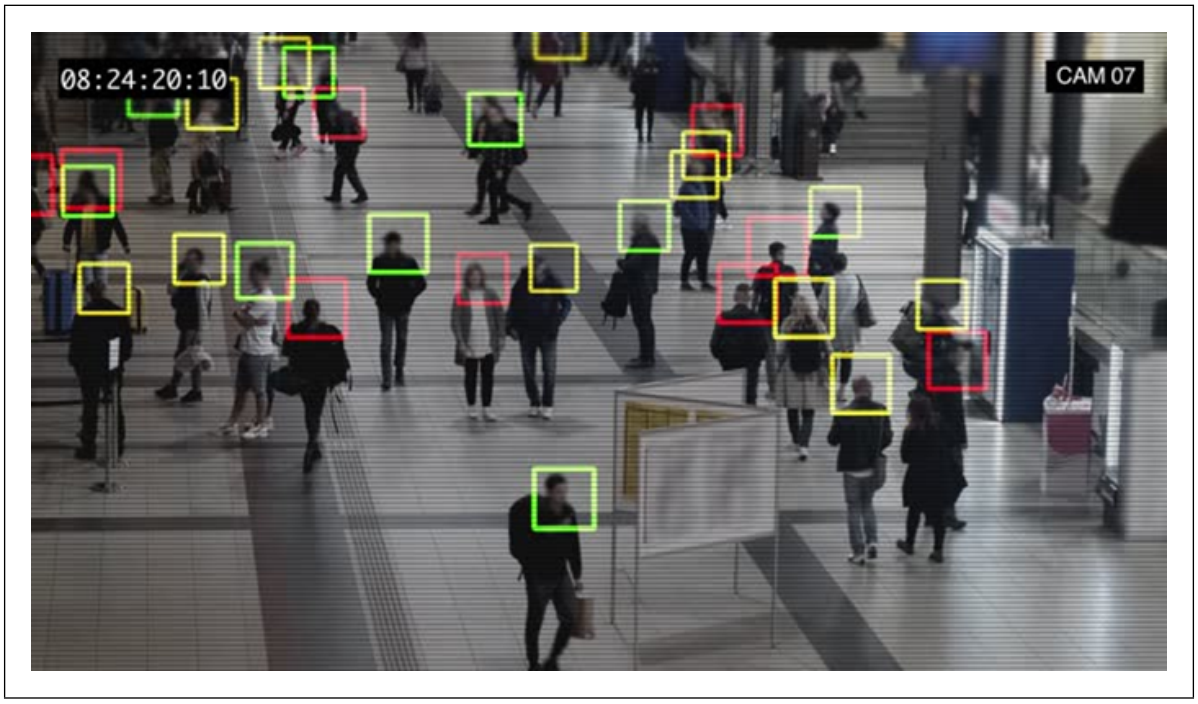

FIGURE 1.3 – Système du vidéosurveillance intelligent. [7]

1.7 Domaines d'utilisation de la vidéosurveillance intelligente

Au-delà de la vidéosurveillance traditionnelle, les systèmes de vidéosurveillance intelligents offrent un large éventail de capacités, en tirant parti des analyses et de l'automatisation avancées, ces systèmes permettent d'atténuer les risques, améliorer la connaissance de la situation et réagir efficacement aux incidents de sécurité en temps réel. Voici quelques-unes des applications les plus importantes des systèmes de vidéosurveillance intelligents :

1.7.1 Analyse comportementale

Les gens sont reconnus à partir de l'entrée visuelle et leurs actions sont analysées ou leurs chemins sont suivis en fonction de l'estimation de pose.[8]

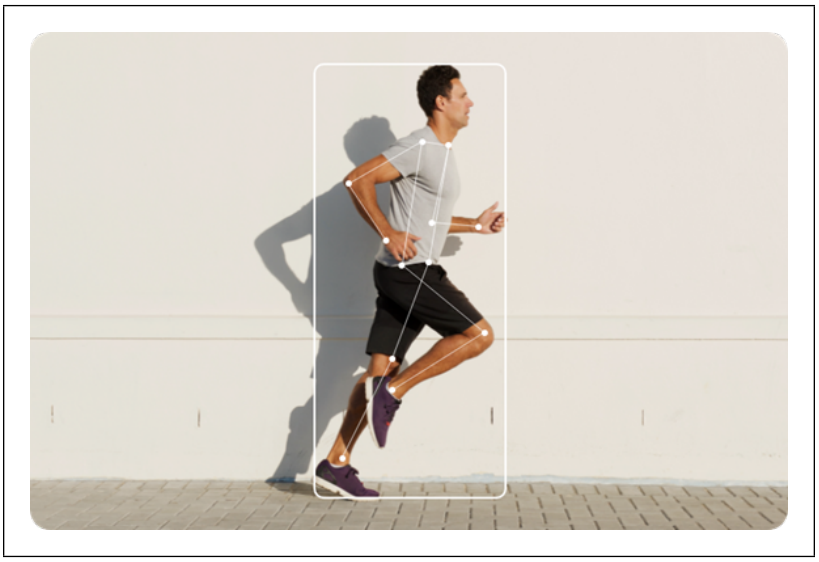

FIGURE 1.4 – Analyse des comportements. [8]

1.7.2 Détection et reconnaissance d'objets

La détection d'objets fait référence à la capacité des systèmes informatiques et logiciels à localiser des objets dans une image/scène et à identifier chaque objet. La détection d'objets est largement utilisée dans la reconnaissance faciale, la détection de véhicules, le comptage de piétons, les images Web, les systèmes de sécurité et les voitures sans conducteur.[9]

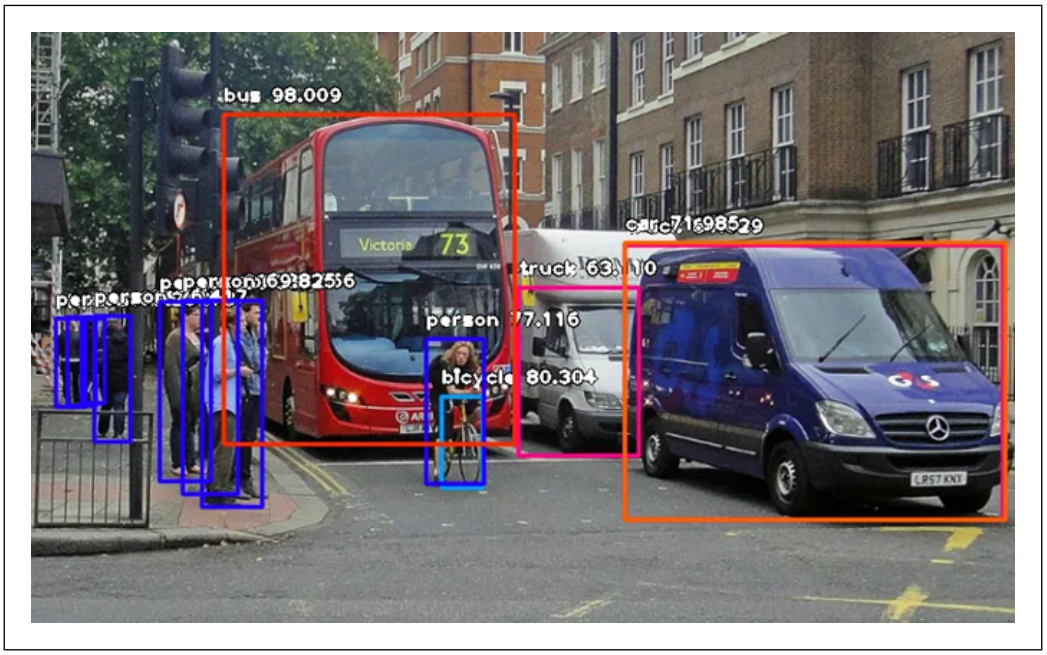

FIGURE 1.5 – Détection d'objets.[9]

1.7.3 Gestion des foules

Dans les zones bondées telles que les stades, les aéroports ou les événements publics, les systèmes de vidéosurveillance intelligents peuvent surveiller la densité des foules, les flux et les menaces potentielles à la sécurité pour assurer la sécurité publique et la gestion efficace des foules.[10]

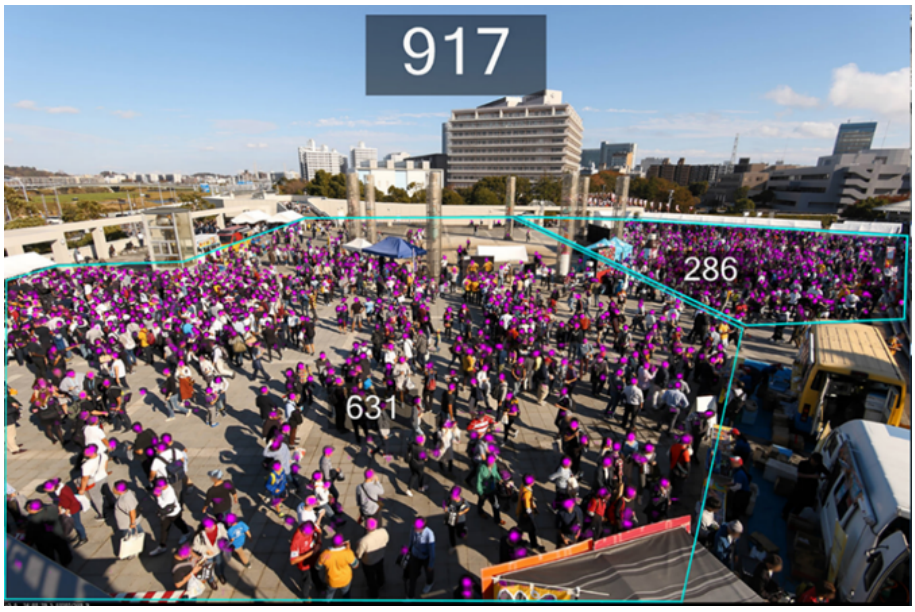

FIGURE 1.6 – Gestion des foules.[10]

1.7.4 Surveillance Environnementale

Il existe de nouvelles applications de vision par ordinateur pour la surveillance environnementale dans diverses industries. La technologie de vision par ordinateur de pointe fonctionne avec des caméras et des capteurs pour collecter des données et identifier et suivre les objets, les événements et les changements environnementaux.[11]

FIGURE 1.7 – Détection des feux de forêt.[11]

1.8 Comparaison entre la vidéosurveillance traditionnelle et la vidéosurveillance intelligente

Traditionnellement, la vidéosurveillance a été utilisée principalement pour enregistrer des événements, nécessitant que quelqu'un qui examine les images, en particulier en cas d'incident. Mais il n'était pas possible de surveiller en permanence des millions de caméras. Aujourd'hui, grâce à l'intelligence artificielle, ces systèmes peuvent faire plus qu'enregistrer. Ils peuvent interpréter, prédire et réagir aux situations au fur et à mesure qu'elles se déroulent. Cette comparaison vise à mettre en lumière l'évolution de la vidéosurveillance .[12]

	Vidéosurveillance traditionnelle	**Vidéosurveillance intelligente**
Surveillance/detection	Le contrôle manuel est sujet à des erreurs humaines.	Les systèmes d'IA peuvent analyser les images en continu, 24/7.
Stockage	Nécessité d'avoir une grande mémoire et de la vider souvent.	Utilise soit le stockage dans le cloud, soit une carte mémoire micro SD.
Précision	Dépend fortement de l'expérience et de l'expertise de l'opérateur.	Maintiennent un niveau de précision constant, réduisant les erreurs et les biais humains.
Consommation d'énergie	Fonctionnent à l'électricité.	Fonctionnent avec des batteries rechargeables.
Efficacité	L'extension d'un dispositif de sécurité se traduit souvent par une augmentation linéaire des coûts.	Permettent d'ajouter des caméras sans augmenter proportionnellement le personnel de surveillance.
Flexibilité et intégration	Limites en ce qui concerne l'intégration des nouvelles technologies.	Conception axée sur l'intégration. Ils fonctionnent souvent de manière transparente avec d'autres appareils IoT et plateformes technologiques.
Temps de réponse	Cela dépend du temps de réaction de l'homme.	Déclencher des alertes automatisées dès qu'elles détectent des anomalies.

TABLE 1.1 – Comparaison entre vidéosurveillance traditionnelle et intelligente. [13]

1.9 Détection des mouvements suspects

La reconnaissance de l'activité humaine à partir de la surveillance vidéo est un domaine de recherche actif dans le traitement de l'image et la vision par ordinateur. Elle consiste à reconnaître et à classer les activités humaines comme normales ou anormales. Les activités anormales font référence à des comportements inhabituels ou suspects rare-

ment observés dans les lieux publics, tels que le vol, la violence, les agressions, le vandalisme et le franchissement de frontières. En revanche, les activités normales font référence à des comportements typiques couramment observés dans les lieux publics, tels que la course, le jogging, la marche et l'agitation des mains. La vidéosurveillance analyse les images vidéo capturées pour identifier les activités inhabituelles ou suspectes, grâce à l'utilisation d'algorithmes précis et efficaces créés à l'aide de techniques d'apprentissage automatique et d'analyse de données. [13]

1.10 Processus de détection des mouvements suspects

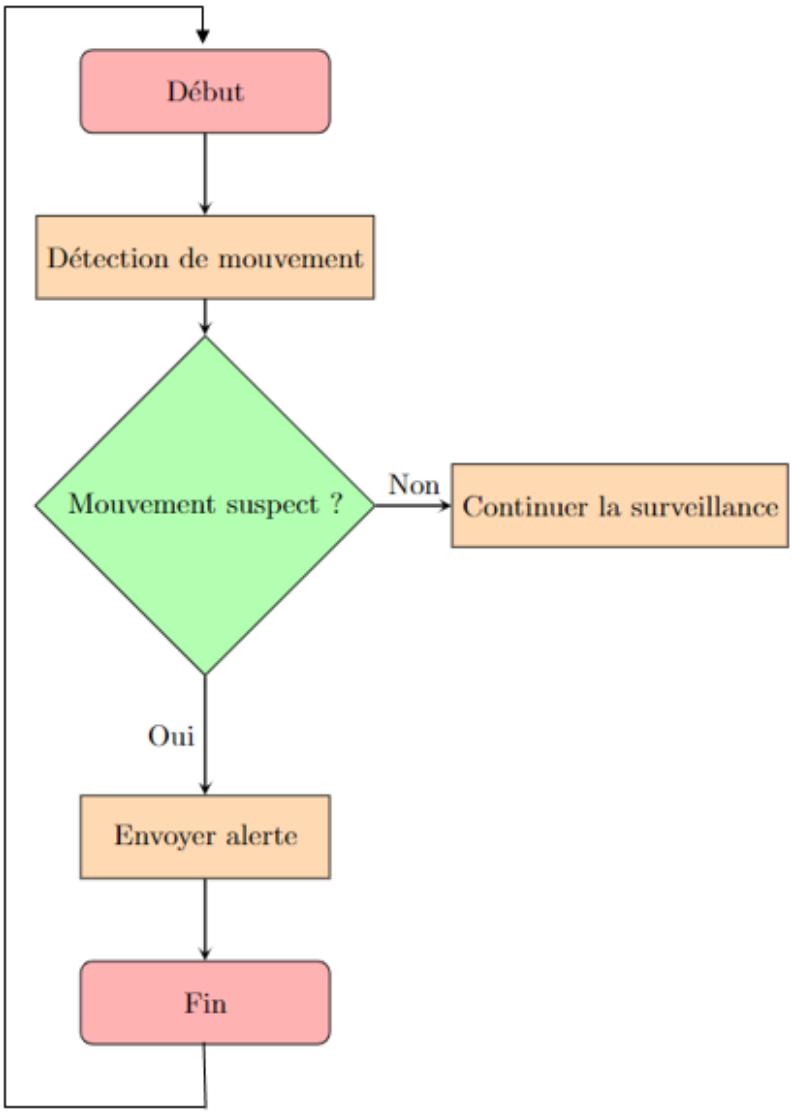

FIGURE 1.8 – Processus de détection des mouvements suspects

1.11 Applications de détection des mouvements suspects

Voici quelques-unes des activités que la détection des mouvements suspects peut identifier et analyser :

• Abandon d'objet : Les sacs ou colis abandonnés ne sont généralement pas très préoccupants, car ils peuvent être le résultat d'un voyageur distrait ou intentionnellement placé avec de mauvaises intentions. Cependant, avec l'utilisation de la détection des mouvements suspects, un objet abandonné peut être rapidement identifié et le personnel de sécurité ou d'application de la loi peut être alerté de sa présence. [14]

FIGURE 1.9 – Détection d'un objet abandonné et de son propriétaire. [14]

• Détection de la chute des personnes : Une personne peut être au sol, par exemple pour ramasser un objet déposé. Cependant, ils peuvent également être en détresse en raison d'un problème de santé ou avoir été poussés au sol. La détection des mouvements suspects peut fournir une analyse contextuelle pour identifier la situation avec précision, alertant le personnel approprié pour enquêter. [15]

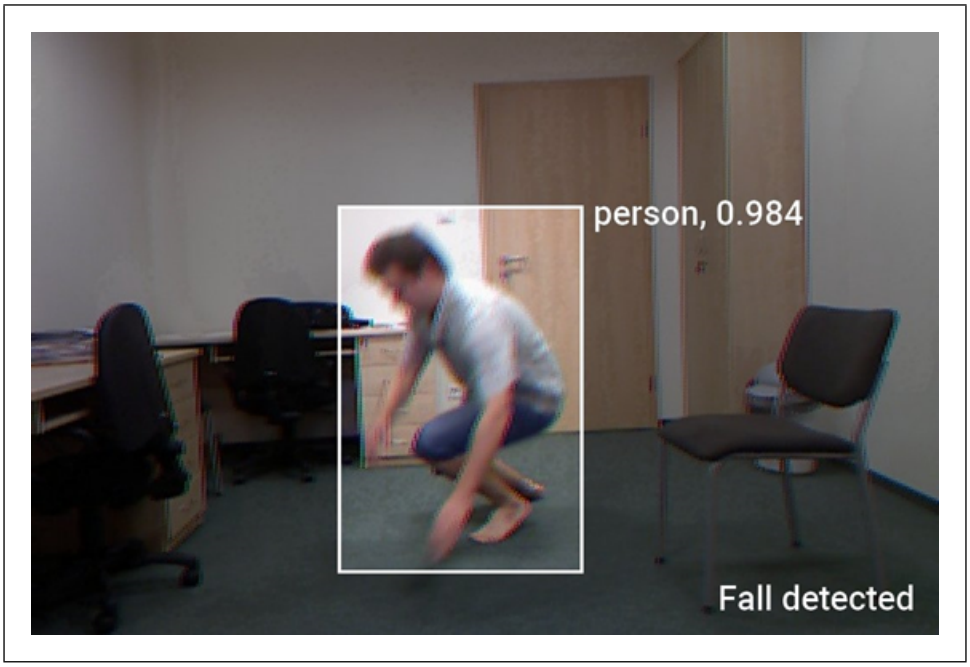

FIGURE 1.10 – Détection de la personne tombante. [15]

• Personne tenant une arme : La détection d'armes par la détection des mouvements suspects est une méthode puissante pour identifier les situations menaçantes et potentiellement dangereuses tout en réduisant les fausses alarmes. [16]

FIGURE 1.11 – Détection de personne tenant une arme. [16]

• Violence : Les cas de violence, qu'ils soient manifestes ou subtils, peuvent être alarmants dans n'importe quel environnement. Reconnaître ces comportements nécessite un œil attentif et une action rapide. La détection des mouvements suspects est utilisée pour discerner et catégoriser ces incidents à mesure qu'ils se déroulent. [17]

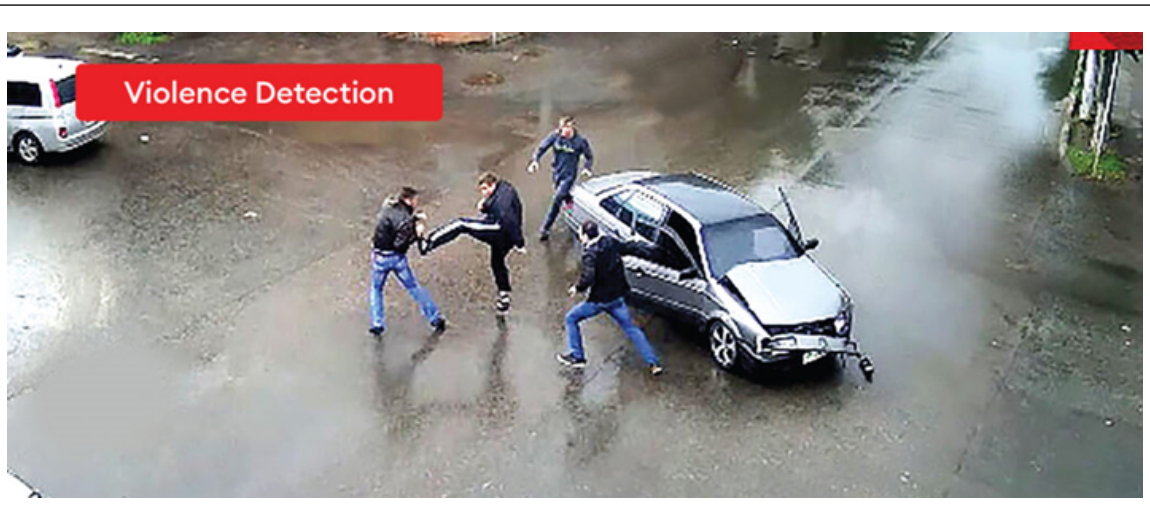

FIGURE 1.12 – Détection de violence. [17]

• Détection des intrusions : Parmi les progrès les plus novateurs, la détection d'intrusion. Les faux positifs ou les fausses alarmes peuvent éroder la confiance dans les systèmes de surveillance, entraînant une lassitude à l'égard des alertes ou des bruits. Les solutions

d'IA peuvent éliminer les alarmes en définissant l'objet d'intérêt pour la surveillance et en se concentrant sur la zone d'intrusion. Cette fonction fait appel à des algorithmes sophistiqués pour non seulement détecter les intrusions, mais aussi pour distinguer les menaces réelles des événements inoffensifs. [18]

FIGURE 1.13 – Détection d'intrusion. [18]

1.12 Problématique

À l'ère moderne, les systèmes de vidéosurveillance sont largement utilisés dans les espaces publics, les aéroports et les stades. Ces systèmes sont devenus un outil indispensable dans le domaine de la sécurité et de la surveillance en raison de leur capacité à détecter et à enregistrer les mouvements suspects. Cependant, malgré leur utilité, ces systèmes présentent encore des limites en termes de précision et d'efficacité dans la détection des actes de violence.

Comment développer les systèmes de vidéosurveillance afin d'améliorer leur capacité à classer et à détecter avec précision les mouvements violents pour renforcer la sécurité publique ? En outre, comment intégrer ces avancées technologiques tout en respectant les droits individuels et la vie privée ?

Dans le cadre de notre projet, nous travaillons sur l'amélioration de l'efficacité des systèmes de vidéosurveillance dans la détection des mouvements suspects et sur l'amélioration des techniques de reconnaissance des formes et des modèles inhabituels afin d'améliorer et de développer les systèmes de vidéosurveillance. Nous explorons des techniques avancées de traitement d'images et d'apprentissage automatique pour identifier plus rapidement et

plus précisément les situations potentiellement dangereuses, tout en assurant un équilibre entre la sécurité publique et le respect des libertés individuelles.

1.13 Objectifs et l'impact de détection de violence sur la vidéosurveillance

L'objectif de cette recherche sur la détection de la violence dans la vidéosurveillance est d'améliorer la précision et la rapidité de la détection de la violence, ce qui est essentiel pour renforcer la sécurité publique.

Les approches traditionnelles sont souvent sujettes à des erreurs en raison de la complexité des données vidéo, de la variabilité du comportement humain et de la nécessité d'un contrôle manuel important. L'apprentissage profond peut surmonter ces difficultés en apprenant à partir de vastes ensembles de données vidéo annotées, ce qui permet une détection automatisée et précise des comportements violents. L'impact de cette recherche est important car elle peut contribuer à des réponses plus efficaces et plus rapides aux incidents violents, ce qui peut permettre d'éviter l'escalade et de réduire les dommages. En outre, l'utilisation de l'apprentissage profond pour la détection de la violence permet aux systèmes d'affiner leur précision au fil du temps et de devenir plus rentables à long terme, en réduisant le besoin de supervision humaine.

En résumé, la recherche sur la détection de la violence dans la vidéosurveillance à l'aide de l'apprentissage profond peut fournir des approches plus efficaces et plus fiables pour la détection et la réponse aux incidents, conduisant à des environnements plus sûrs dans les espaces publics, les institutions et les communautés,

1.14 Conclusion

Dans ce chapitre nous avons donné un bref aperçu et une introduction de la vidéosurveillance, on a fourni une vue d'ensemble complète de l'histoire de la vidéosurveillance et comment elle a évolué et s'est modernisée avec le temps, et aussi comment il s'intègre à l'intelligence artificielle pour devenir plus efficace et plus précis. Le chapitre 2 se concentre sur les domaines de l'IA, en mettant l'accent sur les défis de l'apprentissage automatique et de l'apprentissage profond dans la vidéosurveillance.

Chapitre 2

L'intelligence artificielle dans l'analyse des scenes

2.1 Introduction

Le domaine de la vidéosurveillance est profondément transformé par l'intelligence artificielle qui offre des capacités avancées d'analyse et de détection automatique. Les systèmes de vidéosurveillance intelligents peuvent analyser en temps réel les flux vidéo en utilisant des algorithmes avancés de traitement d'image et de machine learning afin de détecter des anomalies, des comportements suspects et des incidents de sécurité, y compris les actes de violence. En automatisant la détection et la classification des événements, l'intelligence artificielle améliore considérablement l'efficacité des systèmes de surveillance, ce qui permet de réduire la dépendance à l'intervention humaine et de réduire le taux de fausses alertes. En outre, les technologies de reconnaissance faciale et de suivi des objets offrent une identification et un suivi plus précis et plus rapides des individus et des objets.

Dans ce chapitre, nous allons explorer l'intelligence artificielle et ses technologies clés. Nous aborderons les concepts fondamentaux du machine learning (ML) et du deep learning (DL), ainsi que les différentes architectures de réseaux de neurones telles que les réseaux monocouches, les MLP (Multi-Layer Perceptron), les CNN (Convolutional Neural Networks), les RNN (Recurrent Neural Networks) et les LSTM (Long Short-Term Memory).

2.2 Définition de l'IA

L'IA est une discipline qui cherche à reproduire l'intelligence humaine grâce à des systèmes informatiques, des données et des algorithmes sophistiqués. Des technologies comme l'apprentissage automatique et le deep learning permettent aux machines d'acquérir des connaissances à partir de données et de prendre des décisions en imitant le raisonnement humain. L'intelligence artificielle est devenue une composante essentielle de nombreuses industries, fournissant des bénéfices considérables dans des secteurs tels que la médecine, la finance, la production, la logistique et bien d'autres. Par exemple, en médecine, l'intelligence artificielle est employée afin d'analyser des scans médicaux et d'assister les médecins dans la mise en place de diagnostics plus précis. En outre, dans l'industrie financière, elle est employée afin de repérer les fraudes et gérer les dangers. Ainsi, l'influence de l'intelligence artificielle sur la société et l'économie est importante, et sa pertinence ne cesse de croître à mesure que de nouvelles applications se développent. [19]

2.3 Histoire de l'IA

2.3.1 Origine de l'IA

Depuis au moins le 1er siècle avant notre ère, l'homme a travaillé sur la conception de machines capables d'imiter le raisonnement humain. L'expression « intelligence artificielle » est une création récente, en 1955, par John McCarthy. John McCarthy et ses collègues ont organisé en 1956 une conférence intitulée « Dartmouth Summer Research Project on Artificial Intelligence » qui a conduit à l'émergence du machine learning, du deep learning, des analyses prédictives et, récemment, des analyses prescriptives. On a également vu émerger un nouveau champ d'étude : la science des informations. [20]

2.3.2 Évolution de l'IA

Depuis les débuts marqués par les travaux de John McCarthy et Alan Turing, l'intelligence artificielle a connu plusieurs phases de développement. Dans les années 1960 et 1970, les chercheurs se sont concentrés sur la résolution de problèmes et la représentation des connaissances, menant à la création des premiers systèmes experts. Ces systèmes étaient capables de simuler le raisonnement humain dans des domaines spécifiques, mais ils étaient limités par leur dépendance à des règles codées manuellement. Les années 1980 ont vu l'émergence de l'apprentissage automatique, où les machines ont commencé à apprendre à partir de données, plutôt que de suivre des instructions programmées. Cette période a également été marquée par des progrès dans les réseaux neuronaux, bien que les capacités de calcul limitées de l'époque aient restreint leur application.

L'évolution la plus significative est survenue à partir des années 2000, avec l'augmentation exponentielle de la puissance de calcul et la disponibilité massive de données. Les avancées en apprentissage profond (deep learning), basées sur des réseaux de neurones artificiels complexes, ont permis des progrès spectaculaires dans des domaines comme la vision par ordinateur, la reconnaissance vocale et le traitement du langage naturel. Des chercheurs tels que Yann LeCun, Geoffrey Hinton et Yoshua Bengio ont joué des rôles clés dans ces avancées, en développant des technologies comme les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN), qui ont transformé des applications pratiques allant de la conduite autonome à l'assistance virtuelle. [21]

FIGURE 2.1 – L'ordinateur Deep Blue d'IBM a battu le champion du monde d'échecs Garry Kasparov en 1997. [21]

2.4 Principales Technologies et Techniques de l'IA

2.4.1 L'apprentissage automatique (Machine Learning)

L'apprentissage automatique est un domaine de l'informatique et de l'intelligence artificielle (IA) qui utilise des algorithmes et des données pour permettre à l'IA d'apprendre d'une manière qui imite l'apprentissage humain, en améliorant progressivement sa précision. Bien que les termes "apprentissage automatique" et "intelligence artificielle" soient souvent utilisés de manière interchangeable, ils ont des significations distinctes. L'IA fait référence à la tentative de créer des machines dotées de capacités cognitives semblables à celles de l'homme. L'apprentissage automatique, quant à lui, utilise spécifiquement des algorithmes et des ensembles de données pour atteindre cet objectif. Il se concentre sur la création et la mise en œuvre d'algorithmes qui facilitent les décisions et les prédictions. Ces algorithmes améliorent leurs performances au fil du temps, devenant plus précis et plus efficaces à mesure qu'ils traitent davantage de données. Dans l'apprentissage automatique, l'ordinateur reçoit un ensemble de données et une tâche à accomplir. L'ordinateur détermine alors comment accomplir la tâche en se basant sur les exemples qui lui sont donnés. Par exemple, si nous voulons qu'un ordinateur reconnaisse des images de chats, nous lui fournissons des milliers d'images de chats et laissons l'algorithme d'apprentissage automatique identifier les modèles et les caractéristiques communes qui définissent un chat. Au fil du temps, l'algorithme devient plus compétent pour reconnaître les chats, même lorsqu'on lui présente de nouvelles images. Cette capacité à apprendre à partir des données et à s'améliorer au fil du temps fait de l'apprentissage automatique un outil puissant et polyvalent. [44]

Les Types d'apprentissage automatique Les algorithmes d'apprentissage automatique peuvent être formés de différentes manières, chacune ayant ses propres avantages et inconvénients. L'apprentissage automatique se divise en quatre principaux types :

2.4.2 L'apprentissage supervisé

L'apprentissage supervisé est la forme la plus courante d'apprentissage automatique. Dans cette approche, le modèle est formé sur un ensemble de données étiquetées. En d'autres termes, les données sont accompagnées d'une étiquette que le modèle tente de prédire. Il peut s'agir d'une étiquette de catégorie ou d'un nombre réel. Au fur et à mesure que les données d'entrée sont introduites dans le modèle, celui-ci ajuste ses poids jusqu'à ce qu'il soit correctement adapté. Cette opération s'inscrit dans le cadre du processus de validation croisée, qui permet de s'assurer que le modèle n'est pas surajusté ou sous-ajusté. Le modèle apprend une correspondance entre l'entrée (caractéristiques) et la sortie (étiquette) au cours du processus de formation. Une fois formé, le modèle peut prédire la sortie pour de nouvelles données inédites. [45]

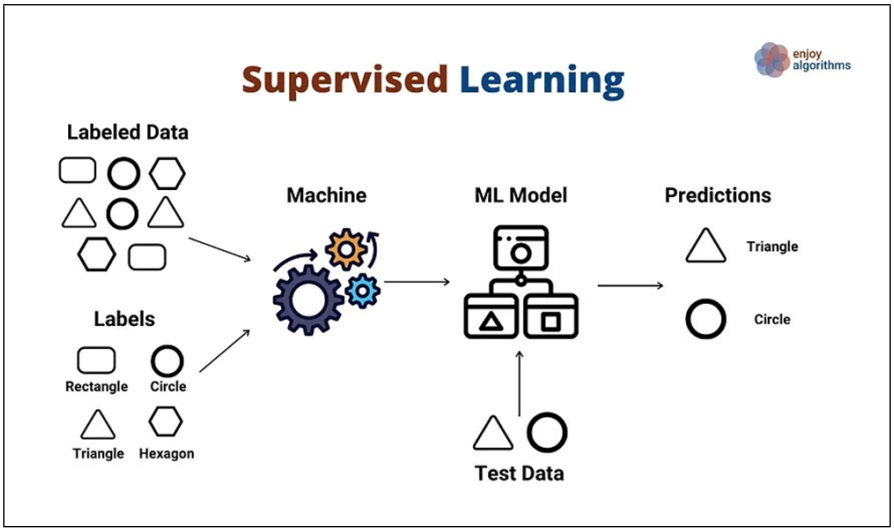

FIGURE 2.2 – Apprentissage supervisé. [46]

L'apprentissage automatique supervisé se divise en deux catégories principales : la classification et la régression.

-Les algorithmes de classification sont utilisés pour résoudre des problèmes où la variable de sortie est catégorique, comme oui ou non, vrai ou faux, homme ou femme, etc.

-Les algorithmes de régression sont utilisés pour traiter les problèmes de régression lorsqu'il existe une relation linéaire entre les variables d'entrée et de sortie. Ils sont couramment utilisés pour prédire des variables de sortie continues, telles que les modèles météorologiques et les tendances du marché. [45]

2.4.3 L'apprentissage non supervisé

-L'apprentissage non supervisé utilise des ensembles de données non étiquetées pour former les algorithmes. Dans ce processus, l'algorithme est alimenté par des données qui ne contiennent aucune étiquette, ce qui l'oblige à découvrir des modèles par lui-même, sans aucune aide extérieure. Ces algorithmes découvrent des modèles cachés ou des regroupements de données sans intervention humaine. Sa capacité à découvrir des similitudes et des différences dans les informations en fait un outil idéal pour l'analyse exploratoire des données, les stratégies de vente croisée, la segmentation de la clientèle et la reconnaissance des images et des formes. Il est également utilisé pour réduire le nombre de caractéristiques d'un modèle grâce au processus de réduction de la dimensionnalité. [45]

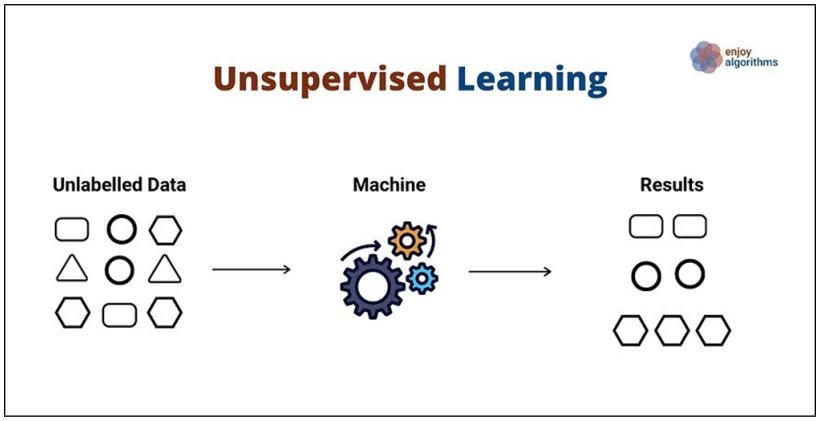

FIGURE 2.3 – Apprentissage non supervisé. [46]

L'apprentissage automatique non supervisé est classé en deux catégories : Clustering et Association.

- Le clustering consiste à regrouper des objets sur la base de similitudes ou de différences, comme le regroupement de clients en fonction des produits qu'ils achètent.

-L'apprentissage par association consiste à identifier les relations typiques entre les variables d'un vaste ensemble de données. Il détermine la dépendance de divers éléments de données et met en correspondance les variables associées. Il est couramment utilisé dans l'exploration de l'utilisation du web et l'analyse des données de marché. [45]

2.4.4 L'apprentissage semi- supervisé

-L'apprentissage semi-supervisé combine les caractéristiques de l'apprentissage automatique supervisé et non supervisé. Il utilise une combinaison d'ensembles de données étiquetés et non étiquetés pour former ses algorithmes. En règle générale, dans le cadre de l'apprentissage automatique semi-supervisé, les algorithmes reçoivent d'abord une petite quantité de données étiquetées pour guider leur développement, puis des quantités beaucoup plus importantes de données non étiquetées pour compléter le modèle. Par exemple, un algorithme peut être entraîné sur un petit ensemble de données vocales étiquetées, puis sur un ensemble beaucoup plus important de données vocales non étiquetées, afin de développer un modèle d'apprentissage automatique capable de reconnaître la parole. [45]

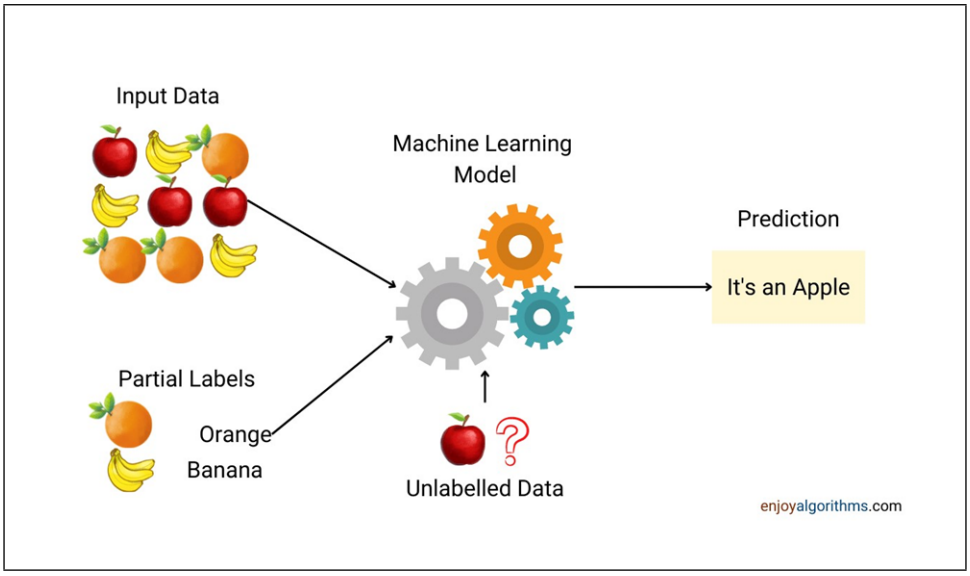

FIGURE 2.4 – Apprentissage semi-supervisé. [47]

2.4.5 L'apprentissage par renforcement

-L'apprentissage par renforcement est une technique d'apprentissage automatique dans laquelle un agent interagit avec son environnement pour prendre des décisions. L'agent reçoit des récompenses ou des pénalités pour ses actions, dans le but de maximiser la récompense totale. Cet algorithme acquiert progressivement une compréhension de son environnement, optimisant ses actions pour obtenir des résultats spécifiques, un peu comme un enfant apprend. Par exemple, un algorithme peut être optimisé par des parties d'échecs successives, ce qui lui permet d'apprendre de ses succès et de ses échecs passés. [45]

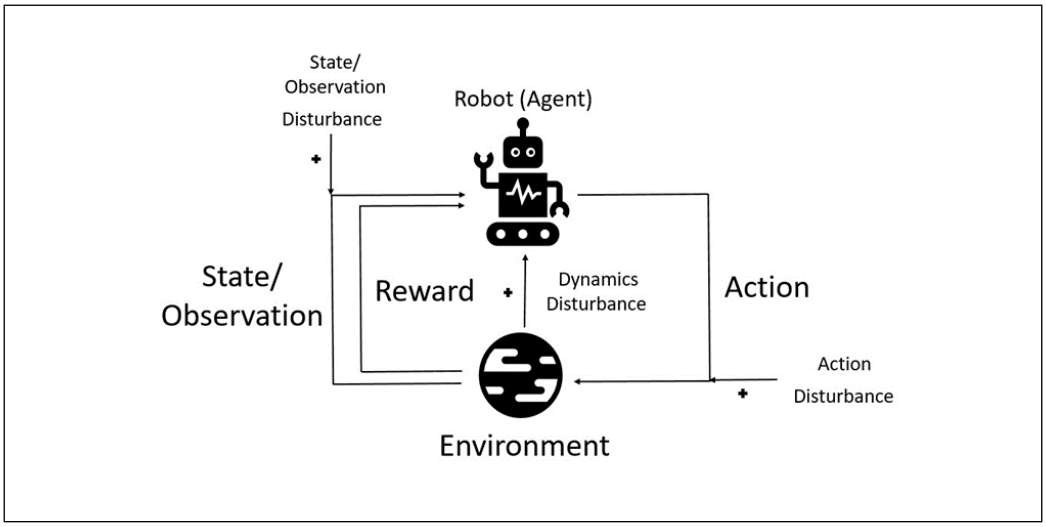

FIGURE 2.5 – Apprentissage par renforcement. [48]

L'apprentissage par renforcement se divise en deux types de méthodes : le renforcement positif et le renforcement négatif. -Le renforcement positif consiste à ajouter un stimulus de renforcement après un comportement spécifique de l'agent, ce qui augmente la probabilité que le comportement se reproduise à l'avenir, comme l'ajout d'une

récompense après un comportement. - L'apprentissage par renforcement négatif renforce un comportement spécifique qui évite un résultat négatif. [45]

2.4.6 Apprentissage Profond (Deep Learning)

Le deep learning est une sous-discipline de l'apprentissage automatique qui utilise des réseaux de neurones artificiels pour réaliser des calculs sur de grandes quantités de données. Inspiré par la structure et le fonctionnement du cerveau humain, le deep learning repose sur des couches multiples de neurones artificiels, ou "couches profondes", qui permettent de modéliser des relations complexes dans les données. [25]

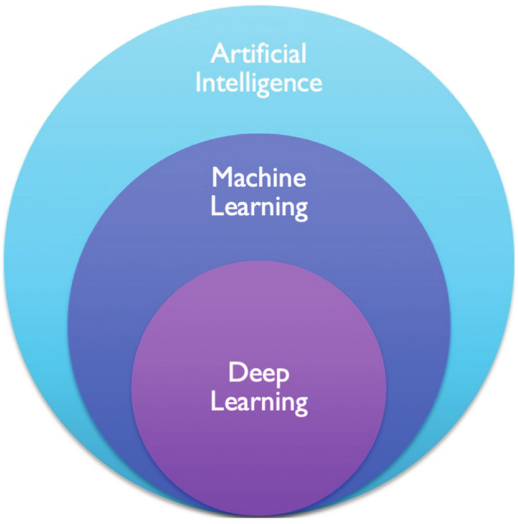

FIGURE 2.6 – Sous-ensembles d'IA. [81]

2.4.7 Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels sont au cœur du deep learning et ont révolutionné plusieurs domaines grâce à leur capacité à apprendre et à généraliser à partir de grandes quantités de données. Inspirés par le fonctionnement du cerveau humain, ces réseaux sont capables de modéliser des relations complexes et d'extraire des caractéristiques pertinantes des données d'entrée Un réseau de neurones est structuré comme le cerveau humain et se compose de neurones artificiels, également appelés nœuds. Ces nœuds sont organisés en trois couches :

— La couche d'entrée (The input layer)
— La ou les couches cachées (The hidden layer(s))
— La couche de sortie (The output layer)

Les données fournissent à chaque nœud des informations sous forme d'entrées. Le nœud multiplie les entrées avec des poids aléatoires, les calcule et ajoute un biais. Enfin, des fonctions non linéaires, également appelées fonctions d'activation, sont appliquées pour déterminer quel neurone doit être activé. [24]

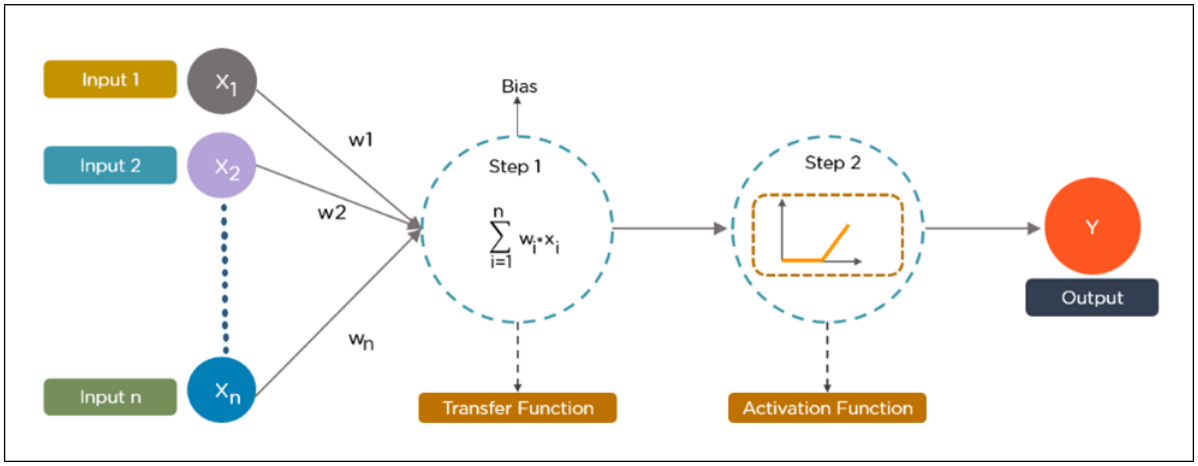

FIGURE 2.7 – un réseau de neurones artificiels. [73]

2.4.8 processus représenté dans un réseau de neurones artificiels

Entrées (Input 1, Input 2, Input n) Les entrées représentent les données initiales fournies au réseau de neurones. Chaque entrée est un élément de l'ensemble de données que le réseau va traiter. Dans une application de reconnaissance d'images, les entrées pourraient être les pixels d'une image. [26]

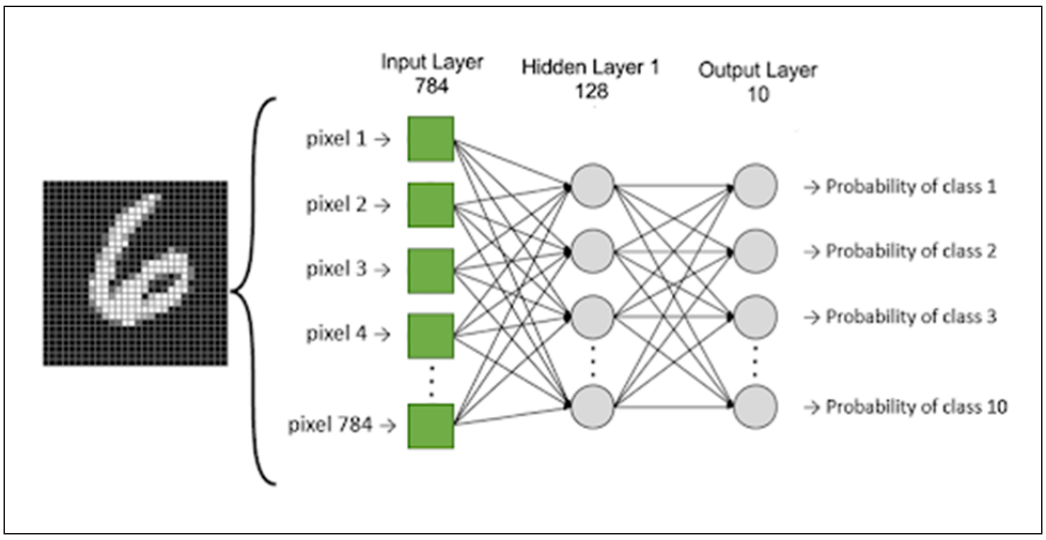

FIGURE 2.8 – Réseau de neurones pour la reconnaissance d'image. [22]

Neurones (X1, X2, Xn) : Les neurones sont des unités de base du réseau. Chaque entrée est associée à un neurone dans la couche d'entrée. Ces neurones prennent les valeurs d'entrée et les transmettent à la couche suivante.

Poids (w1, w2, wn) : Les poids sont des coefficients appliqués aux entrées. Chaque connexion entre un neurone d'entrée et un neurone de la couche suivante a un poids associé. Les poids ajustent l'importance de chaque pixel dans le processus de reconnaissance d'image.

Fonction de transfert (Transfer Function) : La fonction de transfert est utilisée pour combiner les entrées pondérées. Elle effectue une somme pondérée des entrées et ajoute un biais.

Biais (Bias) : Le biais est une constante ajoutée au résultat de la fonction de transfert. Il permet de décaler la fonction d'activation pour mieux ajuster les données. Le biais peut aider à ajuster les prévisions du modèle même lorsque toutes les entrées sont nulles. [26]

Fonction d'activation (Activation Function) :

La fonction d'activation applique une transformation non linéaire au résultat de la fonction de transfert. Elle détermine si un neurone doit être activé ou non. [37]

Sortie (Output) : La sortie est le résultat final produit par le réseau de neurones après avoir passé toutes les couches de neurones et les transformations. C'est la réponse du réseau pour les entrées données.

Exemple : Dans une application de reconnaissance d'image, la sortie pourrait être la probabilité que l'image représente une certaine classe

2.4.9 Le perceptron

Le perceptron est l'un des concepts fondamentaux en intelligence artificielle et en apprentissage automatique. Développé par Frank Rosenblatt en 1957, le perceptron est un type de modèle de réseau de neurones artificiels simple mais puissant, utilisé principalement pour la classification binaire telles que la détection de spam, la classification de documents, la reconnaissance de caractères, etc. [22]

Fonctionnement du Perceptron :

Le perceptron est un modèle à une seule couche, composé d'entrées, de poids, d'un biais et d'une fonction d'activation. Le rôle du Perceptron est en fait une fonction mathématique. Les coefficients de poids (w) sont multipliés par les données d'entrée (x). Le produit obtenu est une valeur. Il est possible que cette valeur soit positive ou négative. Si la valeur est positive, le neurone artificiel se met en marche. Il ne commence donc à fonctionner que lorsque le poids calculé des données d'entrée dépasse un seuil spécifique. On compare le résultat prédit au résultat connu. Si une différence se produit, l'erreur est rectifiée pour permettre d'ajuster les poids. [22]

Entraînement Pendant la phase d'entraînement, les poids et le biais i sont ajustés itérativement à l'aide d'un algorithme d'optimisation tel que la descente de gradient, afin de minimiser une fonction de perte, comme l'erreur quadratique moyenne.

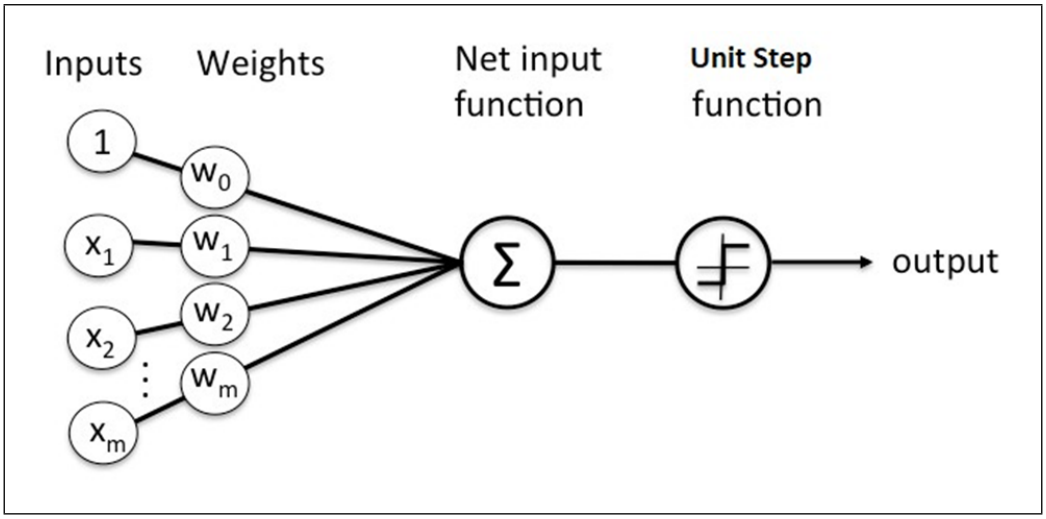

FIGURE 2.9 – Perceptron .[72]

Limitations du Perceptron Bien que le perceptron soit un modèle simple et efficace, il présente certaines limitations :

- Il ne peut pas résoudre des problèmes non linéairement séparables.
- Il peut converger vers une solution mais pas nécessairement vers la meilleure solution.
- Il peut être sensible à l'initialisation des poids et au choix de la fonction d'activation.

Ces limitations ont conduit au développement de modèles de réseau de neurones plus complexes, tels que les perceptrons multicouches et les réseaux de neurones profonds, capables de modéliser des relations plus complexes dans les données. [23]

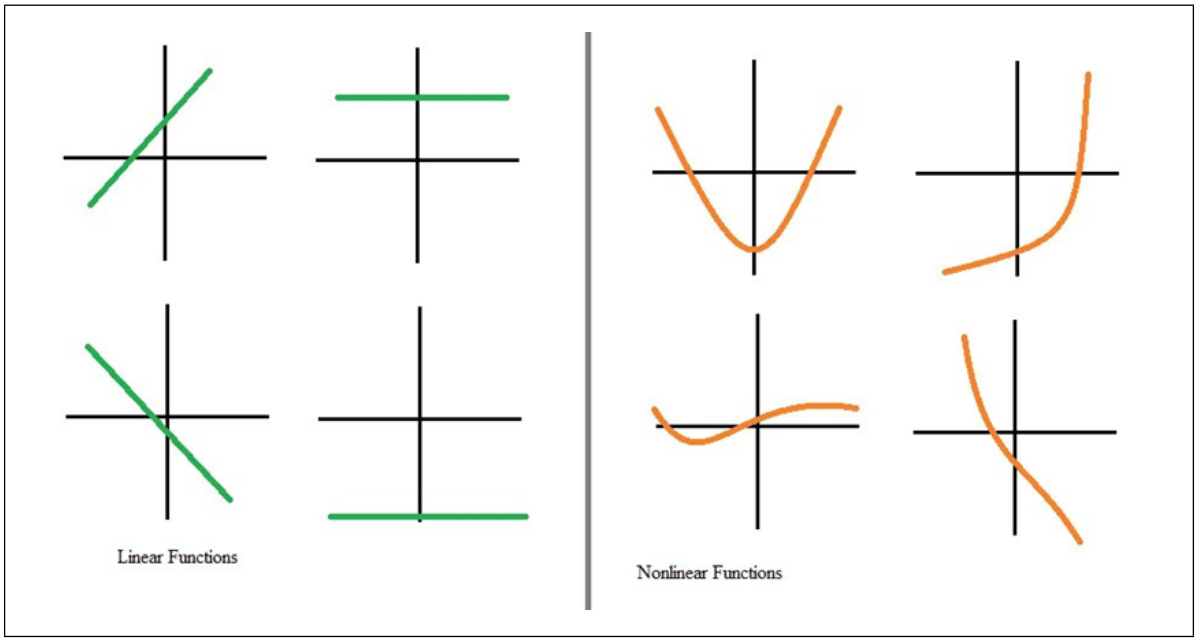

FIGURE 2.10 – Fonctions Lineaire et non lineaire. [72]

Problème du XOR Le perceptron simple a montré ses limites lorsqu'il est confronté à des problèmes non linéaires, comme le célèbre problème du XOR (ou exclusif ou). Ce

problème consiste à déterminer la sortie d'une fonction logique XOR qui ne peut pas être séparée par une seule ligne droite (ou un hyperplan en dimensions plus élevées). Les sorties du XOR pour deux entrées binaires sont [22] :

Entrée A	**Entrée B**	**Sortie XOR**
0	0	0
0	1	1
1	0	1
1	1	0

TABLE 2.1 – Table de vérité de la porte XOR

2.4.10 Perceptron à couche unique vs multicouches (MLP)

On distingue deux types de Perceptron : à couche unique et multicouches. Un Perceptron à couche unique peut apprendre uniquement des fonctions linéaires séparables. Un Perceptron à multiples couches, également connu sous le nom de réseau neuronal « feed-forward », permet de dépasser cette limite et offre une puissance de calcul accrue. Il est également envisageable de fusionner plusieurs Perceptrons afin de former un mécanisme puissant. [22]

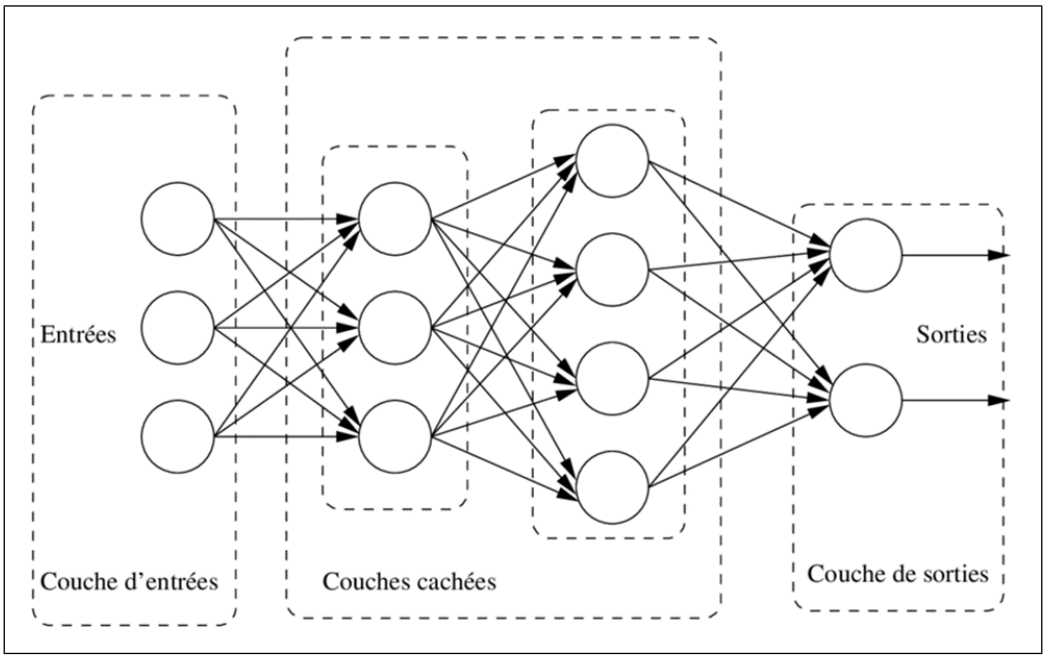

FIGURE 2.11 – MLP. [80]

2.4.11 Les fonctions d'activation

Les fonctions d'activation dans les réseaux de neurones sont utilisées pour calculer la somme pondérée des entrées et des biais, ce qui détermine si un neurone peut être activé ou non. Elles manipulent les données présentées et produisent une sortie pour le réseau de neurones qui contient les paramètres des données. Les fonctions d'activation sont également appelées fonctions de transfert dans certaines littératures. Elles peuvent

être linéaires ou non linéaires selon la fonction qu'elles représentent et sont utilisées pour contrôler la sortie des réseaux de neurones dans différents domaines. [37]

2.5 les differentes fonctions d'ativation

2.5.1 ReLU

La fonction Rectified Linear Unit est la méthode d'activation la plus fréquemment employée dans le domaine de l'apprentissage profond. [38]

$$\mathrm{ReLU}(x) = \max(0, x)$$

Elle affiche x si x dépasse 0, 0 sinon. En d'autres termes, c'est la limite entre x et 0 :

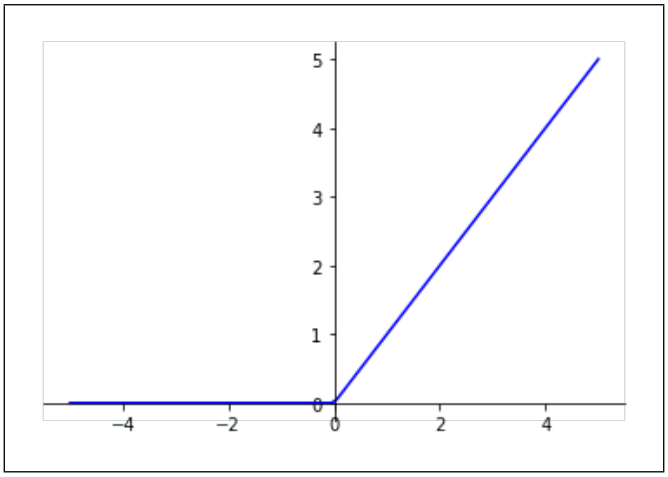

FIGURE 2.12 – Fonction d'activation RELU. [38]

2.5.2 sigmoide

La fonction Sigmoïde est la fonction d'activation employée dans la dernière couche d'un réseau de neurones conçu pour réaliser une tâche de classification binaire. [38]

Elle attribue une valeur allant de 0 à 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

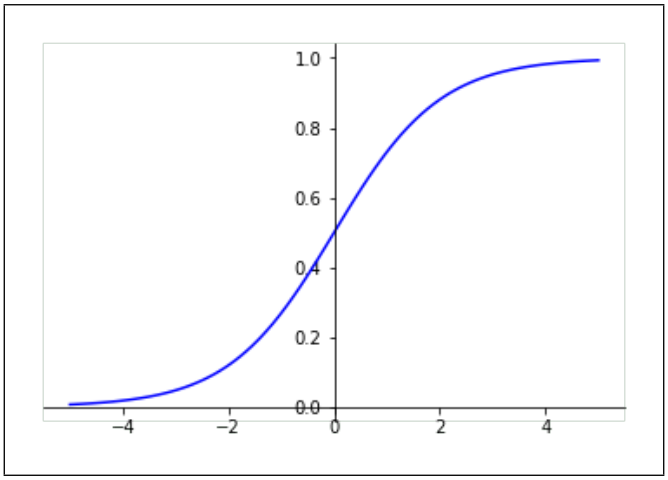

FIGURE 2.13 – Fonction d'activation Sigmoid. [38]

2.5.3 softmax

En dernière couche d'un réseau de neurones, la fonction Softmax est utilisée pour accomplir une tâche de classification multi-classes.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Pour chaque sortie, Softmax affiche un résultat compris entre 0 et 1. En outre, si ces sorties sont combinées entre elles, le résultat est de 1. [38]

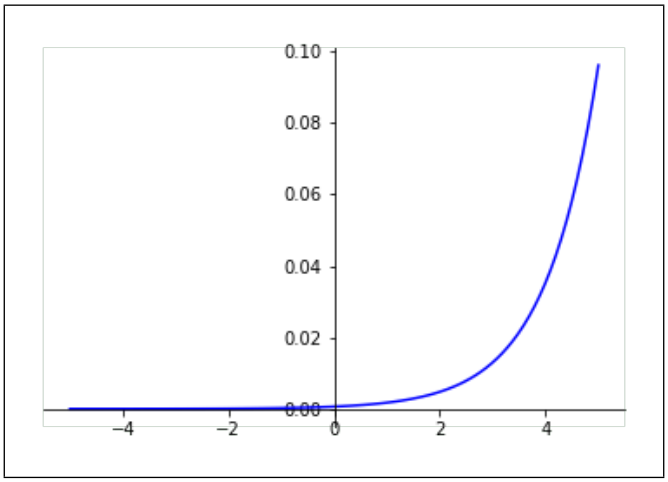

FIGURE 2.14 – Fonction d'activation Softmax. [38]

2.5.4 tanh

Les valeurs d'entrée peuvent être normalisées grâce à la fonction tanh. On peut aussi la substituer à la fonction Sigmoïde dans la dernière couche d'un modèle de classification binaire. [38]

Elle affiche un score compris entre -1 et 1.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

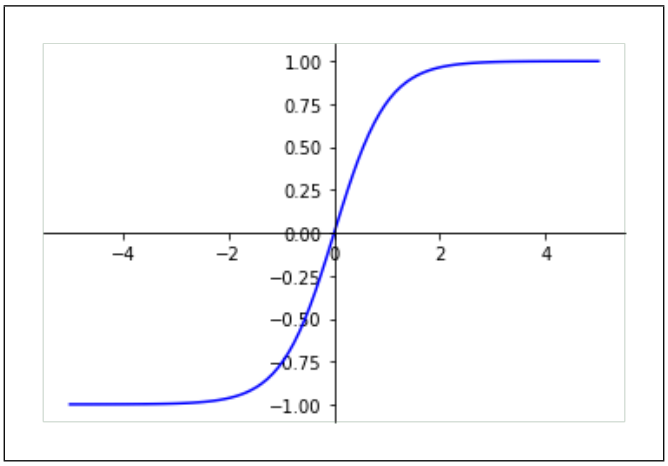

FIGURE 2.15 – Fonction d'activation tanh. [38]

2.6 Réseau de neurones convolutifs - CNN

En apprentissage profond, un réseau de neurones convolutif (CNN/ConvNet) est une catégorie de réseaux de neurones profonds, spécialement conçu pour traiter des données structurées en grille, comme les images, en exploitant les propriétés spatiales locales des données. Les réseaux convolutifs utilisent un processus appelé convolution, pour extraire des caractéristiques à partir des données d'entrée, ce qui permet de capturer des motifs locaux comme les bords, les textures et les formes. Ces réseaux sont particulièrement efficaces pour les tâches de reconnaissance d'images, de détection d'objets, de segmentation d'images et d'autres applications liées à la vision par ordinateur. Les CNN sont caractérisés par leur capacité à apprendre automatiquement des représentations hiérarchiques des données, où les couches profondes capturent des caractéristiques de plus en plus complexes et abstraites.

Un réseau de neurones convolutif (CNN) possède plusieurs couches cachées qui aident à extraire des informations à partir d'une image. Les quatre couches importantes dans un CNN sont :

— Couche de convolution
— Couche ReLU
— Couche de pooling
— Couche entièrement connectée

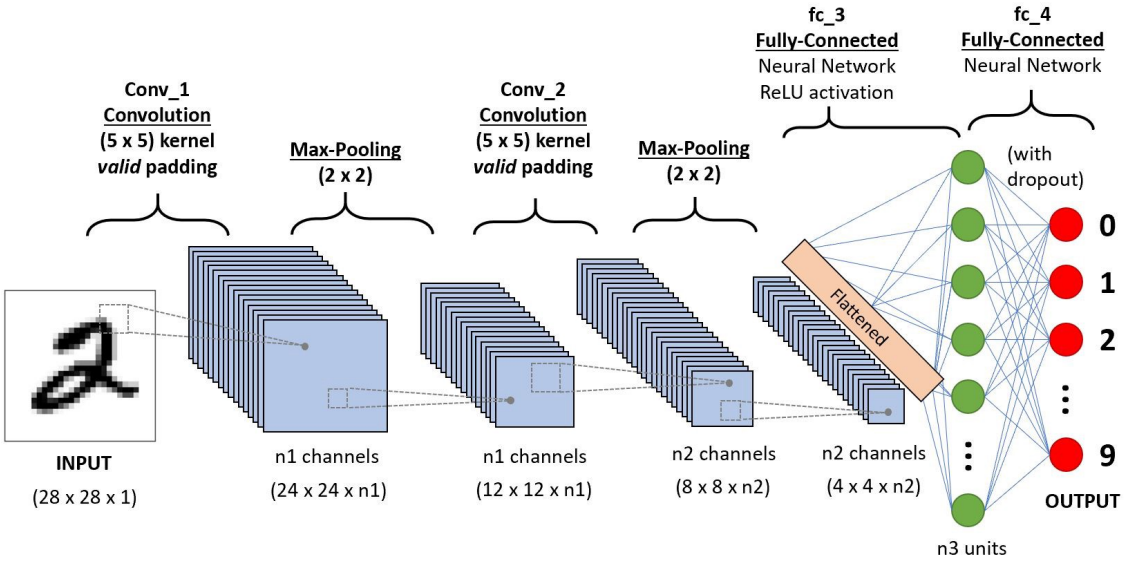

FIGURE 2.16 – Architecture des premiers réseaux de neurones convolutifs. [24]

2.6.1 Couche convolutive (Convolutional Layer)

Son objectif est d'appliquer un filtre de convolution à l'image afin de détecter les caractérisques de l'image. Une image passe à travers une serie de filtres, ou noyaux de convolution, ce qui génère de nouvelles images connues sous le nom de cartes de convolutions. Des filtres intermédiaires permettent de diminuer la résolution de l'image en effectuant une opération de maximum local. Au final, les cartes de convolutions sont alignées et combinées en un vecteur de détails. [33]

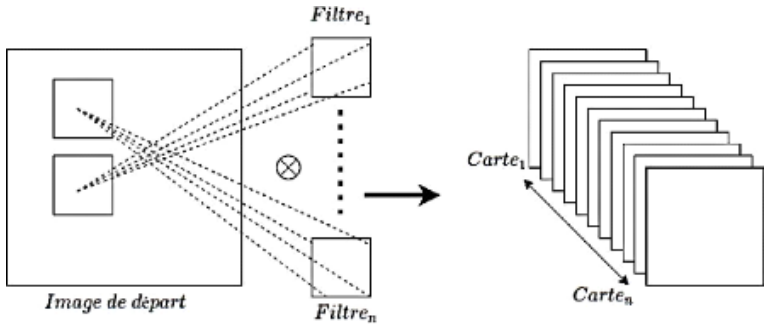

FIGURE 2.17 – Représentation générale des cartes de caractéristiques.[33]

2.6.2 Filtre (Kernel)

Un filtre (ou kernel) est une petite matrice de poids utilisée pour scanner une entrée (comme une image) et effectuer l'opération de convolution. Le filtre "glisse" sur l'entrée, élément par élément, et à chaque position, il effectue une multiplication élément par élément suivie d'une somme pour produire une seule valeur en sortie. Cette valeur constitue une partie de la carte de caractéristiques (feature map) générée. [24]

Détails du Filtre

Dimensions

La taille du filtre est typiquement beaucoup plus petite que celle de l'entrée. Par exemple, dans le traitement d'images, les tailles courantes des filtres sont 3×3, 5×5, ou 7×7. [24]

Poids Les éléments du filtre sont des poids ajustables, qui sont appris pendant l'entraînement du réseau pour détecter des motifs spécifiques dans l'entrée. [24]

Stride et Padding

Stride : La taille du pas avec lequel le filtre se déplace sur l'entrée. Un stride de 1 signifie que le filtre se déplace d'un pixel à la fois, tandis qu'un stride de 2 signifie qu'il se déplace de deux pixels à la fois. [24]

Padding : Les bordures ajoutées autour de l'entrée pour contrôler la taille des cartes de caractéristiques. Un padding de zéro (zero-padding) est souvent utilisé pour maintenir la dimension de l'entrée après convolution.[24]

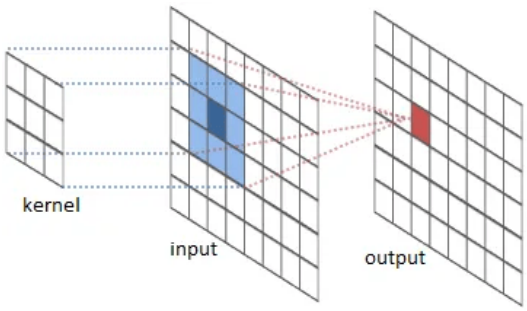

FIGURE 2.18 – Image convolution — filtrage du kernel. [69]

2.6.3 Couche de correction ReLU

Les cartes de convolution sont passées par une couche d'activation non linéaire, comme la Rectified Linear Unit (ReLu), qui consiste à remplacer les nombres négatifs des images filtrées par des zéros d'activation.

2.6.4 Couche de pooling

la couche de pooling qui implique de réduire progressivement la taille de l'image en ne conservant que les informations les plus essentielles, comme pour chaque groupe de 4 pixels, le pixel ayant la valeur maximale (Max Pooling, le plus populaire) ou la moyenne des pixels (AVG pooling).[40]

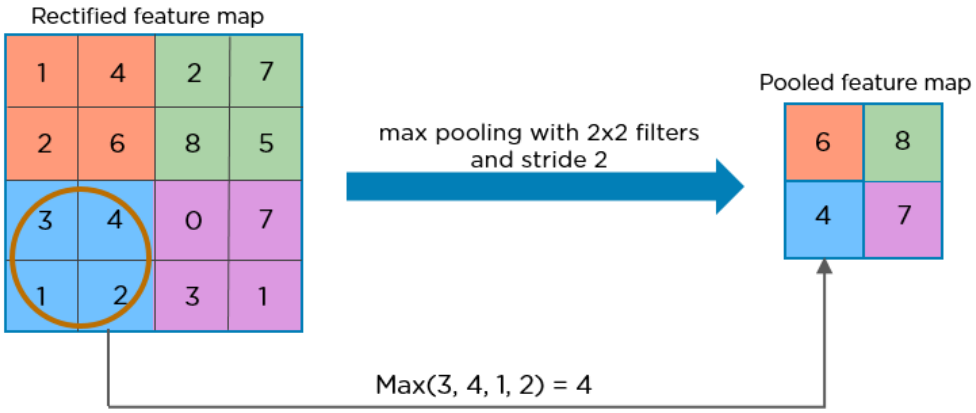

FIGURE 2.19 – Filtres de pooling. [40]

En utilisant la méthode de pooling, on réduit la quantité de paramètres et de calculs dans le réseau, ce qui permet de contrôler le sur-apprentissage.

Une fois que les caractéristiques des entrées ont été extraites, on connecte un perceptron ou un MLP (Multi Layer perceptron), également connu sous le nom de Fully connected, à la fin du réseau.

2.6.5 fully connected Layer

En entrée, elle reçoit un vecteur contenant les pixels aplatis de toutes les images filtrées, corrigées et réduites grâce au pooling. [70]

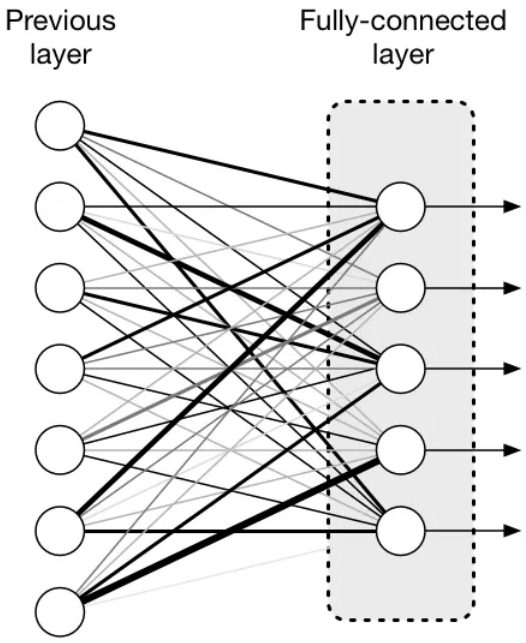

FIGURE 2.20 – couche entièrement connectée. [70]

2.6.6 Les differentes utilisation d'un CNN

Les buts sont multiples et spécifiques aux tâches de traitement des images et des vidéos. Voici une liste des principaux objectifs d'un CNN :

Extraction des caractéristiques : Les CNN sont conçus pour extraire automatiquement les caractéristiques importantes des images, telles que les contours, les textures, et les formes, à différents niveaux d'abstraction.

Classification d'images : Un des objectifs primaires des CNN est de classer les images en différentes catégories. Par exemple, reconnaître si une image contient un chat, un chien, ou une voiture.

Détection d'objets : Les CNN peuvent identifier et localiser plusieurs objets dans une image. Ceci est souvent utilisé dans des applications telles que la reconnaissance faciale, la surveillance, et les voitures autonomes.

Segmentation d'images : Les CNN peuvent être utilisés pour la segmentation sémantique ou par instance, où chaque pixel de l'image est classé en fonction de l'objet auquel il appartient.

Reconnaissance de motifs et de structures : Les CNN sont capables de reconnaître des motifs complexes et des structures dans les données visuelles, ce qui est utile dans des domaines comme la médecine pour l'analyse d'images médicales.

Réduction de la dimensionnalité : En extrayant des caractéristiques importantes et en ignorant les détails superflus, les CNN peuvent réduire la dimensionnalité des données, facilitant ainsi l'analyse et le traitement des informations visuelles.

Robustesse aux variations : Les CNN visent à être robustes aux variations des images, telles que les changements d'éclairage, les rotations, et les échelles, ce qui est crucial pour de nombreuses applications pratiques.

Amélioration des images : Les CNN peuvent être utilisés pour des tâches d'amélioration des images telles que la super-résolution, la suppression de bruit, et la restauration d'images.

Analyse vidéo : Les CNN peuvent également être utilisés pour analyser des séquences vidéo, permettant des applications comme la détection de mouvement, la reconnaissance d'activités, et le suivi d'objets en mouvement.

Prévision de séries temporelles d'images : Dans certaines applications, les CNN peuvent être utilisés pour prédire des séries d'images futures, par exemple dans la surveillance météorologique ou la prévision du trafic.

Ces objectifs sont réalisés grâce à une architecture spécifique de couches convolutives, de pooling, de normalisation et de couches entièrement connectées, qui permettent au réseau d'apprendre à partir de grandes quantités de données visuelles et de généraliser ces apprentissages à de nouvelles images.

2.7 Les réseaux neuronaux récurrents RNN

Les réseaux neuronaux récurrents sont très efficaces pour traiter des entrées de taille variable grâce à leur mémoire à court terme. Ils facilitent également une meilleure compréhension

du contexte en traitant les paquets de données presque simultanément. Mais cette mémoire à court terme n'est pas suffisante pour certaines tâches en raison du problème bien connu du "Vanishing Gradient Problem". [43]

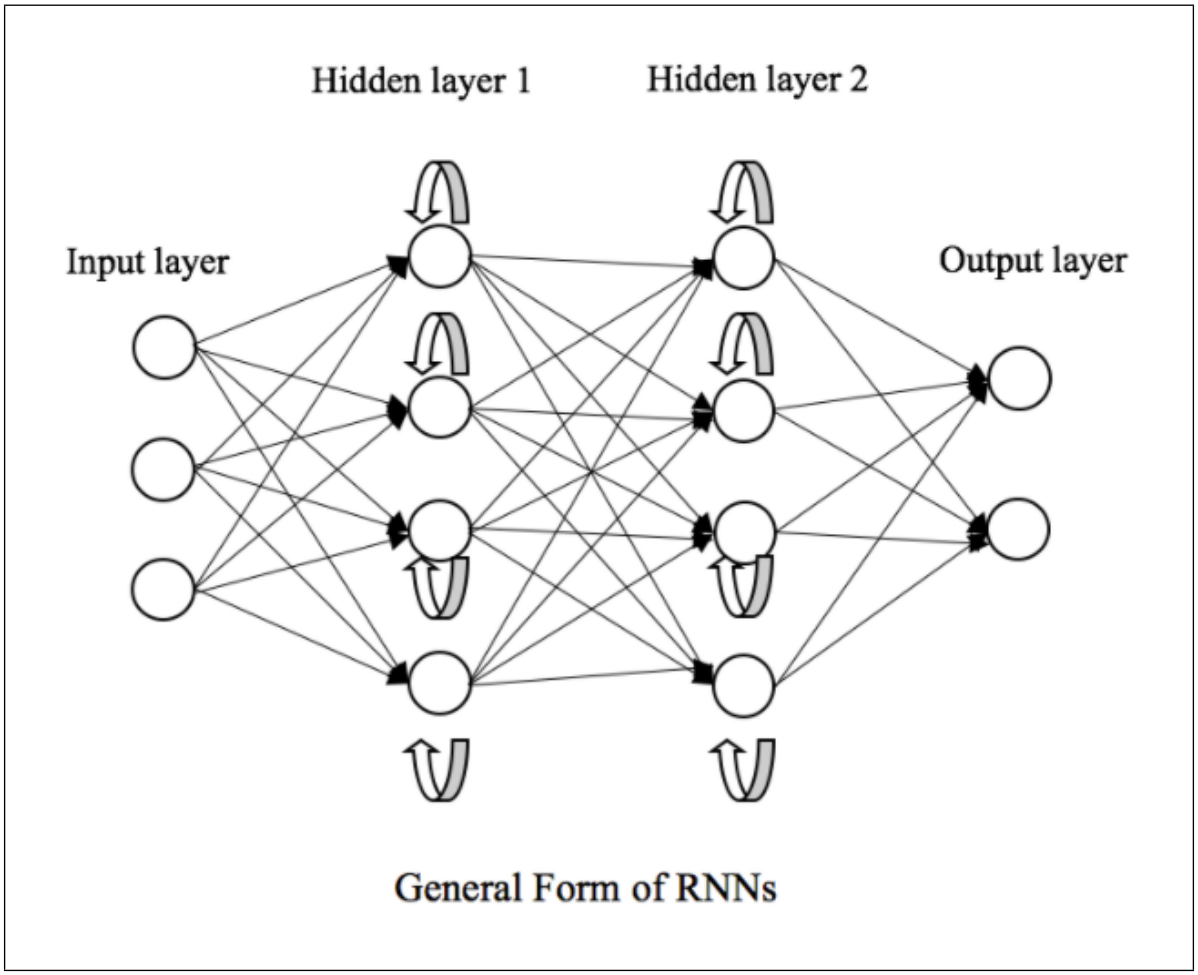

FIGURE 2.21 – Schéma d'une réseau neuronal récurrent. [71]

2.7.1 Le problème Vanishing Gradient

Le problème du "vanishing gradient" est un défi courant lors de l'apprentissage des réseaux neuronaux récurrents (RNN). Il se produit lorsque les RNN tentent de modéliser des dépendances à long terme dans des séquences. Lorsque les gradients sont propagés au cours de la rétropropagation, leur norme tend à diminuer de manière exponentielle, ce qui entraîne des mises à jour de poids négligeables pour les neurones situés près du début de la séquence. Par conséquent, les RNN ont du mal à apprendre et à saisir les dépendances à long terme. [43]

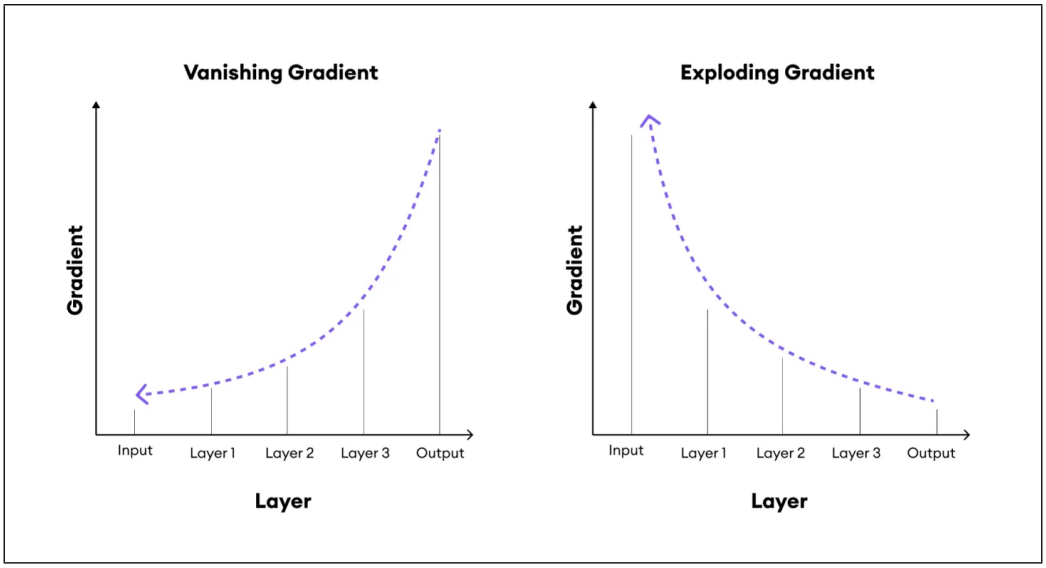

FIGURE 2.22 – Schémas de la disparition et de l'explosion du gradient. [75]

2.8 Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) est une architecture de réseau neuronal récurrent (RNN) couramment utilisée dans l'apprentissage profond. Elle est particulièrement efficace pour capturer les dépendances à long terme, ce qui la rend idéale pour les tâches de prédiction de séquences. Contrairement aux réseaux neuronaux traditionnels, le LSTM intègre des connexions de rétroaction, ce qui lui permet de traiter des séquences entières de données, plutôt que des points de données individuels. Cela le rend très efficace pour comprendre et prédire des modèles dans des données séquentielles telles que des séries temporelles, des textes et des discours. La LSTM est un outil puissant dans le domaine de l'intelligence artificielle et de l'apprentissage profond, qui permet de réaliser des percées dans divers domaines en découvrant des informations précieuses à partir de données séquentielles. [41]

2.8.1 L'architecture du LSTM

Dans l'ensemble, LSTM fonctionne de manière très similaire à une cellule RNN. L'architecture du réseau LSTM se compose de trois parties. Ces trois parties d'une unité LSTM sont appelées portes et contrôlent le flux d'informations entrant et sortant de la cellule de mémoire ou de la cellule LSTM. La première porte est appelée Forget Gate. Elle détermine si l'information provenant de l'horodatage précédent doit être mémorisée ou si elle n'est pas pertinente et peut être oubliée. La deuxième porte est connue sous le nom de Input Gate, la cellule essaie d'apprendre de nouvelles informations à partir de l'entrée de cette cellule. Enfin, la troisième porte est Output Gate, la cellule transmet les informations mises à jour de l'horodatage actuel à l'horodatage suivant.

Une unité LSTM composée de ces trois portes et d'une cellule de mémoire ou cellule LSTM peut être considérée comme une couche de neurones dans un réseau neuronal feedforward traditionnel, où chaque neurone possède une couche cachée et un état actuel. [41]

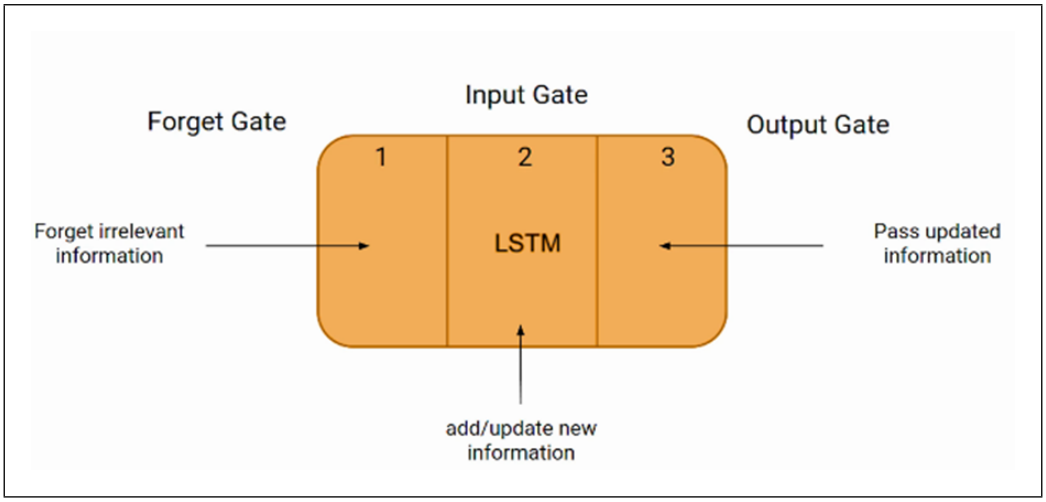

FIGURE 2.23 – Cellule LSTM avec barrières [41].

LSTM a également un état caché, où H(t-1) est l'état caché de l'horodatage précédent et Ht est l'état caché de l'horodatage actuel. En outre, LSTM possède également un état cellulaire représenté par C(t-1) et C(t) pour l'horodatage précédent et l'horodatage actuel, respectivement. On appelle l'état caché la mémoire à court terme et l'état de la cellule la mémoire à long terme. [41]

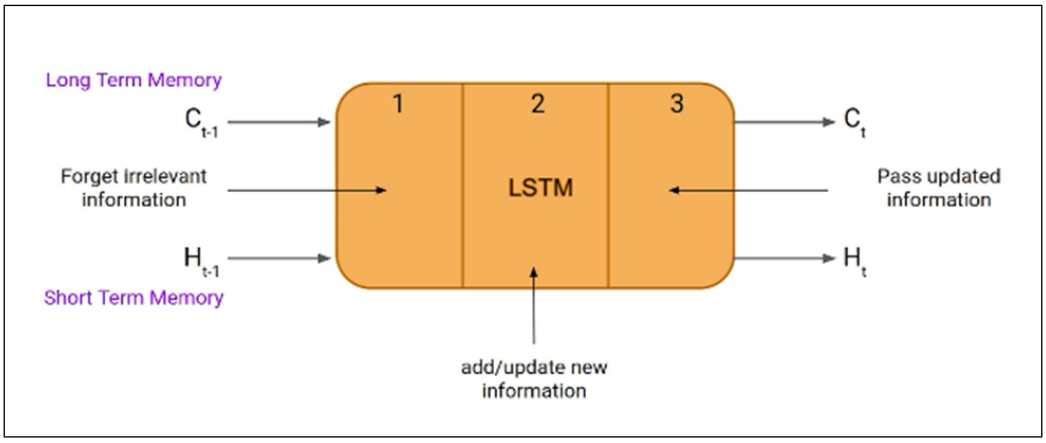

FIGURE 2.24 – Cellule LSTM avec états cachés et cellules.[41]

2.8.2 Forget Gate

L'information qui n'est plus utile dans l'état de la cellule est supprimée à l'aide de Forget Gate. Deux entrées xt (entrée actuelle) et ht-1 (sortie de la cellule précédente) sont transmises à la porte et multipliées par des matrices de poids, suivies de l'ajout d'un biais. La résultante passe par une fonction d'activation qui produit une sortie binaire. Si la sortie est 0 pour un état cellulaire donné, l'information est oubliée, et si la sortie est 1, l'information est conservée pour une utilisation ultérieure. [42]

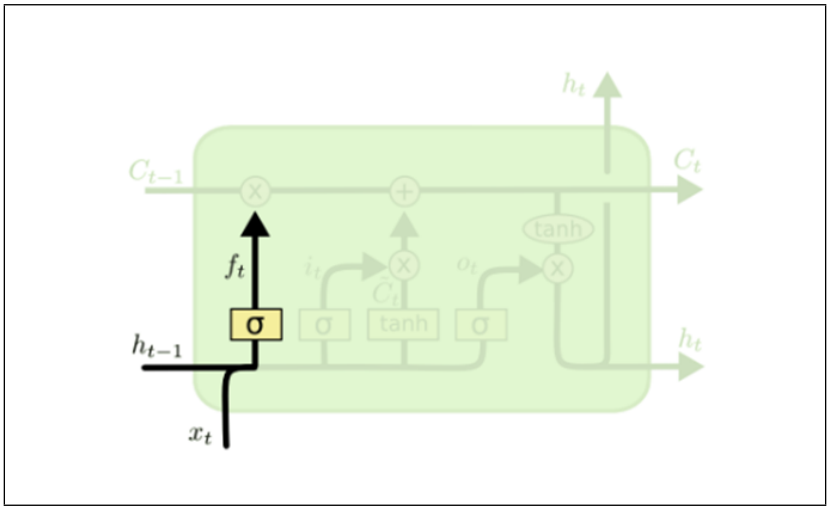

FIGURE 2.25 – Forget Gate (opérateur d'oubli d'informations).[42]

L'équation de la Forget Gate est la suivante :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.1}$$

où :

- W_f représente la matrice de poids associée à la porte d'oubli.
- $[h_{t-1}, x_t]$ représente la concaténation de l'entrée actuelle (x_t) et de l'état caché précédent (h_{t-1}) : l'opération de combinaison avant le passage de la porte d'oubli
- b_f est le biais associé à la porte d'oubli.
- σ est la fonction d'activation sigmoïde. [42]

2.8.3 Input gate

L'ajout d'informations utiles à l'état de la cellule est effectué par Input Gate. Tout d'abord, l'information est régulée à l'aide de la fonction sigmoïde et les valeurs à stocker sont filtrées de manière similaire à la porte d'oubli à l'aide des entrées ht-1 et xt. Ensuite, la fonction tanh est utilisée pour créer un vecteur avec une sortie de -1 à +1 qui contient toutes les valeurs possibles de ht-1 et xt. Enfin, les valeurs du vecteur et les valeurs régulées sont multipliées pour obtenir les informations utiles. [42]

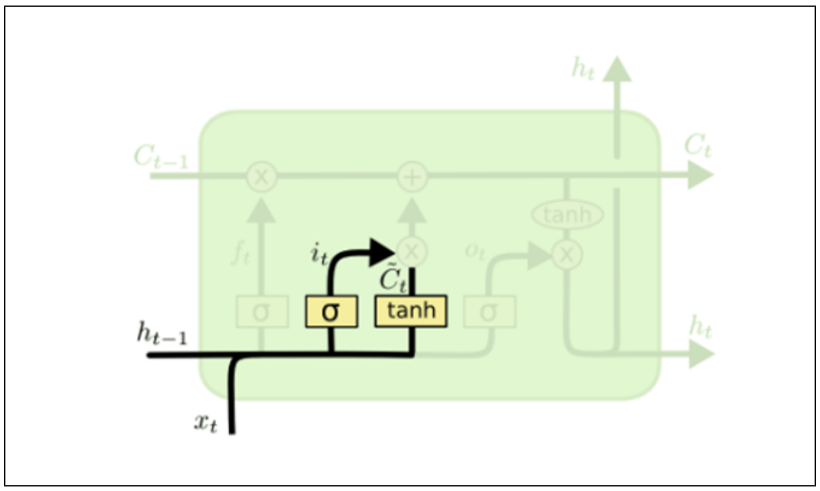

FIGURE 2.26 – Input Gate (fopérateur d'ajout d'informations).[42]

L'équation de la Input Gate est la suivante :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.3}$$

Nous multiplions l'état précédent par ft, en ignorant l'information que nous avons précédemment choisi d'ignorer. Ensuite, nous incluons Ceci représente les valeurs candidates mises à jour, ajustées par le montant que nous avons choisi pour mettre à jour chaque valeur d'état. [42]

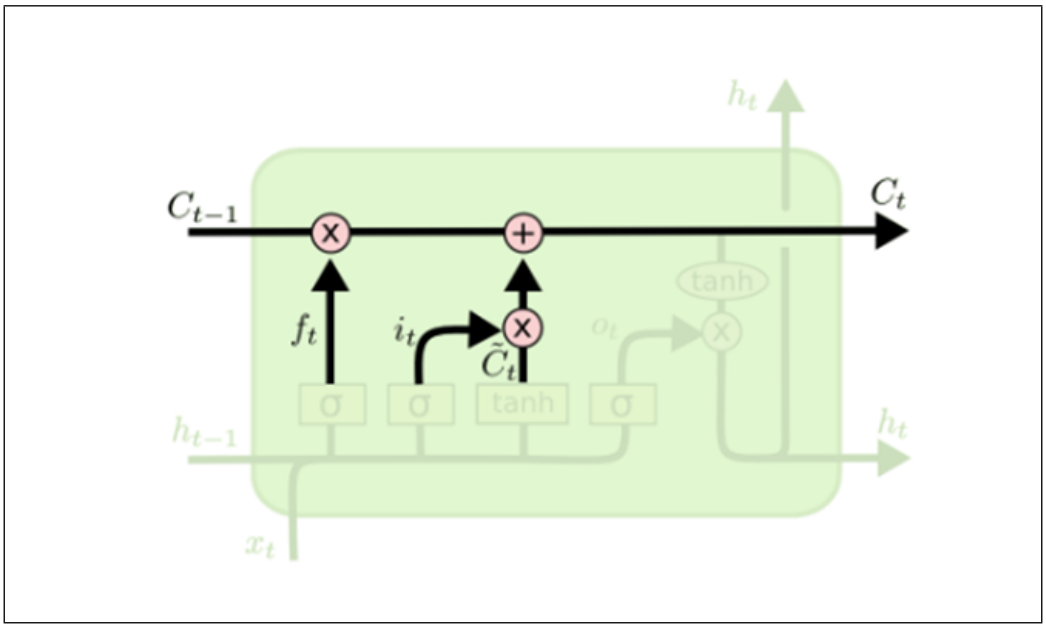

FIGURE 2.27 – Mise à jour de la mémoire C. [42]

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{2.4}$$

2.8.4 Output Gate

Output gate permet d'extraire les informations utiles de l'état actuel de la cellule pour les présenter en sortie. Tout d'abord, un vecteur est généré en appliquant la fonction tanh à la cellule. Ensuite, l'information est régulée à l'aide de la fonction sigmoïde et filtrée par les valeurs à mémoriser à l'aide des entrées ht-1 et xt. Enfin, les valeurs du vecteur et les valeurs régulées sont multipliées pour être envoyées comme sortie et entrée à la cellule suivante. [42]

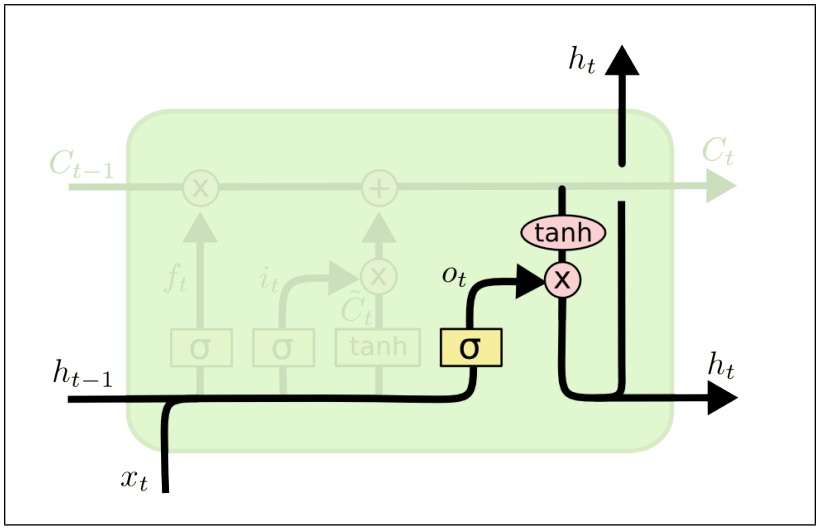

FIGURE 2.28 – Output Gate (sortie de la couche cachée). [42]

L'équation de la porte de sortie est la suivante :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2.6}$$

2.9 Mécanismes d'entraînement des modèles DL

2.9.1 Propagation avant -Forward Propagation

La propagation vers l'avant (ou forward propagation) est le processus par lequel les données d'entrée sont introduites dans un réseau de neurones, et traversent les différentes couches du réseau pour générer une prédiction ou une sortie. Voici les étapes principales :

Chaque couche du réseau (qu'il s'agisse de couches entièrement connectées, convolutionnelles, récurrentes, etc.) reçoit en entrée les données provenant de la couche précédente ou directement des données d'entrée. Chaque neurone dans une couche calcule une combinaison linéaire des entrées pondérées par les poids associés à ce neurone. À cette combinaison linéaire, une fonction d'activation non linéaire est appliquée pour introduire de la non-linéarité dans le modèle. Les sorties de chaque couche deviennent les entrées de la couche suivante jusqu'à ce que la prédiction finale soit obtenue. [25]

2.9.2 Fonction de Perte - Loss function

Les fonctions de perte mesurent à quel point les prédictions d'un modèle sont éloignées des vraies valeurs (labels) attendues. Voici quelques types courants de fonctions de perte :

— **Mean Squared Error (MSE)** : Calcul de la moyenne des carrés des différences entre les prédictions et les vraies valeurs. Utile pour les problèmes de régression.

— **Cross-Entropy** : Également connue sous le nom de log loss, elle est souvent utilisée pour les problèmes de classification où les prédictions sont des probabilités.

Elle mesure la divergence entre la distribution des prédictions et celle des vraies étiquettes.

L'importance de la fonction de perte réside dans le fait qu'elle guide le processus d'optimisation du modèle pendant l'entraînement. Minimiser la fonction de perte est l'objectif principal de l'entraînement afin d'ajuster les poids du réseau pour obtenir des prédictions plus précises.[26]

2.9.3 La Retropropagation (Backpropagation)

La rétropropagation (ou backpropagation) est l'algorithme clé utilisé pour calculer les gradients de la fonction de perte par rapport aux poids du réseau, en vue de mettre à jour ces poids pour minimiser la perte. Voici les étapes principales de la rétropropagation :

— **Calcul des gradients** : Utilisation de la règle de la chaîne pour calculer les gradients de la fonction de perte par rapport à chaque poids du réseau.[27]

— **Mise à jour des poids** : Utilisation des gradients calculés pour mettre à jour les poids du réseau à l'aide d'un algorithme d'optimisation, tel que l'optimiseur Adam. [27]

2.9.4 Optimiseurs

Les optimiseurs sont des algorithmes qui ajustent les poids du modèle pendant l'entraînement pour minimiser la fonction de perte. Voici quelques optimiseurs couramment utilisés :

2.9.5 Descente du Gradient

La descente du gradient est un algorithme d'optimisation utilisé pour minimiser une fonction de coût (ou fonction de perte) en ajustant itérativement les paramètres d'un modèle. Voici les points clés :

Gradient : À chaque itération, le gradient de la fonction de coût par rapport aux paramètres du modèle est calculé. Le gradient indique la direction dans laquelle les paramètres doivent être ajustés pour réduire la fonction de coût.[30]

Pas d'apprentissage (learning rate) : Un hyperparamètre crucial dans la descente du gradient, il détermine la taille des pas effectués dans la direction opposée au gradient. Un pas trop petit peut ralentir la convergence, tandis qu'un pas trop grand peut entraîner une divergence.[30]

Optimisation stochastique : La descente du gradient stochastique (SGD) est une variante où les gradients sont calculés et les paramètres sont mis à jour par mini-lots d'échantillons plutôt que sur l'ensemble des données, ce qui accélère le processus sur de grands ensembles de données [30]

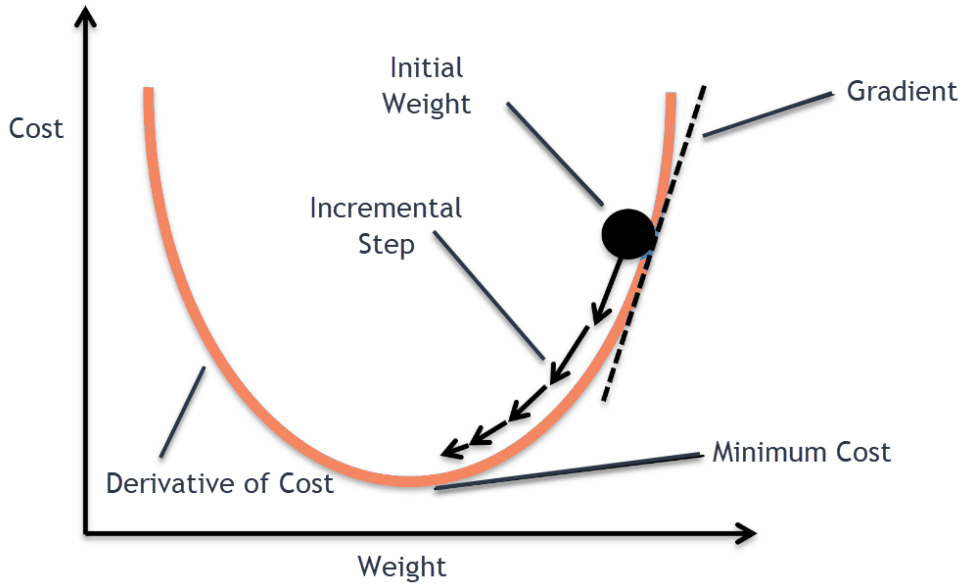

FIGURE 2.29 – Correction des parametre avec descente du gradient. [31]

— **Adam (Adaptive Moment Estimation)** : Un optimiseur populaire qui combine les avantages des méthodes adaptatives du taux d'apprentissage et des moments pour converger plus efficacement.[30]

2.9.6 Fonctions d'Évaluation

Les fonctions d'évaluation mesurent les performances d'un modèle une fois qu'il est entraîné. Voici quelques métriques couramment utilisées :

— **Précision (Accuracy)** : Le pourcentage de prédictions correctes par rapport au nombre total de prédictions. [29]

— **Précision, Rappel et F1-score** : Des métriques utilisées principalement pour évaluer les performances des modèles de classification en tenant compte des faux positifs, des faux négatifs et de la sensibilité de la prédiction.[29]

2.10 conclusion

Ce chapitre a fourni une vue d'ensemble complète de l'intelligence artificielle (IA) dans l'analyse des scènes. Nous avons couvert son histoire, ses principales technologies et techniques, notamment l'apprentissage automatique et l'apprentissage profond. Nous avons expliqué les réseaux de neurones artificiels, les perceptrons, les fonctions d'activation et les réseaux de neurones convolutifs (CNN). Enfin, nous avons abordé les réseaux neuronaux récurrents (RNN) et les mémoires à court terme (LSTM). Ce panorama offre une compréhension des bases et des avancées en IA appliquées à l'analyse des scènes.

Chapitre 3

Méthodes d'analyse des scenes en video-surveillance

3.1 Introduction

Ces dernières années, le domaine de la détection de la violence dans la vidéosurveillance a fait l'objet d'une attention considérable en raison de son importance critique dans l'amélioration de la sûreté et de la sécurité publiques.

Dans ce chapitre, nous explorons les différentes méthodes utilisées pour la détection de la violence dans la vidéosurveillance. En examinant un large éventail d'études, nous visons à mettre en évidence les points forts et les limites des différentes méthodologies et à fournir une perspective claire sur les progrès réalisés dans ce domaine

3.2 Les approches et méthodes utilisées pour la détection des scenes : Les methodes de traitement d'image

3.2.1 La soustraction de l'arrière-plan

La soustraction de l'arrière-plan (Background subtraction), est une technique permettant de détecter des objets en mouvement dans une séquence d'images provenant d'une caméra statique. Elle permet d'extraire l'avant-plan (objet en mouvement) et l'arrière-plan (objet immobile) de l'image pour un traitement ultérieur, tel que la reconnaissance d'objets. L'idée de base de la technique de soustraction d'arrière-plan est de soustraire l'arrière-plan du premier plan, ce qui nous aide à identifier les objets en mouvement. Les techniques de soustraction d'arrière-plan renvoient généralement un masque qui peut être considéré comme l'avant-plan dans un ordre relatif. Cette technique peut rapidement fournir des informations sur les objets en mouvement dans la vidéo, sans trop d'efforts ni de puissance de calcul, qui peuvent être utilisées comme données d'entrée pour d'autres algorithmes. Bien qu'il s'agisse d'une technique ancienne, la soustraction de l'arrière-plan reste un choix pragmatique pour de nombreuses applications liées à la vidéo, car elle offre une combinaison de simplicité, d'efficacité et d'aptitude au traitement en temps réel.[49]

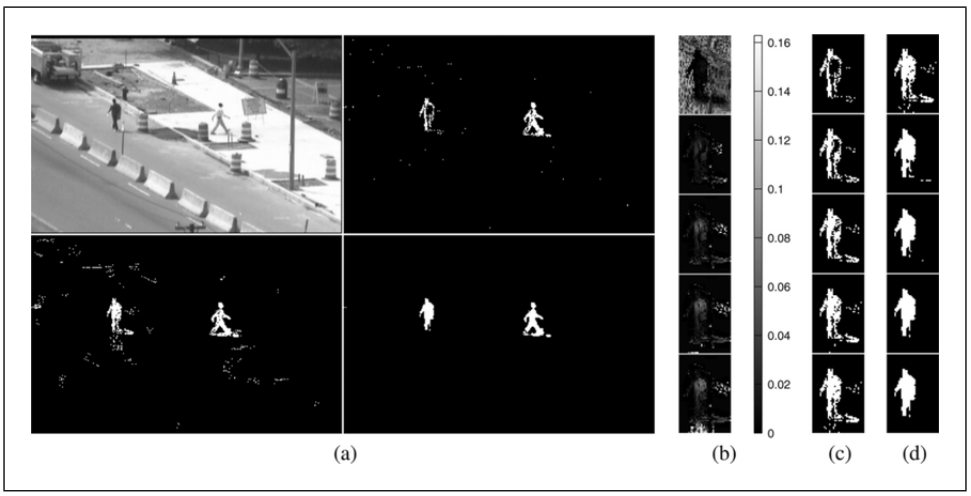

FIGURE 3.1 – Soustraction de l'arrière-plan. [50]

3.2.2 Le flux optique

Le flux optique est une technique utilisée pour décrire le mouvement des images. Elle est basée sur le principe du flux optique, qui est le mouvement des objets entre les images successives d'une séquence. Ce mouvement est causé par le mouvement relatif entre l'objet et la caméra. Le flux optique est typiquement appliqué à une série d'images séparées par un petit pas de temps, comme des images vidéo. Le flux optique calcule la vitesse des points dans les images et fournit une estimation de l'endroit où les points pourraient se trouver dans la séquence d'images suivante. Il existe deux types principaux de flux optique : le flux optique clairsemé et le flux optique dense. Le flux optique clairsemé utilise les vecteurs de flux de certaines "caractéristiques intéressantes" (par exemple, quelques pixels représentant les bords ou les coins d'un objet) dans l'image pour suivre leurs vecteurs de vitesse. En revanche, le flux optique dense renvoie les vecteurs de flux de l'ensemble de l'image (tous les pixels), chaque pixel pouvant avoir jusqu'à un vecteur de flux. Les vecteurs de flux optique servent d'entrée à une variété de tâches de plus haut niveau qui nécessitent une compréhension de la scène des séquences vidéo. [51]

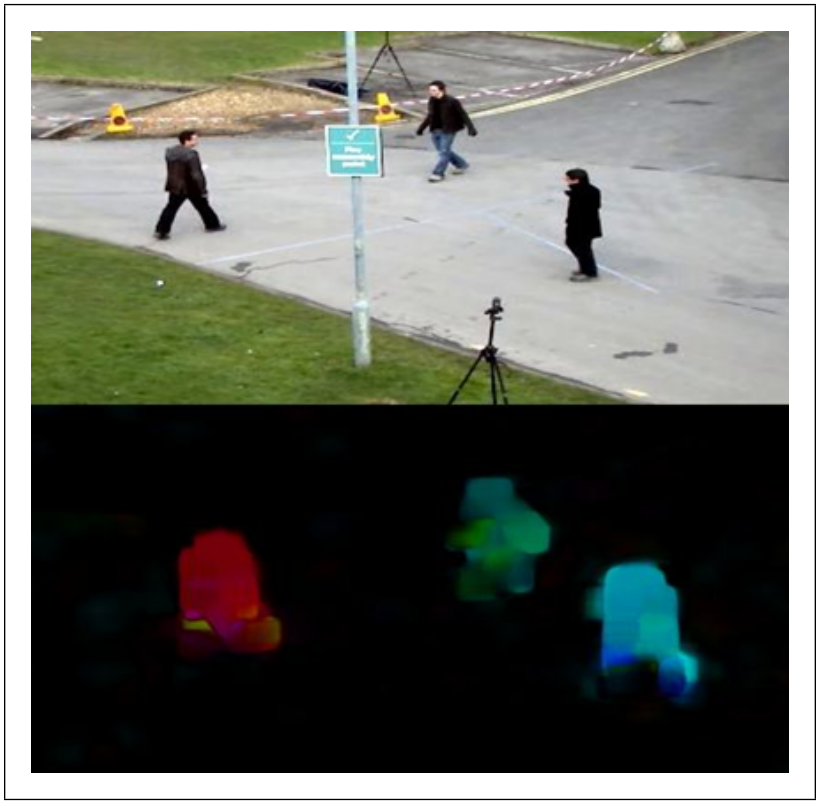

FIGURE 3.2 – Flux optique. [52]

3.2.3 Classificateur de Haar

Un classificateur de Haar est un algorithme d'apprentissage automatique de détection d'objets qui identifie les objets dans une image ou une vidéo, indépendamment de leur échelle dans l'image et de leur emplacement. Par exemple, il peut être utilisé pour trouver des visages dans une image. Cet algorithme est relativement simple et peut fonctionner en temps réel. Il peut être entraîné à détecter divers objets, tels que des voitures, des vélos, des bâtiments, etc. Il utilise une approche de fenêtre en cascade, dans laquelle

les caractéristiques sont calculées dans chaque fenêtre et la fenêtre est classée comme positive ou négative. Le classificateur est entraîné sur un ensemble d'images avec des classes positives et négatives. L'objectif du modèle est de localiser la classe positive dans une image en extrayant les caractéristiques à l'aide des filtres/noyaux de Haar. L'un des principaux avantages des classificateurs en cascade de Haar est leur simplicité, qui permet un apprentissage relativement simple de la détection de divers objets. Bien qu'ils n'atteignent pas le même niveau de précision que les approches plus complexes basées sur l'apprentissage profond, les classificateurs en cascade de Haar restent largement utilisés en raison de leur rapidité et de leur efficacité, en particulier dans les scénarios où les ressources informatiques sont limitées ou les performances en temps réel sont critiques.[53]

FIGURE 3.3 – Détection de visage utilisant la cascade de Haar.[54]

3.3 Les approches utilisant le Deep Learning

3.3.1 Estimation de la pose humaine

L'estimation de la pose humaine est une tâche de vision par ordinateur qui représente l'orientation d'une personne dans un format graphique. Cette technique est largement utilisée pour prédire les positions des parties du corps ou des articulations d'une personne. En général, une technique basée sur un modèle est utilisée pour représenter et déduire les poses du corps humain dans l'espace 2D et 3D. Il s'agit principalement d'un moyen de capturer un ensemble de coordonnées en définissant les articulations du corps humain telles que le poignet, l'épaule, le genou, les yeux, les oreilles, les chevilles et les bras, qui sont des points clés dans les images et les vidéos qui peuvent décrire la pose d'une personne. Lorsqu'une image ou une vidéo est donnée en entrée au modèle d'estimation de la pose, celui-ci identifie les coordonnées de ces parties du corps et articulations détectées en sortie, ainsi qu'un score de confiance indiquant la précision des estimations. [55]

FIGURE 3.4 – Points Clés de la Posture Humaine. [55]

Pipeline de Détection de Violence a. Capture des Images/Vidéos Les données visuelles sont capturées à partir de caméras de surveillance ou de vidéos en direct.

b. Estimation de Pose Un modèle d'estimation de pose (comme OpenPose, PoseNet, ou AlphaPose) est appliqué pour identifier les points clés du corps dans chaque image ou cadre vidéo. Ces points clés représentent les articulations et les segments du corps humain.

c. Extraction des Caractéristiques À partir des points clés, des caractéristiques spécifiques sont extraites, telles que les angles des articulations, les trajectoires des mouvements, et les vitesses des segments du corps.

d. Analyse des Comportements Les caractéristiques extraites sont analysées pour détecter des comportements anormaux ou violents. Des algorithmes de machine learning ou de deep learning, tels que les SVM, les réseaux neuronaux récurrents (RNN), ou les réseaux de neurones à convolution 3D (3D-CNN), sont souvent utilisés pour classifier les comportements. [55]

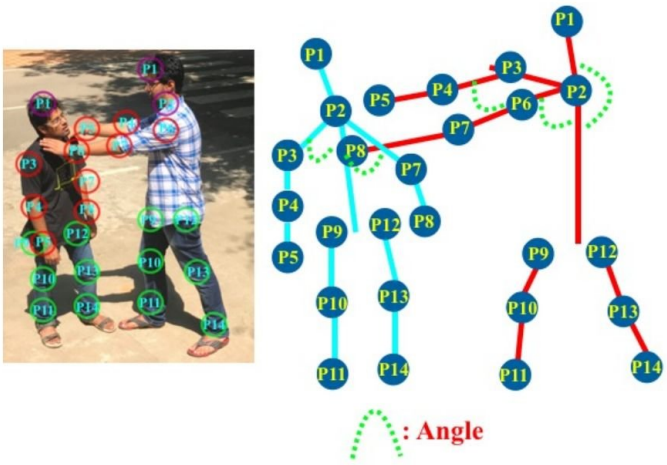

FIGURE 3.5 – Angles du squelette. [32]

3.3.2 La reconnaissance de scenes humaines

La reconnaissance de scènes/actions humaines fait référence au processus d'identification et d'interprétation des activités humaines et des contextes environnementaux à partir de données visuelles, généralement obtenues à partir d'images ou de séquences vidéo. [39]

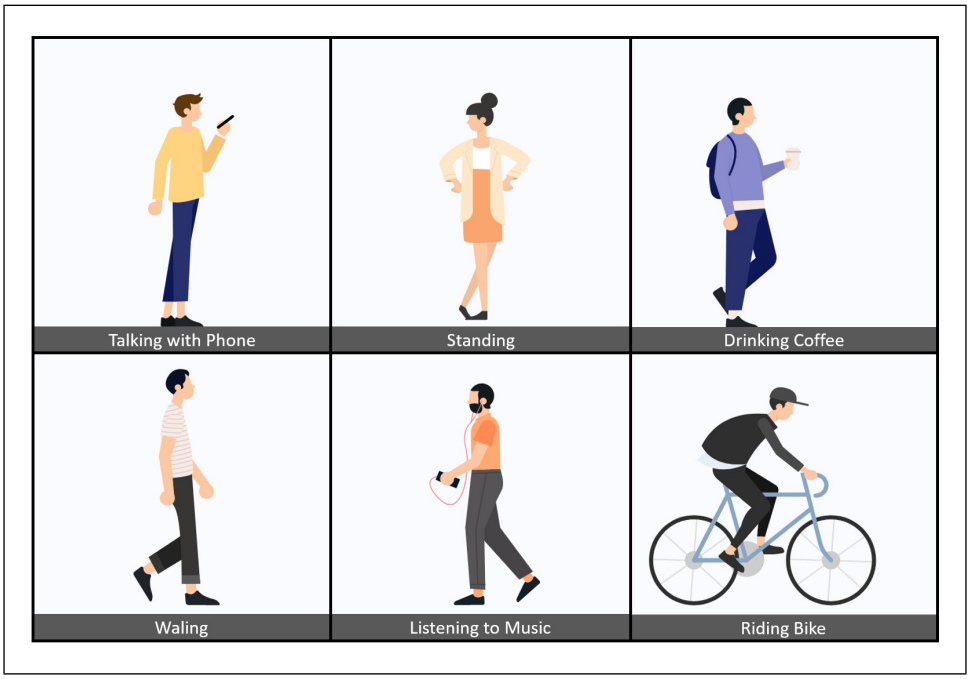

FIGURE 3.6 – Reconnaissance d'actions.[39]

Ce domaine interdisciplinaire implique la vision par ordinateur, l'apprentissage automatique et l'intelligence artificielle pour permettre aux ordinateurs de comprendre et d'analyser les comportements et interactions humains dans divers contextes. Voici les principaux composants et aspects :

Reconnaissance de Scène

Classification de Scène : Identifier le type d'environnement ou de décor où l'action se déroule (par exemple, cuisine, rue, bureau). [39]

Compréhension du Contexte : Comprendre les éléments contextuels dans une scène qui peuvent influencer les actions humaines (par exemple, reconnaître les objets et leurs dispositions). [39]

Reconnaissance d'Action

Détection d'Action : Identifier des actions spécifiques effectuées par des humains, comme marcher, courir, s'asseoir ou interagir avec des objets. Analyse Temporelle : Analyser des séquences d'images au fil du temps pour comprendre la dynamique des actions (par exemple, reconnaître des activités complexes comme cuisiner ou danser). [39]

3.3.3 Techniques et Méthodes

Utiliser des modèles comme les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN) pour classifier et prédire les actions et les scènes.

3.3.4 L'apprentissage par transfert Transfert learning

Transfert learning L'apprentissage par transfert est une technique qui consiste à utiliser un modèle CNN pré-entraîné, formé sur un ensemble de données vaste et diversifié comme ImageNet, puis à l'adapter à une tâche spécifique de vision industrielle. Cette approche permet de tirer parti des connaissances et des caractéristiques apprises par le modèle pré-entraîné, réduisant ainsi le temps et le coût de l'entraînement, tout en obtenant de meilleures performances que l'entraînement depuis zéro.

VGG16 et VGG19 VGG16 est un algorithme de détection et de classification d'objets, entraîné sur le jeu de données Imagenet qui contient plus d'un million d'images capable de classifier 1000 images de 1000 catégories différentes avec une précision de 92,7. C'est l'un des algorithmes populaires pour la classification d'images et il est facile à utiliser avec l'apprentissage par transfert. Ce modèle fonctionne avec un système imbriqué de 3*3 couches convolutives. Chaque couche est centrée sur un élément graphique. Le VGG-19 est une amélioration du VGG-16. [56] [58]

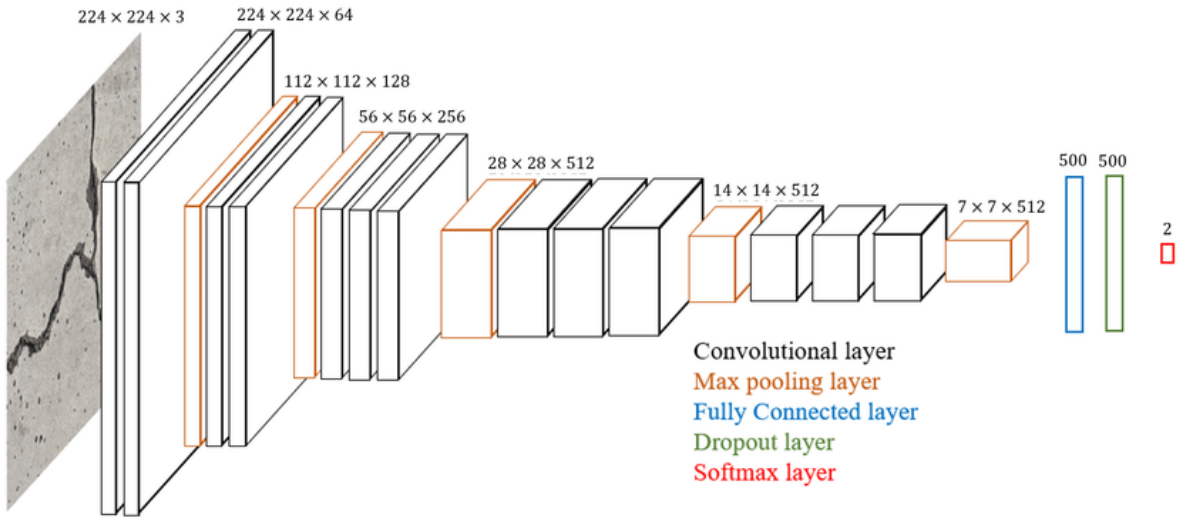

FIGURE 3.7 – Architecture du VGG16. [57]

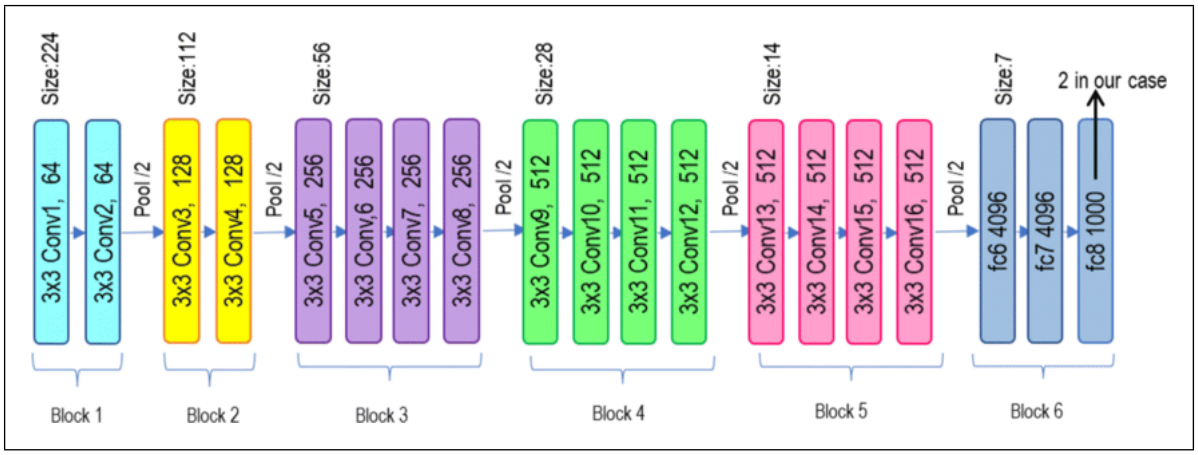

FIGURE 3.8 – Architecture du VGG19. [58]

MobilenetV1

MobilenetV1 est un modèle de réseau de neurones convolutifs (CNN) léger et efficace, conçu spécifiquement pour les appareils mobiles et les applications embarquées. Il se distingue par sa capacité à offrir une bonne précision tout en réduisant considérablement la taille du modèle et les besoins en calcul. MobilenetV1 atteint cet objectif en utilisant des convolutions séparables en profondeur, ce qui décompose les convolutions standards en deux étapes : une convolution en profondeur suivie d'une convolution ponctuelle. Cette approche réduit le nombre de paramètres et les calculs nécessaires, permettant ainsi une exécution rapide et une consommation de ressources réduite, tout en maintenant des performances élevées pour des tâches de classification et de détection d'objets. [74]

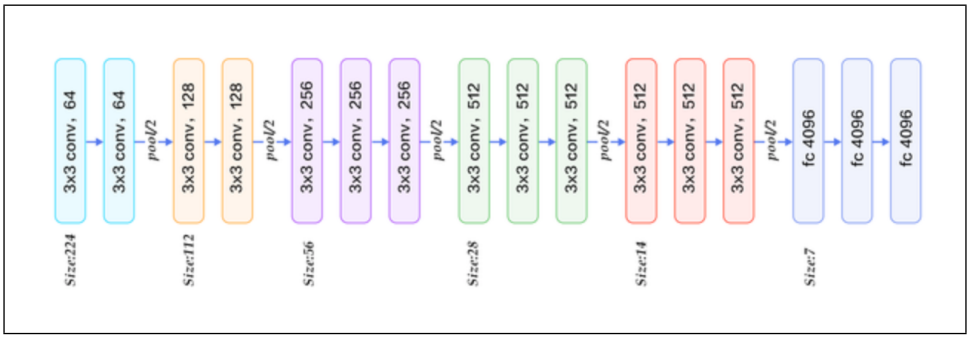

FIGURE 3.9 – Architecture MobilenetV1. [83]

Architecture :

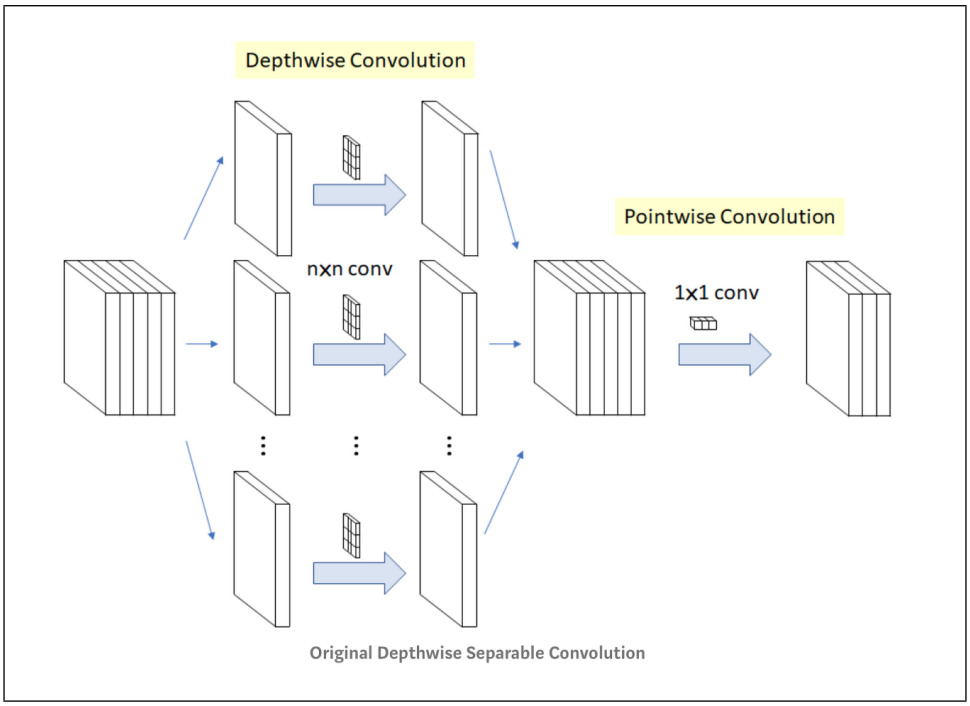

FIGURE 3.10 – Architecture MobilenetV1. [83]

MobilenetV1 V1 repose sur l'utilisation de convolutions séparables en profondeur (depthwise separable convolutions), une approche qui décompose les opérations de convolution standard en deux étapes distinctes :

— **Convolution par profondeur (Depthwise Convolution)** : Cette étape applique un filtre convolutionnel unique à chaque canal d'entrée indépendamment. Si l'entrée comporte N canaux, on applique N convolutions distinctes.

— **Convolution ponctuelle (Pointwise Convolution)** : Cette étape utilise une convolution 1×1 pour combiner les sorties de la convolution par profondeur. Cette étape permet de mélanger les informations entre les canaux.

Cette décomposition réduit considérablement le nombre de calculs et de paramètres par rapport aux convolutions standard.

Convolution standard :

$$\text{FLOPs} = D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f \tag{3.1}$$

— où D_k est la taille du noyau

— M est le nombre de canaux d'entrée

— N est le nombre de canaux de sortie

— D_f est la taille spatiale de la sortie.

Convolution par profondeur suivie de convolution ponctuelle :

$$\text{FLOPs}_{\text{depthwise}} = D_k \cdot D_k \cdot M \cdot D_f \cdot D_f \tag{3.2}$$

$$\text{FLOPs}_{\text{pointwise}} = M \cdot N \cdot D_f \cdot D_f \tag{3.3}$$

Ce qui donne un total de :

$$\text{FLOPs}_{\text{total}} = D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f \tag{3.4}$$

Bloc de base de MobilenetV1 : Chaque bloc de MobilenetV1 V1 est constitué d'une convolution par profondeur suivie d'une convolution ponctuelle, chacune suivie d'une normalisation par lot (Batch Normalization) et d'une activation ReLU non linéaire. Ces blocs sont empilés pour former l'architecture complète.MobileNetV1 introduit deux hyperparamètres pour ajuster le compromis entre précision et efficacité :

— **Facteur de largeur (Width Multiplier)** α **:** Réduit uniformément le nombre de canaux dans chaque couche. Si $\alpha < 1$, le modèle est plus petit et plus rapide mais peut avoir une précision réduite.

— **Facteur de résolution (Resolution Multiplier)** ρ **:** Réduit la résolution de l'image d'entrée. Une résolution plus faible diminue les calculs mais peut également réduire la précision.

Avantages :

— **Efficacité de calcul et mémoire** : Grâce aux convolutions séparables en profondeur, MobileNet V1 est beaucoup plus léger que les architectures CNN traditionnelles comme VGG ou ResNet.

— **Flexibilité** : Les hyperparamètres permettent d'ajuster l'architecture pour différents besoins en termes de ressources et de précision.

— **Performance** : MobileNet V1 offre un bon équilibre entre efficacité et précision, ce qui le rend adapté pour des applications en temps réel sur des appareils avec des capacités limitées.

3.4 Travaux relatifs à la détection de la violence

Au fil des ans, la détection de la violence est restée au centre des préoccupations des chercheurs en apprentissage automatique, conduisant au développement de nombreux algorithmes visant à détecter les comportements violents dans divers contextes.

À l'aide de méthodes et de modèles différents, les chercheurs ont réalisé des progrès remarquables et obtenu des résultats prometteurs.

— - Sunanda Das et al [59] en 2019, a proposé un système pour détecter la violence à partir de clips vidéo. Tout d'abord, toutes les images sont extraites de chaque clip vidéo, et un processus de sélection des images sélectionne certaines des images de chaque clip vidéo.Le système a appliqué HOG (Histogramme de gradient orienté) comme descripteur de caractéristiques pour extraire les caractéristiques des images. L'histogramme de gradient orienté applique une normalisation du contraste local qui se chevauche afin d'améliorer l'efficacité. Enfin, le système a utilisé différents modèles de classification ainsi qu'une technique de vote majoritaire pour déterminer si un clip vidéo contient de la violence ou non. Le système est capable d'atteindre un taux de précision de 86 en utilisant le classificateur Random Forest.

— -J. Mahmoodi et A. Salajeghe [60] en (2019) ont proposé le descripteur HOMO (Histogram of Optical flow Magnitude and Orientation) pour détecter les comportements violents dans des scènes surpeuplées et non surpeuplées. Leur méthodologie applique différentes valeurs de seuil aux changements de magnitude et d'orientation du flux optique. Les images d'entrée sont d'abord converties en niveaux de gris, puis le flux optique entre les images consécutives est calculé. Six indicateurs binaires, reflétant les changements d'amplitude et d'orientation, sont obtenus et moyennés sur toutes les images. Les histogrammes de ces moyennes sont concaténés pour former le vecteur HOMO. La classification est effectuée avec un classificateur SVM. Les résultats expérimentaux sur l'ensemble de données Hockey Fight montrent que HOMO surpasse les autres descripteurs en performance, tout en étant plus compact et plus rapide à calculer que ViF, bien qu'il soit moins approprié pour les scènes encombrées.

— - Hongchang Li et al en 2019 [61], a développé une nouvelle approche d'apprentissage profond multi-flux pour la compréhension vidéo Violent interaction detection (VID). Le flux RVB spatial basé sur l'attention apprend les régions d'attention par le biais d'un mécanisme d'attention, améliorant ainsi les performances des informations spatiales globales. Le flux spatial local peut apprendre plus efficacement les caractéristiques locales en divisant simplement l'image. Le flux temporel permet d'obtenir une représentation des caractéristiques temporelles de la vidéo en utilisant une méthode de flux optique. En fusionnant les trois flux, ils ont obtenu des performances de pointe sur les ensembles de données de combats de hockey, de films, de VID et de leur EsV. En analysant l'influence des différentes modalités sur les performances de détection, ils ont identifié la combinaison la plus efficace entre les différentes modalités. L'algorithme proposé n'est pas limité à l'application de la VID ; il peut également être appliqué à d'autres applications de reconnaissance d'activité

— - L'étude de Mann B. Patel [65] en 2021 présente un système de détection de

violence pseudo- temps réel pour les vidéos de surveillance, comparant deux approches avec une préférence pour CNN + LSTM. L'analyse des données audio est explorée et les méthodes sont testées sur trois ensembles de données : Hockey Fight, Movies, et Violent-Flows. Malgré des ressources limitées, l'implémentation montre des résultats prometteurs, soulignant l'importance du prétraitement des données et des paramètres d'entraînement. Une application complémentaire pour la détection de la violence en temps réel est développée. Les architectures utilisées combinent CNN et LSTM, avec RESNET50 CNN pour extraire les caractéristiques visuelles et des couches de dropout, de pooling maximum, et des couches denses pour la classification finale.

— - En 2022, Dr. Ajay Shanker Singh et son équipe [64] ont mit en œuvre une étude qui explore l'utilisation de deux architectures de réseaux neuronaux, MobileNetV2 et InceptionV3, pré- entraînées sur des ensembles de données non liés à la violence, pour la détection de la violence dans les vidéos. Les caractéristiques visuelles des vidéos sont extraites à partir d'images clés à l'aide de ces réseaux. Des techniques d'extraction et de classification des caractéristiques sont appliquées pour distinguer les vidéos violentes des non violentes. De plus, des méthodes de regroupement des images clés et des modules d'attention spatiale- temporelle sont introduits pour améliorer la précision de la détection. Le modèle résultant, basé sur MobileNetV2, démontre une précision de validation de 90,26 dans la détection de la violence, montrant ainsi l'efficacité de cette approche dans la reconnaissance des comportements violents dans les vidéos de surveillance.

— - Fernando J. Rendón-Segador et al.[62], en 2023 ont présenté une étude de cas sur la détection d'événements violents dans les vidéos avec un modèle d'apprentissage profond innovant, CrimeNet, qui combine les architectures NSL (Neural Structured Learning) et ViT (Vision Transformer). CrimeNet a surpassé l'état de l'art en détection de violence, avec des gains de 9,4 à 22,17 points dans l'AUC ROC sur quatre ensembles de données. Une étude d'ablation a montré que l'utilisation de NSL au lieu de l'apprentissage supervisé améliore les résultats de 9,55 à 12,75 points dans l'AUC ROC. Le modèle a également montré des performances de pointe sur plusieurs ensembles de données, atteignant des AUC ROC de 73,5% et 70,2% pour les ensembles de données multiclasses, et de 81,35% et 78,9% pour les ensembles de données binaires, améliorant les modèles précédents de 12,39 à 25,22 points. Ces résultats démontrent la robustesse du modèle et son potentiel pour des applications futures telles que la classification vidéo, la détection, la classification d'actions humaines et le suivi d'éléments dans une vidéo.

— -U. Rachna et al.[63] en 2023 ont proposé un pipeline de détection de violence en temps réel utilisant OpenPose, YoloV3 combiné avec Deepsort, DTW, et trois classificateurs (KNN, Random Forests, Naive Bayes) pour distinguer les actions violentes (coups de poing et de pied) des actions non violentes (marche). OpenPose permet une estimation rapide des poses des individus, tandis que YoloV3 et Deepsort assurent une détection et un suivi efficaces des personnes en temps réel. Les angles des articulations sont extraits pour les différentes actions et les scores DTW entre eux sont calculés puis introduits dans le classificateur. L'approche prend en compte la nature temporelle des données tout en maintenant un faible coût de calcul, augmentant ainsi la fiabilité et l'efficacité du système dans la reconnaissance d'actions avec des données minimales, principalement pour des

activités violentes dans des zones bien éclairées avec une visibilité totale du corps.

— Il existe quelques travaux sur la détection de la violence, avec des modifications dans le modèle utilisé. A. Datta, M. Shah et al. (2002) ont proposé un système qui infère la violence à partir de la trajectoire de mouvement des membres tracée par des modèles d'estimation de pose, puis en la faisant passer dans un LSTM pour obtenir l'inférence. Tao Zhang, Zhijie Yang et al. (2016) ont proposé un modèle gaussien de flux optique qui, lorsqu'il est passé au classificateur linéaire, donne des régions où la violence est inférée.[67] [68]

3.5 conclusion

Ce chapitre a exploré diverses méthodes d'analyse des scènes en vidéo-surveillance, allant des techniques traditionnelles de traitement d'image comme la soustraction de l'arrière-plan, le flux optique et les classificateurs de Haar, aux approches modernes utilisant le deep learning. Nous avons abordé des méthodes avancées telles que l'estimation de pose et les pipelines de détection de violence, ainsi que la reconnaissance de scènes humaines, incluant la reconnaissance de scène et d'action.

Nous avons également discuté des techniques d'extraction de caractéristiques, des modèles d'apprentissage automatique, et de l'apprentissage par transfert, en mettant en avant l'efficacité des modèles CNN pré-entraînés pour la détection de violence. Ces approches modernes permettent d'améliorer la précision et de réduire les coûts et le temps de développement par rapport aux méthodes traditionnelles.

En somme, ce chapitre a fourni un aperçu des principales techniques utilisées dans l'analyse des scènes de vidéo-surveillance, mettant en évidence l'évolution vers des solutions plus robustes et efficaces grâce au deep learning.

Chapitre 4

Implementation d'un système de detection de violence

4.1 Introduction

Ce chapitre détaille la mise en œuvre de notre système de détection de violence, le résultat d'une recherche approfondie et d'une expérimentation rigoureuse. Nous avons consacré beaucoup de temps à explorer diverses approches et à tester de nombreux modèles avant d'arriver à notre solution actuelle.

Nous commencerons par la configuration de l'environnement de travail, en expliquant les outils et les logiciels nécessaires. Ensuite, nous décrirons le processus de création de notre dataset, compilé à partir de plusieurs sources en ligne pour garantir sa diversité et sa représentativité.

Notre parcours de recherche nous a conduit à expérimenter avec plusieurs architectures de réseaux neuronaux, et après une évaluation comparative approfondie, nous avons sélectionné la combinaison optimale pour notre application. L'entraînement de notre modèle a été réalisé sur Google Colab, en utilisant le transfert d'apprentissage pour améliorer l'efficacité de l'extraction de caractéristiques.

Cette section détaillera les étapes suivies, depuis la préparation des données jusqu'à l'optimisation des hyperparamètres, et présentera une évaluation complète des performances du modèle en utilisant des métriques clés telles que la précision, le rappel et le F-score.

Enfin, nous discuterons des défis rencontrés et des solutions innovantes mises en œuvre pour les surmonter. Ce chapitre témoigne de notre engagement à explorer, comparer et perfectionner chaque aspect de notre approche pour développer un système de détection de violence efficace et précis.

4.2 Environnement de travail

4.2.1 Le pc portable utilisé

— Processor : Intel(R) Core(TM)i5-10G RAM : 8,00 GB

4.2.2 Langage python

Python est un langage de programmation interprété de haut niveau, réputé pour sa simplicité et sa lisibilité. Sa facilité d'utilisation, ses bibliothèques puissantes, sa communauté active et sa grande flexibilité en font un choix privilégié pour le développement de l'intelligence artificielle. Grâce à Python, les chercheurs et les développeurs peuvent se concentrer sur la résolution de problèmes complexes en IA, bénéficiant d'un large éventail d'outils et de ressources qui facilitent le prototypage rapide et l'implémentation efficace de solutions avancées.

4.2.3 Bibliothèques

4.2.4 Tensorflow

TensorFlow, développée par Google en 2011, est une bibliothèque open-source qui permet de calculer numériquement et d'apprendre automatiquement à grande échelle. Il

regroupe une gamme de modèles et d'algorithmes de machine learning et de deep learning, facilitant leur utilisation grâce à l'application de métaphores programmatiques communes. TensorFlow peut être utilisé pour développer des modèles pour une variété de tâches, y compris le traitement du langage naturel, la reconnaissance d'image, la reconnaissance d'écriture manuscrite et différentes simulations basées sur le calcul, telles que les équations aux dérivées partielles.

4.2.5 keras

Keras est une bibliothèque open source de réseaux de neurones, écrite en Python. Elle est conçue pour être simple et modulable, permettant le prototypage rapide de modèles d'apprentissage profond (deep learning). Keras fonctionne comme une interface haut niveau pour des bibliothèques comme TensorFlow.

4.2.6 Numpy

Le terme NumPy est une abréviation de "Numerical Python." C'est une bibliothèque open-source Python qui est largement utilisée pour les calculs mathématiques et scientifiques. Il présente un large éventail de fonctionnalités et d'outils qui peuvent être profitables pour la programmation scientifique, en particulier dans les domaines de la science des données, de l'ingénierie, des mathématiques et des sciences. NumPy est très utile pour effectuer des calculs logiques et mathématiques sur des tableaux et des matrices. Cet outil est capable d'effectuer ces opérations à un rythme nettement plus rapide que les listes Python, tout en nécessitant moins de mémoire et d'espace de stockage.

4.2.7 opencv

OpenCV, abréviation de "Open Source Computer Vision Library", est une bibliothèque open-source spécialisée dans le traitement d'images et la vision par ordinateur. Elle offre une large gamme de fonctionnalités pour manipuler des images et des vidéos, notamment la lecture, l'écriture, le traitement et l'analyse d'images.

4.2.8 Tkinter

Tkinter est la bibliothèque standard de Python dédiée à la création d'interfaces graphiques (GUI). Elle offre des outils simples et efficaces pour développer des applications interactives grâce à une intégration fluide avec Python. Tkinter permet de concevoir des interfaces utilisateur attrayantes en utilisant une variété de widgets comme des boutons, des labels et des boîtes de dialogue.

4.3 Environnement de développement intégré (IDE)

4.3.1 Vscode

Visual Studio Code (communément appelé VS Code) est un éditeur de texte libre et gratuit développé par Microsoft et disponible pour Windows, Linux et macOS. Malgré sa nature relativement légère, l'éditeur intègre une suite de fonctionnalités sophistiquées qui ont contribué à son ascension comme l'un des outils d'environnement de développement

les plus populaires de ces derniers temps. VS Code prend en charge une gamme complète de langages de programmation, notamment Java, C++ et Python, ainsi que CSS, Go et Dockerfile. En outre, VS Code permet aux utilisateurs d'installer et de créer de nouvelles extensions, notamment des linters de code, des débogueurs et des supports de développement cloud et web.

4.4 Environnement de Développement en Ligne

4.4.1 Kaggle

Kaggle est une plateforme en ligne pour la compétition en science des données. Elle permet aux utilisateurs de trouver et de publier des ensembles de données, d'explorer et de construire des modèles en collaboration avec d'autres data scientists, et de participer à des compétitions pour résoudre des défis complexes en science des données.

4.4.2 colab

Google Colab (ou Colaboratory) est un environnement de développement intégré (IDE) en ligne gratuit qui permet d'écrire et d'exécuter du code Python directement dans le navigateur. Colab offre un accès gratuit aux GPU, ce qui le rend particulièrement utile pour les tâches d'apprentissage profond et d'analyse de données.

4.5 Implémentation du Système de Détection de Violence

Cette section détaille le processus de mise en œuvre d'un système de détection de violence dans les vidéos. Nous aborderons les différentes étapes nécessaires, depuis la creation de notre dataset jusqu'à l'entraînement et l'évaluation du modèle. En décrivant les outils, les modèles utilisés, et les défis rencontrés, nous présenterons comment nous avons développé une solution efficace pour identifier les actes de violence dans des séquences vidéo.

4.5.1 Création de notre Dataset (jeux de données)

La création de notre dataset a été une étape cruciale et complexe de notre projet. Initialement, nous avons exploré plusieurs datasets disponibles en ligne pour entraîner notre modèle de détection de violence. Chaque dataset a été utilisé individuellement pour évaluer sa pertinence et son efficacité. Cependant, nous avons rapidement rencontré plusieurs problèmes spécifiques à l'utilisation de chaque dataset seul :

— **Imbalance des Classes** : Certains datasets présentaient un déséquilibre marqué entre les classes de violence et de non-violence, ce qui a conduit à un modèle biaisé et une mauvaise performance en conditions réelles.

— **Manque de Diversité** : D'autres datasets manquaient de diversité dans les scènes de violence, ce qui limitait la capacité du modèle à généraliser et à détecter des actes de violence dans différents contextes et environnements.

— **Qualité et Résolution des Vidéos** : La qualité et la résolution des vidéos variaient considérablement d'un dataset à l'autre, posant des défis lors de l'entraînement du modèle, qui nécessite une cohérence dans les données d'entrée pour des performances optimales.

— **Annotations Incohérentes** : Les annotations des actes de violence n'étaient pas toujours cohérentes ou précises, compliquant l'entraînement supervisé du modèle.

Face à ces défis, nous avons opté pour une approche combinée en fusionnant plusieurs datasets afin de créer un ensemble de données plus robuste et représentatif. Cette approche a permis de surmonter les limitations individuelles de chaque dataset :

— **Équilibrage des Classes** : En combinant plusieurs sources, nous avons pu équilibrer les classes de violence et de non-violence, réduisant ainsi le biais dans l'entraînement du modèle.

— **Augmentation de la Diversité** : La fusion de datasets provenant de différents contextes a enrichi la diversité des scènes de violence, améliorant la capacité de généralisation du modèle.

— **Standardisation de la Qualité** : Nous avons appliqué des techniques de prétraitement pour normaliser la qualité et la résolution des vidéos, assurant une cohérence des données d'entrée.

— **Amélioration des Annotations** : En consolidant les annotations de multiples sources, nous avons pu affiner et corriger les incohérences, garantissant des labels plus précis pour l'entraînement.

Cette stratégie de création de dataset nous a permis de constituer une base de données solide et fiable, essentielle pour entraîner un modèle performant de détection de violence.

4.5.2 Les jeux de donnees utilisées

— Real life violence situations dataset, by Mohamed Elesawy and 2 collaborators Updated 5 years ago. Size : 4GB

— Violence detection dataset. by Mehdi Hasan Nirob. Updated 4m ago. Size : 3GB

— Pua-dataset. by Habiba Ahmed Elshabasy. Updated 3 months ago. Size : 3GB

— Hockey fight videos. by Yasser Shrief · Updated 3 years ago. Size : 172MB

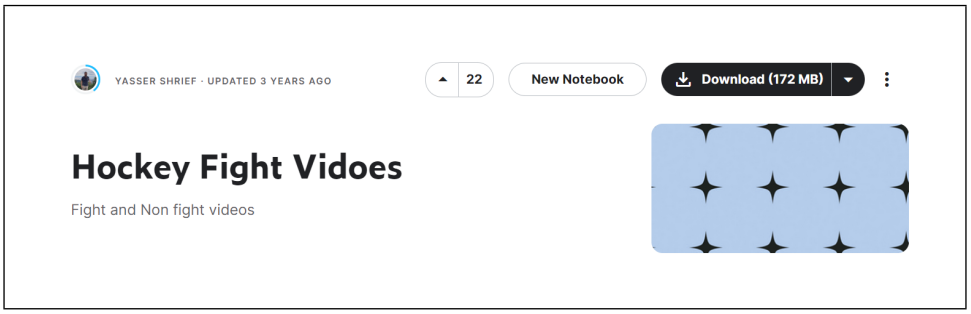

FIGURE 4.1 – Hockey fight videos. [76]

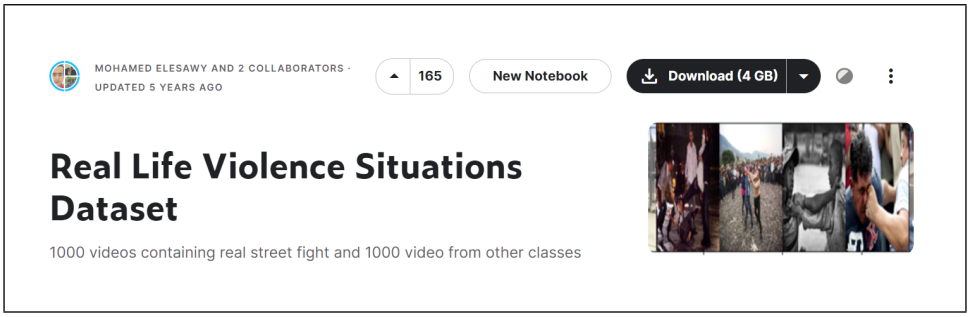

FIGURE 4.2 – Real Life Violence Situations Dataset. [77]

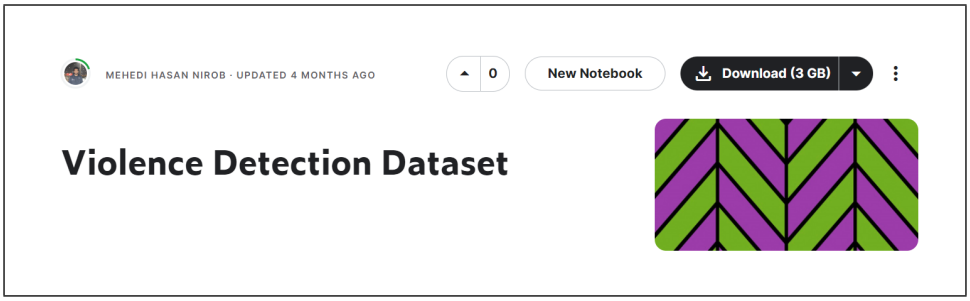

FIGURE 4.3 – Violence Detection Dataset. [78]

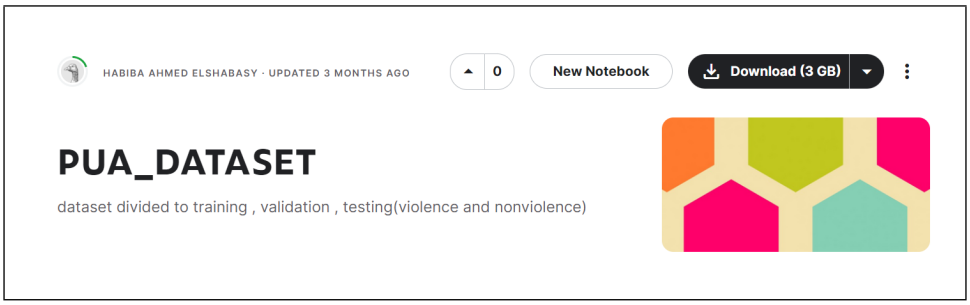

FIGURE 4.4 – PUA dataset. [79]

4.5.3 Prétraitement des Données

Le prétraitement des vidéos était une étape fondamentale pour assurer la qualité et la cohérence des données d'entrée dans notre modèle. Voici les différentes étapes que nous avons suivies :

4.5.4 Extraction des Frames

:

— Chaque vidéo est décomposée en frames (images) à un taux de capture spécifique=20 frames , pour assurer une représentation uniforme de la séquence vidéo.

— Utilisation de bibliothèques comme OpenCV pour extraire les frames, permettant de gérer efficacement des vidéos de différentes durées et résolutions.

4.5.5 Redimensionnement des Images

— Les frames extraites sont redimensionnées à une taille uniforme pour assurer la compatibilité avec notre modèle (224x224 pixels pour MobilenetV1).

4.5.6 Normalisation des Pixels

Normalisation des valeurs des pixels des images pour les ramener dans la plage [0, 1]. Lorsque l'on normalise les valeurs des pixels d'une image, on ajuste leurs intensités lumineuses pour qu'elles soient comprises dans une plage spécifique, [0, 1]. Voici pourquoi cette étape est importante :

— **Amélioration de la Convergence du Modèle** : En normalisant les valeurs des pixels, on s'assure que toutes les caractéristiques (features) ont une échelle comparable. Cela aide le modèle à converger plus rapidement pendant l'entraînement, car les gradients sont plus cohérents et prévisibles.

— **Réduction des Biais** : La normalisation des pixels réduit les biais introduits par des différences d'échelle entre les caractéristiques. Si les valeurs des pixels varient sur une grande échelle, cela peut rendre l'optimisation plus difficile et introduire des instabilités dans le processus d'apprentissage.

— **Préparation des Données pour le modele** : De nombreux modèles de deep learning, notamment ceux basés sur des réseaux de neurones, sont sensibles à l'échelle des données. La normalisation assure une préparation adéquate des données pour maximiser la performance du modèle.

4.5.7 Augmentation des Données

Application de techniques d'augmentation des données telles que la rotation, le recadrage, le renversement horizontal Ces techniques augmentent la diversité des données et aident à prévenir le surapprentissage en simulant une variété de conditions d'éclairage et d'angles de prise de vue.

4.5.8 Encodage des Labels

Les annotations des actes de violence sont encodées de manière cohérente pour être utilisées dans l'entraînement supervisé. Utilisation du codage one-hot pour représenter les catégories de violence et de non-violence

Codage One-Hot Le codage one-hot est une méthode de représentation des variables catégorielles (dans ce cas, les catégories de violence et de non-violence) sous forme de vecteurs binaires. Chaque catégorie est représentée par un vecteur où toutes les valeurs sont à zéro, sauf une, qui est à un, correspondant à la catégorie spécifique :

— **Catégorie de violence (V)** : [1, 0]

— **Catégorie de non-violence (NV)** : [0, 1]

Pourquoi nous avons utiliser le Codage One-Hot ?

Facilité de Traitement : Ce format de représentation est facile à interpréter et à manipuler par les algorithmes d'apprentissage automatique. Chaque classe est distincte-

ment représentée, ce qui facilite le calcul des pertes (losses), des métriques d'évaluation et la rétropropagation des gradients lors de l'entraînement du modèle.

Éviter les Biais : En utilisant le codage one-hot, on évite toute implication d'ordre ou de distance entre les catégories. Chaque classe est traitée de manière égale et indépendante des autres, ce qui réduit les biais potentiels dans l'apprentissage du modèle

4.5.9 Organisation des Données

Les frames extraites et prétraitées sont organisées dans des dossiers structurés, facilitant le chargement efficace des données pendant l'entraînement du modèle.

Ces étapes de prétraitement assurent une qualité optimale des données d'entrée, préparant ainsi le terrain pour un apprentissage efficace et performant du modèle de détection de violence sur Google Colab.

4.6 Entrainement sur colab

4.6.1 Choix de l'Approche

Pour la détection de la violence, nous avons effectué de nombreuses recherches et tests avec plusieurs modèles. Initialement, nous avons exploré diverses architectures de CNN et de modèles récurrents, mais nous avons rencontré des limitations soit en termes de précision soit en termes de capacité à gérer les séquences temporelles complexes des vidéos.

4.6.2 Comparaison des Modèles Pré-entraînés + LSTM

Les modèles examinés comprenaient VGG16, VGG19, NASNet, et MobilenetV1. Voici un résumé des résultats obtenus lors de l'entraînement initial

Pre-trained model	**Loss**	**Accuracy**
VGG16	0.13	0.82
VGG19	0.11	0.84
NASNet	0.07	0.91
MobilenetV1	0.04	0.94

TABLE 4.1 – Comparaison des modèles pré-entraînés sur les critères de perte et précision.

Après une évaluation approfondie de quatre modèles pré-entraînés - VGG16, VGG19, NASNet et MobileNetV1 - nous avons choisi d'utiliser MobileNetV1 avec LSTM. MobileNetV1 s'est distingué par sa précision et ses performances supérieures pour l'extraction de caractéristiques, rendant ce modèle particulièrement adapté à nos besoins. Couplé avec LSTM, qui excelle dans la capture des relations temporelles à long terme, cette approche a offert un équilibre optimal entre performance et précision.

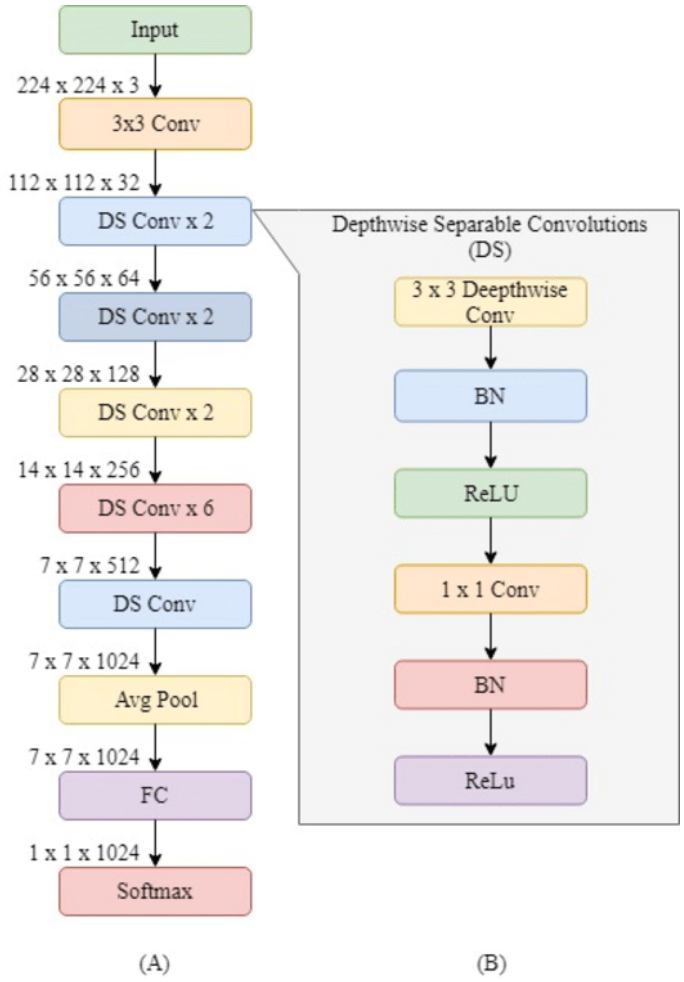

FIGURE 4.5 – Architecture MobilenetV1. [82]

```
[ ]  image_model = MobileNet( weights='imagenet',include_top=True,input_shape=(224,224,3))
```

FIGURE 4.6 – MobilenetV1

```
=================================================================
Total params: 4253864 (16.23 MB)
Trainable params: 4231976 (16.14 MB)
Non-trainable params: 21888 (85.50 KB)
_________________________________________________________________
```

FIGURE 4.7 – Résumé de MobilenetV1

Pour l'extraction des caractéristiques visuelles à partir des vidéos, nous avons opté pour l'utilisation du modèle MobilenetV1 pré-entraîné sur l'ensemble de données ImageNet. Ce choix a été motivé par sa capacité éprouvée à capturer des caractéristiques discriminantes dans des images complexes tout en étant computationally efficient. En initialisant le modèle avec weights='imagenet', nous avons bénéficié des connaissances préalablement acquises par MobilenetV1 sur une vaste gamme de catégories visuelles. La configuration include-top=True nous a permis d'intégrer la couche de classification finale du modèle, facilitant ainsi l'extraction de caractéristiques pertinentes pour notre tâche de détection de la violence dans les vidéos. En spécifiant input-shape=(224,224,3), nous avons adapté le modèle aux dimensions standardisées des images d'entrée, assurant une compatibilité directe avec notre flux de traitement des données.

Cette approche nous a permis de tirer parti des avantages du transfert d'apprentissage, en utilisant les représentations visuelles apprises par MobilenetV1

4.6.3 L'architecture LSTM

Initialisation du Modèle Séquentiel Nous avons initialisé un modèle séquentiel, ce qui signifie que les couches du réseau neuronal sont empilées de manière séquentielle l'une après l'autre.

Couche LSTM Nous avons ajouté une couche LSTM (Long Short-Term Memory). Cette couche est particulièrement adaptée au traitement de données séquentielles comme les séquences temporelles dans les vidéos. Elle est capable de capturer et de mémoriser les dépendances à long terme dans les séquences.

Couches Denses Après la couche LSTM, nous avons ajouté plusieurs couches denses entièrement connectées :

— Une première couche dense avec une fonction d'activation ReLU, qui introduit de la non-linéarité dans le modèle. Cela permet au réseau de modéliser des relations complexes entre les caractéristiques extraites.

— Une deuxième couche dense avec une fonction d'activation sigmoïde. Cette fonction est utilisée pour la classification binaire car elle produit des sorties dans l'intervalle [0, 1], ce qui est interprété comme une probabilité pour chaque classe.

— Couche de Sortie :

La dernière couche dense de notre modèle est une couche de sortie avec deux unités. Cela correspond à votre tâche de classification binaire (détection de violence ou non-violence dans les vidéos). La fonction d'activation softmax est utilisée ici pour convertir les sorties brutes en probabilités, fournissant ainsi une estimation de la probabilité pour chaque classe

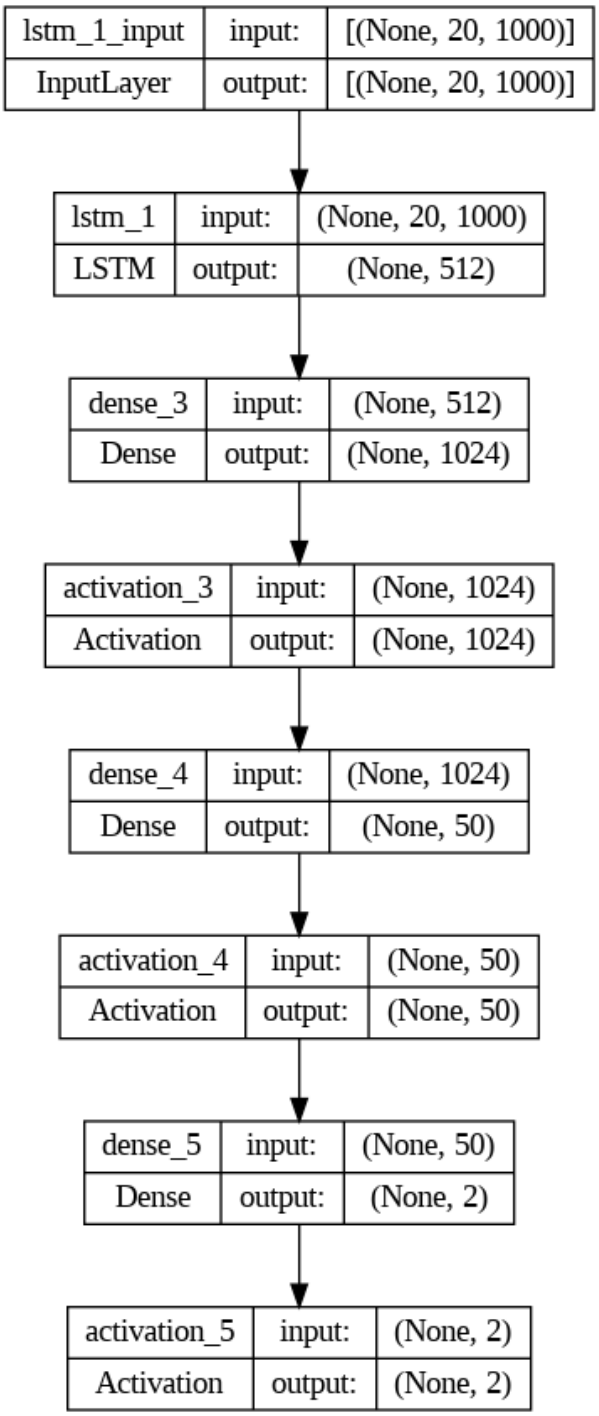

FIGURE 4.8 – Résumé de LSTM

```
# Initialize the Sequential model
model = Sequential()

# Add an LSTM layer with specified number of units and input shape
model.add(LSTM(units=rnn_size, input_shape=(n_chunks, chunk_size)))

# Add a dense layer with 1024 units and ReLU activation function
model.add(Dense(units=1024))
model.add(Activation('relu'))

# Add another dense layer with 50 units and sigmoid activation function
model.add(Dense(units=50))
model.add(Activation('sigmoid'))

# Add the output layer with 2 units (for binary classification) and softmax activation function
model.add(Dense(units=2))
model.add(Activation('softmax'))
```

FIGURE 4.9 – LSTM architecture

4.6.4 Assemblage du Modèle

L'intégration de MobilenetV1 et LSTM s'est effectuée par l'apprentissage par transfert, une méthode où les couches de MobilenetV1 ont été gelées pour conserver les poids pré-appris sur ImageNet. Seules les couches supérieures du réseau LSTM ont été entraînées spécifiquement sur notre jeu de données dédié à la détection de violence.

Cette approche combinée a permis de capitaliser sur les capacités d'extraction de caractéristiques robustes de MobilenetV1 et sur la capacité de LSTM à modéliser les séquences temporelles complexes. En utilisant l'apprentissage par transfert, nous avons pu minimiser les besoins en calculs et en données d'entraînement tout en optimisant l'efficacité du modèle.

Les poids transférés sont stockés de manière efficace au format HDF5, facilitant ainsi la manipulation et l'utilisation lors des phases d'entraînement subséquentes. Le jeu de données a été stratifié en ensembles d'entraînement (80 %) et de validation (20%), permettant au modèle d'apprendre à partir d'une quantité significative de données tout en évaluant sa capacité à généraliser sur des exemples non vus.

4.7 Entraînement du Modèle

— Nous avons décidé d'entraîner notre modèle sur 200 époques. Ce nombre d'époques a été déterminé après plusieurs expérimentations préliminaires visant à trouver un équilibre optimal entre la convergence du modèle et le temps d'entraînement nécessaire. En fixant ce nombre à 200, nous visons à permettre au modèle de parcourir suffisamment de fois l'ensemble des données d'entraînement pour apprendre les caractéristiques pertinentes tout en évitant un entraînement excessif qui pourrait conduire à du surapprentissage.

— La taille du lot (batch-size) a été fixée à 500. Ce choix est motivé par la capacité de la mémoire GPU disponible sur Google Colab, permettant ainsi de traiter effi-

cacement un nombre significatif d'échantillons en parallèle. Une taille de lot plus grande accélère le processus d'entraînement en exploitant pleinement les capacités de calcul disponibles, tout en optimisant l'utilisation des ressources matérielles.

```
# Train the model
epoch = 200  # Number of epochs for training
batchS = 500  # Batch size for training

# Calculate the number of training videos (80% of the dataset)
train_split = int(0.8 * len(data))
```

FIGURE 4.10 – Parametres d'entrainement

Pour évaluer la performance et éviter le surapprentissage, nous avons utilisé une validation croisée en allouant 20 % de nos données à des fins de validation (validation-data). Cette approche nous permet de surveiller régulièrement la capacité du modèle à généraliser sur des données non vues, en ajustant les paramètres du modèle si nécessaire pour améliorer sa capacité de prédiction sur de nouvelles données

4.7.1 Évaluation des Résultats

4.7.2 Métriques d'Évaluation

Pour évaluer les performances de notre modèle de détection de violence, nous avons principalement utilisé les métriques suivantes : la precision (accuracy), la perte (loss) et la matrice de confusion.

Après avoir testé les performances du modèle sur 20 % de l'ensemble des vidéos, le modèle a obtenu une perte de 0,0444 et une précision de 94,24 % sur l'ensemble de données de test.

```
loss 0.04444553330540657
accuracy 0.9424280524253845
```

FIGURE 4.11 – Resultats d'entrainement

Voici le graphe de précision, qui fournit une visualisation complète des performances du modèle en matière d'entraînement et de validation

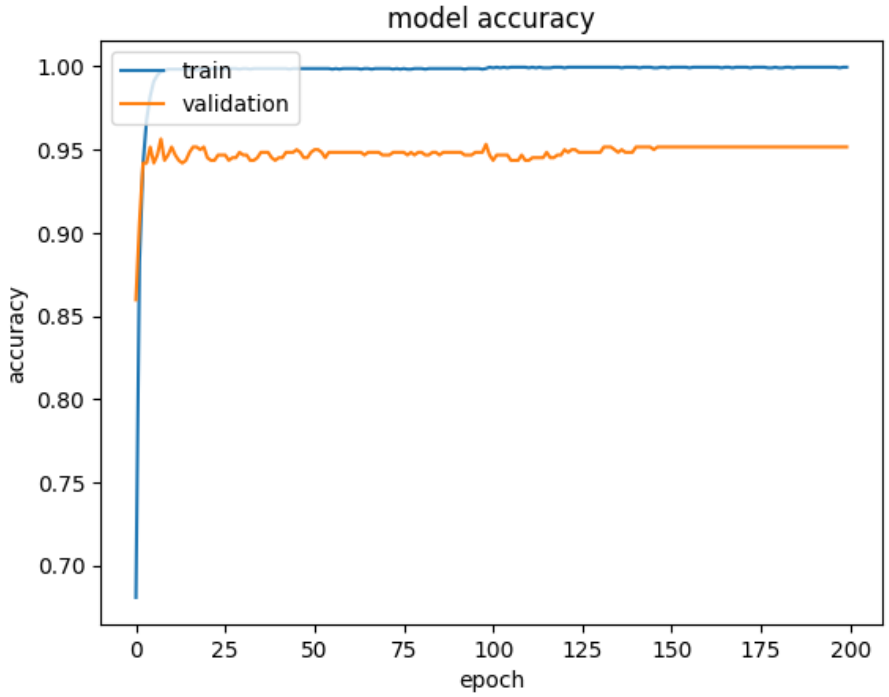

FIGURE 4.12 – La precision

De même, le graphe de perte montre la progression de la fonction de perte du modèle au cours de la formation et du test :

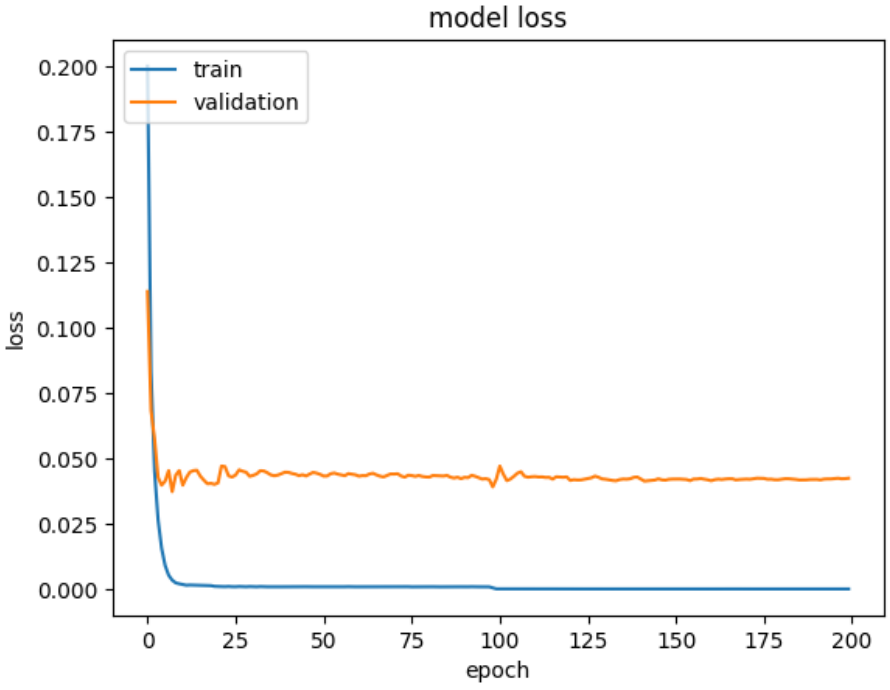

FIGURE 4.13 – Fonction de perte

L'ensemble de données comprend 4 000 vidéos, avec un nombre égal de catégories violentes et non violentes (2 000 chacune).

```
Number of files in directory 1 (/content/drive/MyDrive/Violence Detection Dataset/Nonviolence): 2000
Number of files in directory 2 (/content/drive/MyDrive/Violence Detection Dataset/violence): 2000
```

FIGURE 4.14 – Nombres de videos dans le dataset

4.7.3 Matrice de confusion : Précision

La précision est une métrique d'évaluation utilisée pour mesurer la qualité d'un modèle de classification. Elle est définie comme le rapport entre le nombre de vrais positifs (TP) et le nombre total de prédictions positives (la somme des vrais positifs et des faux positifs (FP)).

La formule de la précision est la suivante :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{4.1}$$

— TP (True Positives) : Le nombre de vrais positifs, c'est-à-dire le nombre d'exemples correctement classés comme positifs.
— FP (False Positives) : Le nombre de faux positifs, c'est-à-dire le nombre d'exemples incorrectement classés comme positifs.

Compte tenu de cet ensemble de données équilibré et de l'analyse de la matrice de confusion, le modèle s'avère très performant dans l'identification des actions violentes : Vrais positifs : Le modèle a correctement identifié 390 cas d'actions Non violentes. Le modèle a classé à tort 20 cas comme des actions violentes alors qu'ils ne l'étaient pas. Vrais négatifs : Le modèle a identifié avec précision 363 cas où il y avait des actions violentes.

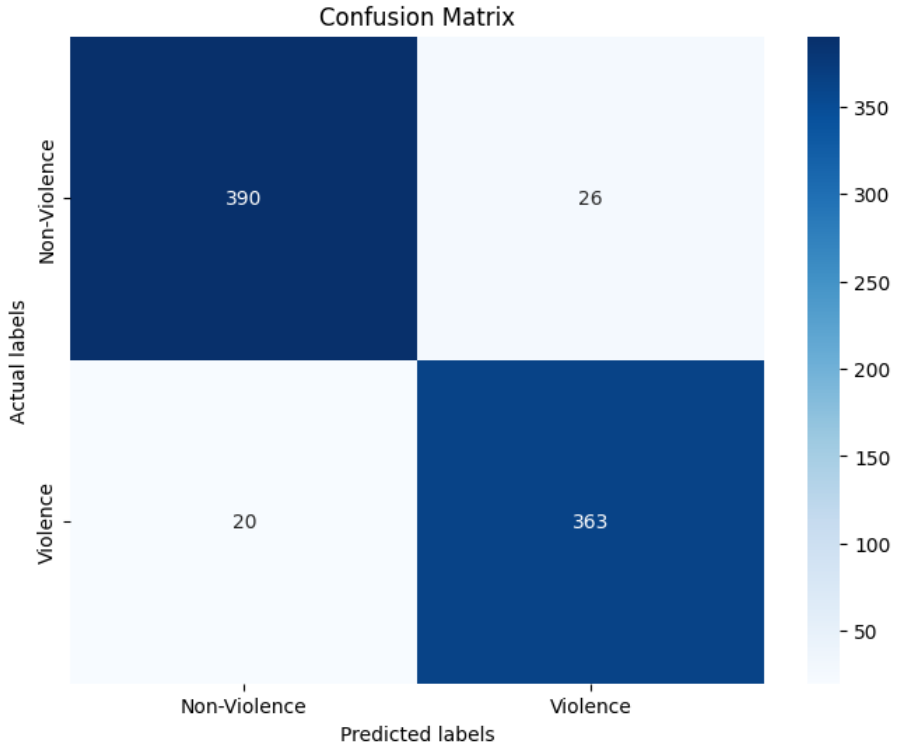

FIGURE 4.15 – Matrice de confusion

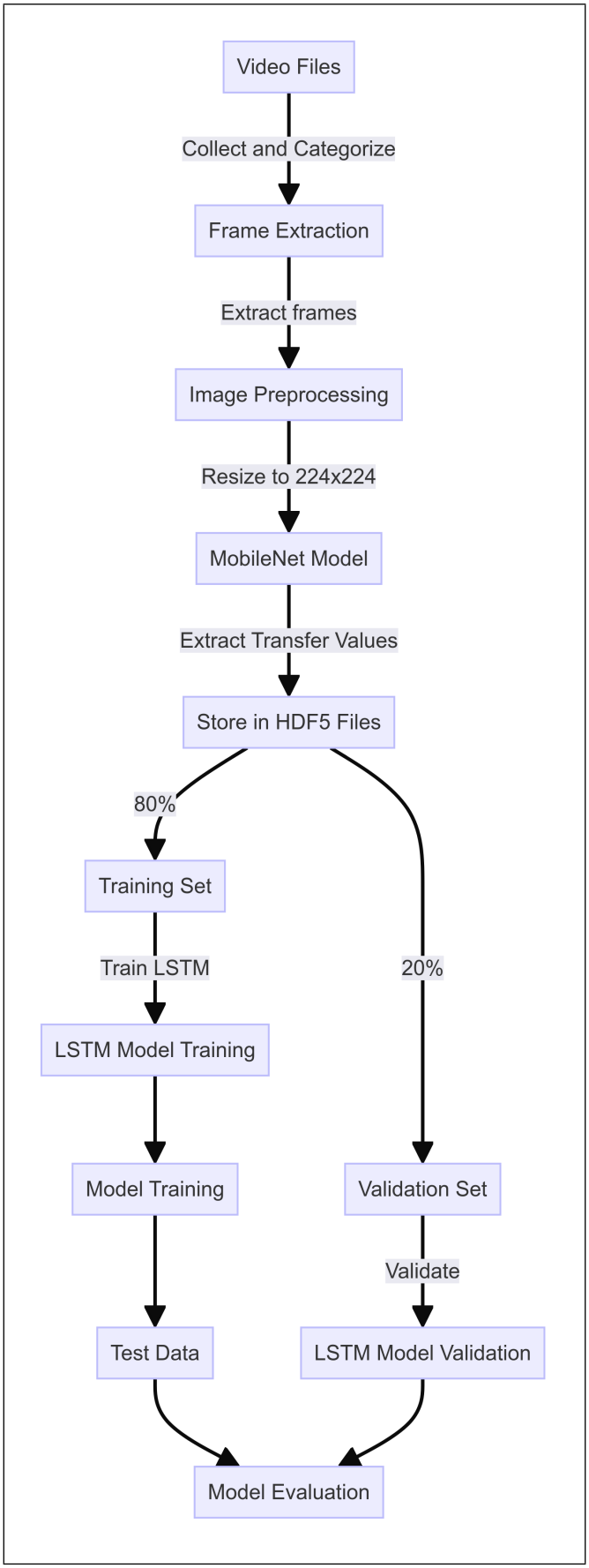

FIGURE 4.16 – Organigramme de l'entrainement et de validation

4.7.4 Implémentation en Temps Réel et Lecture de Vidéos dans VS Code

Nous avons également mis en œuvre un système de détection de la violence en temps réel et à partir de vidéos enregistrées en utilisant VS Code avec une interface utilisateur.

4.7.5 Detection par la Lecture de Vidéos

— **Importation des bibliothèques nécessaires**

Les bibliothèques TensorFlow, OpenCV, Numpy, Deque et PIL sont importées pour gérer les modèles de réseaux de neurones, la capture et le traitement des vidéos, la manipulation des tableaux de données et le dessin sur les images.

— **Chargement du modèle LSTM**

Le modèle LSTM pré-entraîné est chargé à partir d'un fichier HDF5. Ce modèle sera utilisé pour classer les caractéristiques extraites des vidéos afin de détecter des comportements violents.

— **Initialisation de MobilenetV1**

Le modèle MobilenetV1 pré-entraîné est initialisé avec des poids appris sur ImageNet. Ce modèle est utilisé pour extraire des caractéristiques des frames vidéo. La couche de sortie avant la couche de classification finale est utilisée pour obtenir des valeurs de transfert.

— **Prétraitement des frames vidéo**

Une fonction est définie pour prétraiter chaque frame vidéo en convertissant les images de BGR à RGB, en les redimensionnant à une taille spécifique et en normalisant les valeurs des pixels.

— **Capture de la vidéo et initialisation des variables**

Le fichier vidéo est ouvert pour la lecture. Si la vidéo ne peut pas être ouverte, le programme affiche une erreur et se termine. Plusieurs variables sont initialisées pour gérer le traitement des frames, les prédictions et les paramètres de la vidéo.

— **Préparation de l'écriture de la vidéo de sortie**

Un objet de vidéo writer est initialisé pour enregistrer la vidéo de sortie en format MP4 avec le même FPS et la même résolution que la vidéo d'entrée.

— **Boucle principale pour traiter les frames vidéo**

Une boucle while est utilisée pour lire les frames de la vidéo une par une jusqu'à la fin du fichier vidéo. Chaque frame est prétraitée et ajoutée à un buffer.

— **Prédiction de violence**

Lorsque suffisamment de frames sont collectées (20 frames dans ce cas), elles sont passées à travers MobilenetV1 pour obtenir des valeurs de transfert. Ces valeurs sont ensuite reshaped pour correspondre à l'entrée du modèle LSTM. Le modèle LSTM effectue une prédiction de la probabilité de violence pour ces frames.

— **Gestion des résultats de la prédiction**

Les résultats des prédictions sont stockés dans une queue et la moyenne des prédictions est calculée pour obtenir une prédiction plus stable. Si la probabilité de violence est supérieure à un certain seuil, une alerte de violence est affichée sur la vidéo. Sinon, le texte "Normal" est affiché.

— **Affichage et enregistrement des frames traitées**

Chaque frame traitée est convertie au format PIL pour le dessin, puis reconvertie au format OpenCV avant d'être affichée à l'écran et enregistrée dans la vidéo de sortie.

— **Gestion de la fin de la vidéo et de la sortie**

Lorsque la fin de la vidéo est atteinte ou que l'utilisateur appuie sur la touche "ESC", la boucle se termine, et les objets de capture et d'écriture vidéo sont libérés, et toutes les fenêtres OpenCV sont fermées.

4.7.6 Detection en utilisant une webcam

— **Importation des bibliothèques nécessaires**

Les bibliothèques TensorFlow, OpenCV, Numpy, Deque et PIL sont importées pour gérer les modèles de réseaux de neurones, la capture et le traitement des vidéos, la manipulation des tableaux de données et le dessin sur les images.

— **Chargement du modèle LSTM**

Le modèle LSTM pré-entraîné est chargé à partir d'un fichier HDF5. Ce modèle sera utilisé pour classer les caractéristiques extraites des vidéos afin de détecter des comportements violents.

— **Initialisation de MobilenetV1**

Le modèle MobilenetV1 pré-entraîné est initialisé avec des poids appris sur ImageNet. Ce modèle est utilisé pour extraire des caractéristiques des frames vidéo. La couche de sortie avant la couche de classification finale est utilisée pour obtenir des valeurs de transfert.

— **Prétraitement des frames vidéo**

Une fonction est définie pour prétraiter chaque frame vidéo en convertissant les images de BGR à RGB, en les redimensionnant à une taille spécifique et en normalisant les valeurs des pixels.

— **Capture de la vidéo et initialisation des variables**

Le fichier vidéo est ouvert pour la lecture. Si la vidéo ne peut pas être ouverte, le programme affiche une erreur et se termine. Plusieurs variables sont initialisées pour gérer le traitement des frames, les prédictions et les paramètres de la vidéo.

— **Préparation de l'écriture de la vidéo de sortie**

Un objet de vidéo writer est initialisé pour enregistrer la vidéo de sortie en format MP4 avec le même FPS et la même résolution que la vidéo d'entrée.

— **Boucle principale pour traiter les frames vidéo**

Une boucle while est utilisée pour lire les frames de la vidéo une par une jusqu'à la fin du fichier vidéo. Chaque frame est prétraitée et ajoutée à un buffer.

— **Prédiction de violence**

Lorsque suffisamment de frames sont collectées (20 frames dans ce cas), elles sont passées à travers MobilenetV1 pour obtenir des valeurs de transfert. Ces valeurs sont ensuite reshaped pour correspondre à l'entrée du modèle LSTM. Le modèle LSTM effectue une prédiction de la probabilité de violence pour ces frames.

— **Gestion des résultats de la prédiction**

Les résultats des prédictions sont stockés dans une queue et la moyenne des prédictions est calculée pour obtenir une prédiction plus stable. Si la probabilité de violence est supérieure à un certain seuil, une alerte de violence est affichée sur la vidéo. Sinon, le texte "Normal" est affiché.

— **Affichage et enregistrement des frames traitées**

Chaque frame traitée est convertie au format PIL pour le dessin, puis reconvertie au format OpenCV avant d'être affichée à l'écran et enregistrée dans la vidéo de sortie.

— **Gestion de la fin de la vidéo et de la sortie**

Lorsque la fin de la vidéo est atteinte ou que l'utilisateur appuie sur la touche "ESC", la boucle se termine, et les objets de capture et d'écriture vidéo sont libérés, et toutes les fenêtres OpenCV sont fermées.

4.7.7 Interface Utilisateur

Nous avons également développé une interface utilisateur simple pour faciliter l'interaction avec notre application. Cette interface permet de :

Sélectionner la source vidéo : L'utilisateur peut choisir entre l'entrée de la caméra en direct ou une vidéo enregistrée.

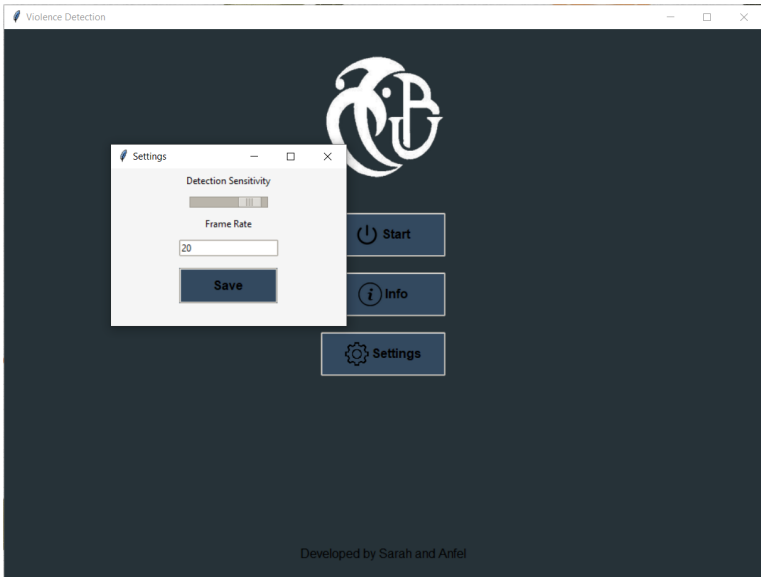

FIGURE 4.17 – Interface

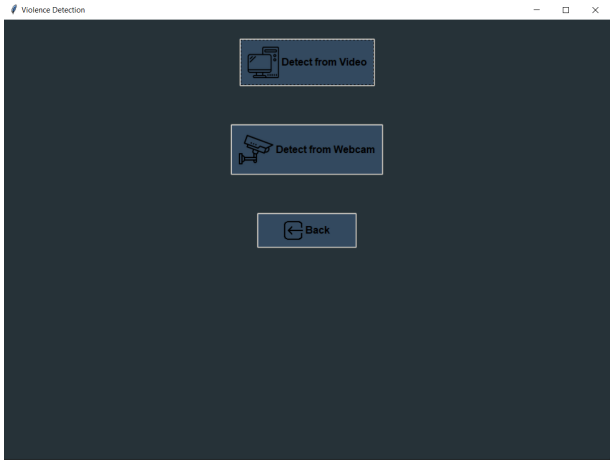

FIGURE 4.18 – Interface.2

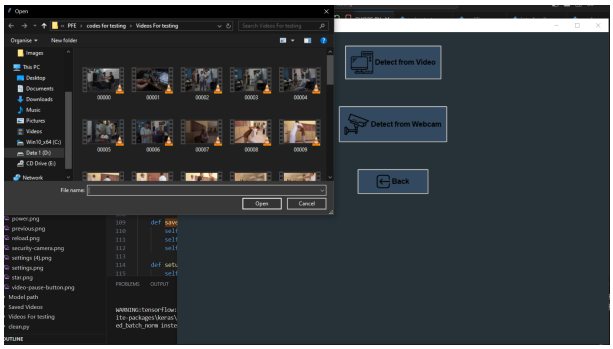

FIGURE 4.19 – Interface.3

Afficher les résultats : Une fenêtre d'affichage en temps réel montre la vidéo avec un affichage d'alerte de détection de violence et le temps de detection

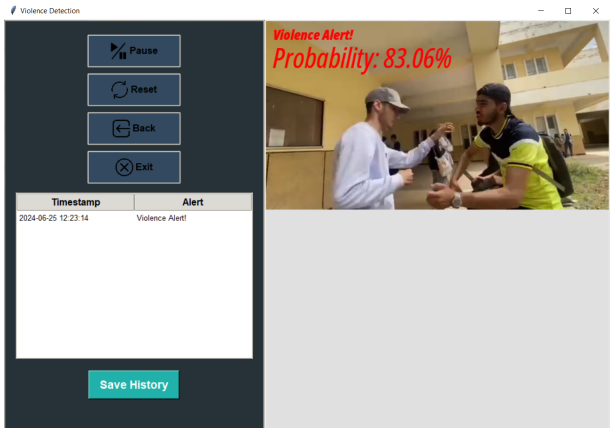

FIGURE 4.20 – Interface.4

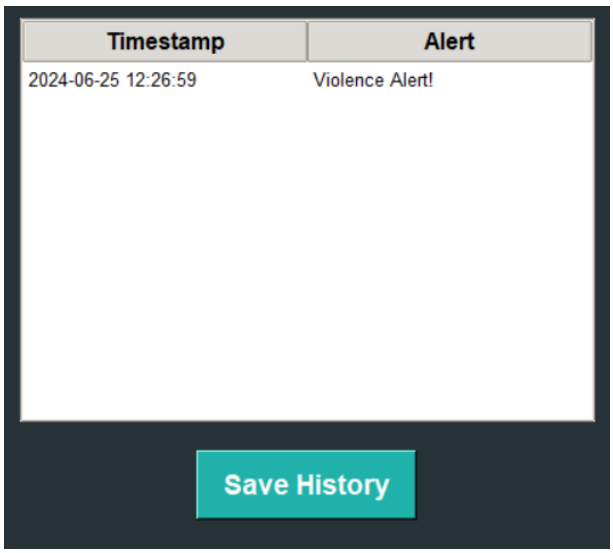

FIGURE 4.21 – Interface.5

Démarrer et arrêter l'analyse : Boutons pour démarrer et arrêter la capture et l'analyse des vidéos.

FIGURE 4.22 – Interface.6

4.7.8 Détection Frame par Frame avec des Vidéos Capturées avec camera Sony A7

Pour évaluer l'efficacité de notre modèle de détection de violence, nous avons utilisé une dataset composée de 30 vidéos capturées avec une caméra Sony A7. Chaque vidéo présente une variété d'angles de vue, de conditions d'éclairage et de postures, ce qui permet d'explorer la capacité du modèle à détecter les comportements violents dans des scénarios diversifiés. La détection frame par frame a été utilisée pour analyser chaque image individuellement,

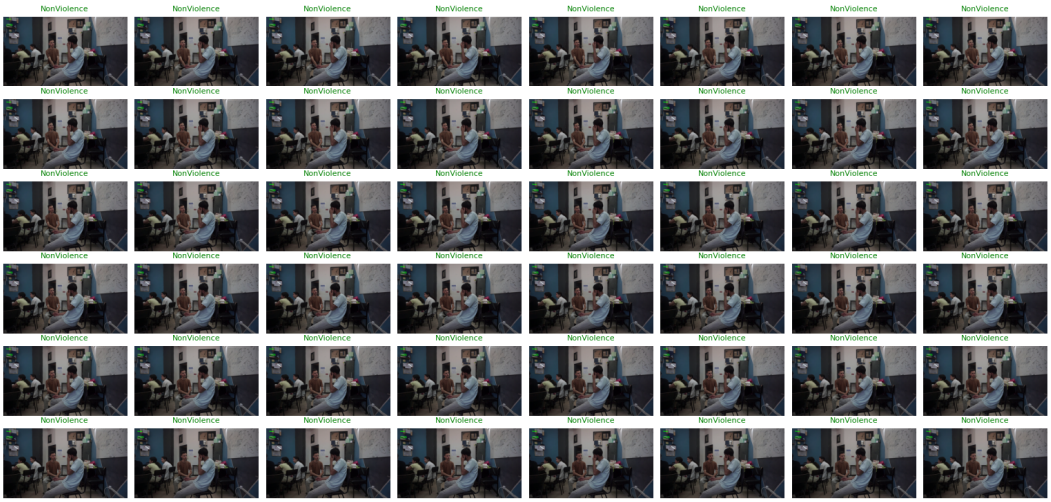

FIGURE 4.23 – Video de non-violence

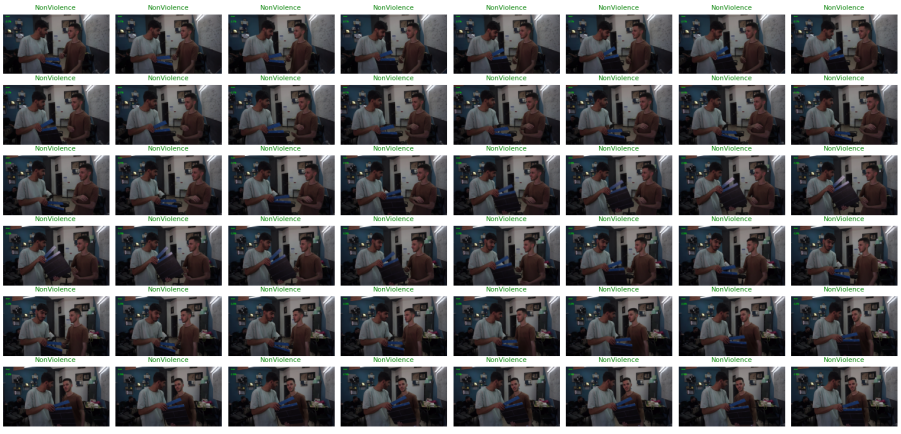

FIGURE 4.24 – Video de non-violence-2

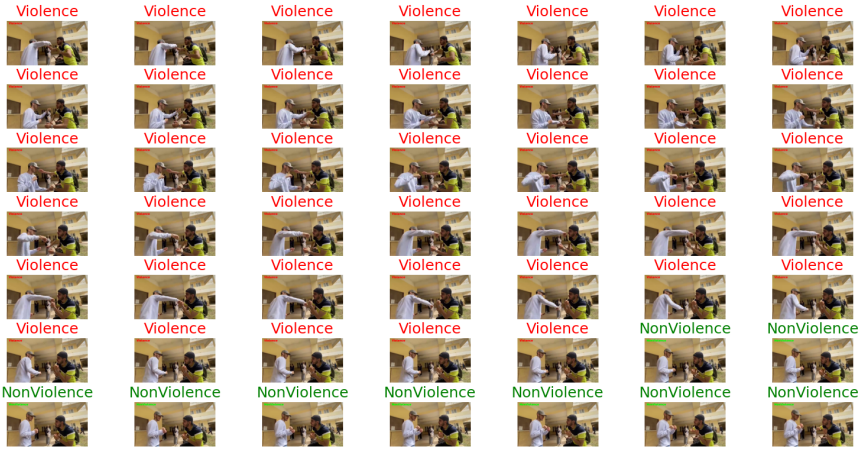

FIGURE 4.25 – Video avec scenes de violence

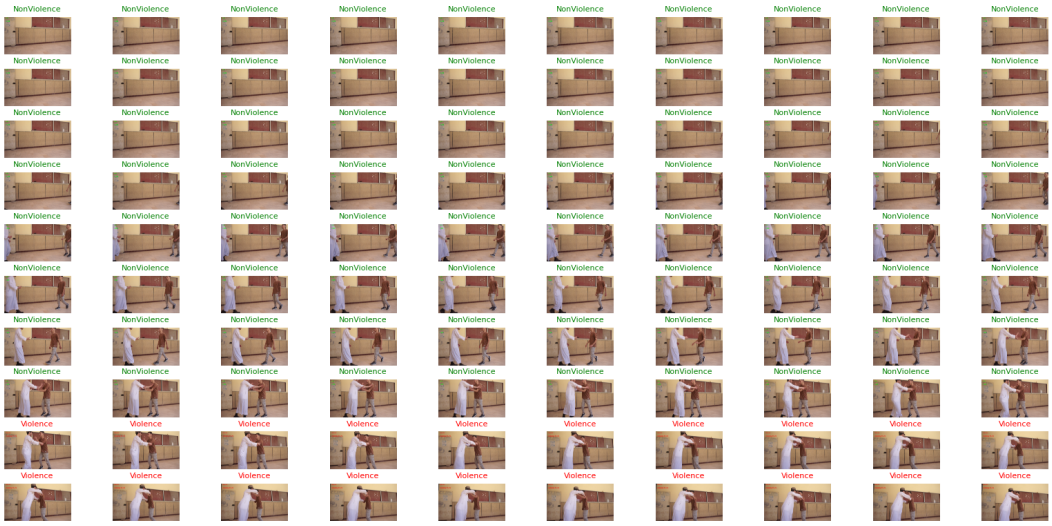

FIGURE 4.26 – Video avec scenes de violence-2

Images où la Détection Fonctionne Bien :

Les images sélectionnées montrent des scènes où notre modèle identifie correctement des comportements violents. Ce succès est attribué à la clarté et à la visibilité des actions violentes, ainsi qu'à la capacité du modèle à capturer des caractéristiques visuelles distinctives, telles que les gestes agressifs ou les interactions physiques intenses. Ces résultats montrent une adéquation entre les caractéristiques détectées par le modèle et les comportements violents attendus, renforçant ainsi la fiabilité de notre approche.

Images où la Détection Échoue Dans d'autres captures d'écran, le modèle rencontre des difficultés à identifier les comportements violents. Les raisons peuvent inclure des conditions d'éclairage défavorables, des angles de caméra peu favorables, ou des actions subtiles qui ne correspondent pas aux modèles appris par le réseau neuronal. Cette variation souligne l'importance d'améliorer la robustesse du modèle face à des conditions environnementales variables et à une diversité d'actions potentiellement violentes.

4.8 Défis et Solutions

Lors de la mise en œuvre de notre système de détection de violence, nous avons rencontré plusieurs défis significatifs qui ont nécessité des solutions innovantes et des ajustements méthodologiques. Voici un aperçu des principaux défis et des approches que nous avons adoptées pour les surmonter :

4.8.1 Variabilité des Conditions de Lumière et de Mouvement

Les vidéos présentant des scènes de violence peuvent varier considérablement en termes de conditions de lumière, d'angles de caméra et de mouvements brusques. Cette variabilité rend difficile l'extraction de caractéristiques cohérentes et fiables.

Solution : Nous avons utilisé des techniques d'augmentation de données pour simuler diverses conditions d'éclairage et d'angles de prise de vue. Ces techniques comprenaient la rotation, le recadrage. Cela a permis d'augmenter la robustesse du modèle face à des variations dans les données d'entrée.

4.8.2 Limitation de la Capacité de Traitement

Le traitement de vidéos en temps réel nécessite une capacité de calcul considérable, ce qui peut être une contrainte pour les systèmes avec des ressources limitées.

Solution : Nous avons exploité la puissance de Google Colab pour l'entraînement de notre modèle, bénéficiant ainsi de GPU gratuits et de ressources de calcul en ligne. De plus, l'utilisation de MobilenetV1, un modèle léger et efficace, a permis de réduire les exigences de calcul tout en maintenant des performances élevées.

4.8.3 Imbalance des Classes

Dans notre dataset, les vidéos de violence étaient moins nombreuses que les vidéos non violentes, créant un déséquilibre de classes qui pourrait biaiser le modèle.

Solution : Pour traiter ce problème, nous pouvons utilisé des techniques de suréchantillonnage des classes minoritaires et de sous-échantillonnage des classes majoritaires. De plus, nous pouvons appliqué des méthodes de pondération des classes dans la fonction de perte pour accorder plus d'importance aux classes minoritaires.

4.8.4 Défis d'Implémentation sur Raspberry Pi 4

L'intégration d'un système de détection de violence sur Raspberry Pi a rencontré plusieurs défis techniques, principalement liés à la compatibilité des bibliothèques et au temps limité disponible pour la mise en œuvre complète.

Problèmes de Compatibilité avec TensorFlow Lite L'un des défis majeurs était la compatibilité de TensorFlow Lite avec l'environnement Raspberry Pi. TensorFlow Lite, conçu pour des appareils mobiles et des systèmes embarqués, nécessite souvent des versions spécifiques de bibliothèques et peut rencontrer des limitations matérielles sur les plateformes comme le Raspberry Pi. La configuration et l'optimisation de TensorFlow Lite pour une exécution efficace sur ce type de matériel peuvent demander des ajustements significatifs et parfois des compromis en termes de fonctionnalités ou de performance.

Problèmes de Bibliothèques et de Dépendances En raison des différences d'architecture et de système d'exploitation entre un ordinateur de bureau et un Raspberry Pi, l'installation et la gestion des bibliothèques nécessaires pour le traitement d'images et l'inférence de modèles de machine learning peuvent être complexes. Des erreurs de compilation, des incompatibilités de versions, ou des dépendances manquantes peuvent survenir, nécessitant une résolution minutieuse et souvent du temps pour rechercher et corriger ces problèmes.

Contraintes de Temps La contrainte de temps a également été un facteur limitant dans la mise en œuvre complète du système sur Raspberry Pi. Les ajustements nécessaires pour résoudre les problèmes de bibliothèques et de compatibilité avec TensorFlow Lite prennent souvent plus de temps que prévu initialement. La nécessité de trouver des solutions alternatives ou de simplifier le processus d'installation pour respecter les délais a été un défi constant.

Ces défis et les solutions apportées démontrent notre engagement à développer un système de détection de violence fiable et performant. En surmontant ces obstacles, nous avons pu améliorer considérablement la robustesse et l'efficacité de notre modèle, aboutissant à une solution capable de fonctionner efficacement dans des environnements variés et complexes.

4.9 Conclusion

Ce chapitre a détaillé la mise en œuvre de notre système de détection de violence, résultant d'une recherche approfondie et d'une expérimentation rigoureuse. Nous avons consacré beaucoup de temps à explorer diverses approches et à tester de nombreux modèles avant d'arriver à notre solution actuelle.

Nous avons commencé par la configuration de l'environnement de travail, en expliquant les outils et les logiciels nécessaires. Ensuite, nous avons décrit le processus de création de notre dataset, compilé à partir de plusieurs sources en ligne pour garantir sa diversité et sa représentativité.

Notre parcours de recherche nous a conduits à expérimenter avec plusieurs architectures de réseaux neuronaux. Après une évaluation comparative approfondie, nous avons sélectionné la combinaison optimale pour notre application. L'entraînement de notre modèle a été réalisé sur Google Colab, en utilisant le transfert d'apprentissage pour améliorer l'efficacité de l'extraction de caractéristiques.

Nous avons suivi plusieurs étapes, depuis la préparation des données jusqu'à l'optimisation des hyperparamètres. Nous avons également présenté une évaluation complète des performances du modèle en utilisant des métriques clés telles que la précision,et la perte.

Enfin, nous avons discuté des défis rencontrés et des solutions innovantes mises en œuvre pour les surmonter. Ce chapitre témoigne de notre engagement à explorer, comparer et perfectionner chaque aspect de notre approche pour développer un système de détection de violence efficace et précis.

Conclusion générale

Dans ce projet, nous avons développé un système de vidéosurveillance innovant capable d'identifier les comportements violents en temps réel, en utilisant l'intégration de MobilenetV1 et LSTM. Nous avons commencé par collecter et intégrer plusieurs kits de jeux de données, y compris des clips avec plusieurs modes d'éclairage, des angles photographiques et différents modèles de mouvement. Nous avons formé ce modèle en utilisant Google Colab pour tirer parti de la puissance de traitement GPU disponible, ce qui a contribué à accélérer le processus d'apprentissage.

MobilenetV1 a été sélectionné en raison de sa capacité à apprendre rapidement, de sa structure légère et de sa capacité à extraire efficacement les fonctionnalités des vidéos. Nous avons ajouté le réseau LSTM pour classer les fonctionnalités extraites en comportements violents et ordinaires.

Cette formation a abouti à un taux de précision de détection élevé de 94%, dépassant les méthodes traditionnelles basées sur des algorithmes de détection de mouvement simples. Nous avons développé une interface utilisateur utilisant TKINTER qui permet aux utilisateurs d'effectuer des détections via des vidéos préenregistrées ou des émissions de caméras en direct.

Grâce à ce travail, nous avons eu l'occasion de découvrir un nouveau domaine intéressant, ainsi que d'utiliser plusieurs technologies que nous pouvons utiliser dans de vastes domaines de la vie. À l'avenir, nous aspirons à étendre les capacités du système pour inclure l'identification de personnes spécifiques ou d'objets suspects, augmentant ainsi son efficacité dans les applications de surveillance réelles. En outre, nous voyons la possibilité d'intégrer d'autres technologies d'IA telles que la reconnaissance vocale pour améliorer la résolution globale du système.

Bibliographie

[1] Videosurveillance-boutique, *Qu'est-ce que la vidéo surveillance ?*, n.d., https://www.videosurveillance-boutique.fr/support/quest-ce-que-la-video-surveillance-271.html, Consulté le 02/03/2024.

[2] Deep Sentinel, *A Brief History Of Surveillance Cameras*, 2022, https://www.deepsentinel.com/blogs/home-security/history-of-surveillance-cameras/, Consulté le 02/03/2024.

[3] Vision Detection Systems, *How Security Cameras Have Evolved Over Time*, n.d., https://visiondetectionsystems.com/how-security-cameras-have-evolved-over-time/, Consulté le 02/03/2024.

[4] Secure-pro, *The History of CCTV*, 2022, https://secure-pro.co.uk/cctv/history-cctv-surveillancecameras-throughtime/, Consulté le 02/03/2024.

[5] Le guide de la video surveillance, 2014, https://www.avenuedelasecurite.fr/img/videosurveillance_guide_achat.pdf. Consulté le 02/03/2024.

[6] Sintrones, *Intelligent Surveillance*, n.d., https://www.sintrones.com/solution/intelligent-surveillance/, Consulté le 02/03/2024.

[7] Istockphot, *Surveillance Vidéos HD libres de droit*, n.d., https://www.istockphoto.com/fr/vidos/surveillance, Consulté le 02/03/2024.

[8] Alchera, *Analyzing Physical Actions*, n.d., https://alchera.ai/en/technology/visual-ai/behavior-analysis, Consulté le 02/03/2024.

[9] Towardsdatascience, *Object Detection with 10 lines of code*, n.d., https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606, Consulté le 02/03/2024.

[10] Canon Global, *Counting People in Crowds with AI*, n.d., https://global.canon/en/technology/count2019.html, Consulté le 02/03/2024.

[11] Matroid, *Empowering Conservation with Matroid's AI-Enabled Environmental Monitoring*, n.d., https://www.matroid.com/ai-enabled-environmental-monitoring-for-conservation/.

[12] Nsoft Vision, *Traditional vs. AI Video Surveillance : A Comparative Insight*, 2024, https://www.nsoft.vision/traditional-vs-ai-video-surveillance, Consulté le 27/05/2024.

[13] Venu, N. and Arun Kumar, A., *Suspicious Activity Tracking Artificial Intelligence Camera*, High Technology Letters, vol. 29, no. 10, 2023, https://doi.org/10.37896/HTL29.10/9518, Consulté le 02/03/2024.

[14] Eocortex, *Algorithms for solving abandoned objects detection issues*, 2024, https://eocortex.com/company/blog/algorithms-for-solving-abandoned-objects-detection-issues, Consulté le 26/05/2024.

[15] Viso.ai, *Fall Detection*, 2024, https://viso.ai/application/fall-detection, Consulté le 26/05/2024.

[16] Medium, *Build a Weapon Detection Algorithm using YOLOv7*, 2024, https://medium.com/the-modern-scientist/build-a-weapon-detection-algorithm-using-yolov7-8d1787c93f96, Consulté le 26/05/2024.

[17] staqu, *Artificial Intelligence For Public Sector*, 2024, https://www.staqu.com/solutions/public-sector, Consulté le 27/05/2024.

[18] Act-my, *INTRUSION DETECTION AND PERIMETER PROTECTION*, 2024, https://act-my.com/ai-solutions/intrusion-detection-system, Consulté le 26/05/2024.

[19] IBM, *Topics : Artificial Intelligence*, https://www.ibm.com/fr-fr/topics/artificial-intelligence. Consulté le 02/05/2024.

[20] *Origine de l'AI*, https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/. Consulté le 02/05/2024.

[21] *Histoire de l'Intelligence Artificielle*, https://www.ia-insights.fr/histoire-de-l-intelligence-artificielle/. Consulté le 02/05/2024.

[22] *Fonction du Perceptron*, https://datascientest.com/perceptron. Consulté le 18/05/2024.

[23] *Limitation du Perceptron*, https://cdn.prod.website-files.com/62233c592d2a1e009d42f46c/6275187a42e01b02d1b21cfe_61ead6b9d5b1b343ea9151b2_Capture%2520d%25E2%2580%2599e%25CC%2581cran%25202022-01-21%2520a%25CC%2580%252016.52.12.png.Consulté le 18/05/2024.

[24] *Medium, Convolutional Neural Networks (CNN) — Architecture Explained*, https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243.Consulté le 15/05/2024.

[25] Goodfellow, Ian, et al., *Deep Learning*, MIT Press, 2016, Chapitre 5 : Optimization for Training Deep Models.

[26] Nielsen, Michael, *Neural Networks and Deep Learning : Determining the Loss*, 2015, http://neuralnetworksanddeeplearning.com/chap3.html.Consulté le 08/05/2024.

[27] Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J., *Learning Representations by Back-Propagating Errors*, Nature, vol. 323, no. 6088, pp. 533-536, 1986.

[28] Kingma, Diederik P. and Ba, Jimmy, *Adam : A Method for Stochastic Optimization*, International Conference on Learning Representations (ICLR), 2015.

[29] Sokolova, Marina and Lapalme, Guy, *A Systematic Analysis of Performance Measures for Classification Tasks*, Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009.

[30] Goodfellow, Ian, et al., *Deep Learning*, MIT Press, 2016, Chapitre 5 : Optimization for Training Deep Models.

[31] *Figure Gradient*, https://seshat.gitlabpages.inria.fr/deeplearning/docHtml/IA1.html#r%C3%A9gression-logistique.Consulté le 15/05/2024.

[32] *Techxplore .System for drone surveillance : How violence is boxed*, https://techxplore.com/news/2018-06-drone-surveillance-violence.html#google_vignette. 12/06/2024.

[33] *CNN Construction*, https://www.memoireonline.com/12/22/13648/m_Construction-dun-modele-predictif-base-sur-le-reseau-de-neurones-profond-pour-la-html. Consulté le 05/06/2024.

[34] *Construction d'un Modèle Predictif basé sur le Réseau de Neurones Profond pour la Détection*, https://www.memoireonline.com/12/22/13648/m_Construction-dun-modele-predictif-base-sur-le-reseau-de-neurones-profond-pour-la-html. Consulté le 05/06/2024.

[35] *Structure du Reseau de Neurones*, https://www.tensorflow.org/api_docs/python/tf/keras/layers/Layer.

[36] *L'environnement de Developpement*, https://keras.io/api/.Consulté le 12/06/2024.

[37] *L'environnement de Developpement*, https://www.tensorflow.org/guide/keras/sequential_model. Consulté le 12/06/2024.

[38] Fonction d'Activation - Comment ça marche ?, https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/#:~:text=La%20fonction%20Rectified%20Linear%20Unit,couramment%20utilis%C3%A9e%20en%20Deep%20Learning.&text=Cette%20fonction%20permet%20d'appliquer,et%20bloque%20les%20valeurs%20n%C3%A9gatives, Consulté le 02/06/2024.

[39] N. Zehra, S. H. Azeem, M. Farhan, Human Activity Recognition Through Ensemble Learning of Multiple Convolutional Neural Networks, 55th Annual Conference on Information Sciences and Systems (CISS), 2021, pp. 1-5, 10.1109/CISS50987.2021.9400290

[40] Layers in Convolutional Neural Networks, https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network. Consulté le 25/05/2024.

[41] Analytics Vidhya, What is LSTM ? Introduction to Long Short-Term Memory, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/, Consulté le 27/03/2024

[42] colah's blog, Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, Consulté le 27/03/2024

[43] La revue IA, Qu'est ce qu'un réseau LSTM ? https://larevueia.fr/quest-ce-quun-reseau-lstm/, Consulté le 27/03/2024

[44] Coursera, What Is Machine Learning ? Definition, Types, and Examples, https://www.coursera.org/articles/what-is-machine-learning, Consulté le 27/03/2024

[45] Spiceworks, What Is Machine Learning ? Definition, Types, Applications, and Trends, https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/, Consulté le 27/03/2024

[46] Medium, Supervised and Unsupervised Learning (an Intuitive Approach), https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644, Consulté le 20/04/2023

[47] GitHub, Handwriting Recognition (HWR). https://github.com/OliverFlow/HandwritingRecognition?tab=readme-ov-file, Consulté le 20/04/2023

[48] Vector Institute. Vector AI Engineering Blog : Benchmarking Robustness of Reinforcement Learning Approaches using safe-control-gym (en anglais). https://vectorinstitute.ai/fr/vector-ai-engineering-blog-benchmarking-robustness-of-reinforcement-learning-appr Consulté le 20/04/2023

[49] Banuba, Background Subtraction 101 : Meaning, Benefits, Use Cases, https://www.banuba.com/blog/background-subtraction-guide, Consulté le 19/04/2024

[50] Electrical Computer Engineering, Foreground-Adaptive Background Subtraction, https://vip.bu.edu/projects/vsns/background-subtraction/fa/, 2007-2008, Consulté le 19/04/2024

[51] Nanonets, A Comprehensive guide to Motion Estimation with Optical Flow, https://nanonets.com/blog/optical-flow/, Apr 2019, Consulté le 19/04/2024

[52] OpenCV, Optical Flow, https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html, Consulté le 19/04/2024

[53] Jmlb, Haar Cascade Classifier, https://jmlb.github.io/flashcards/2018/06/30/computer_vision_haar_cascade/, Jun 2018, Consulté le 19/04/2024

[54] Machine Learning Mastery, Using Haar Cascade for Object Detection, https://machinelearningmastery.com/using-haar-cascade-for-object-detection/, January 2024, Consulté le 19/04/2024

[55] Analytics Vidhya. Human Pose Estimation Using Machine Learning in Python. https://www.analyticsvidhya.com/blog/2022/01/a-comprehensive-guide-on-human-pose-estimation/. Consulté le 06/06/2024.

[56] Everything You Need to Know About VGG16, https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918, Consulté le 07/06/2024.

[57] Choi, Daegyun Bell, William Kim, Donghoon Kim, Jichul. (2021). UAV-Driven Structural Crack Detection and Location Determination Using Convolutional Neural Networks. Sensors. 21. 2650. 10.3390/s21082650.

[58] Medium. Mastering VGG from Scratch : A Journey of Learning and Growth in Computer Vision. https://pub.aimind.so/mastering-vgg-from-scratch-a-journey-of-learning-and-growth-in-computer-vision-a
Consulté le 06/06/2024.

[59] Sunanda Das, Amlan Sarker, Tareq Mahmud, Violence Detection from Videos using HOG Features, Conference paper at the 4th International Conference on Electrical Information and Communication Technology (EICT), 20-22 December 2019, Khulna, Bangladesh, 2019

[60] Javad Mahmoodi, Afsane Salajeghe, A classification method based on optical flow for violence detection, Sama Technical and Vocational Training College, Islamic Azad University, Kerman Branch, Kerman, Iran

[61] Hongchang Li, Jing Wang, Jianjun Han, Jinmin Zhang, Yushan Yang, Yue Zhao, A novel multi-stream method for violent interaction detection using deep learning, Measurement and Control, vol. 53, no. 5-6, pp. 796–806, 2020

[62] Fernando J. Rendón-Segador, Juan A. Álvarez-García, Jose L. Salazar-González, Tatiana Tommasi, CrimeNet : Neural Structured Learning using Vision Transformer for violence detection, Dept of Computer Languages and Systems, University of Seville, Spain Politecnico di Torino Italian Institute of Technology, Italy, 2023

[63] U. Rachna, Varsha Guruprasad, S. Dhruv Shindhe, S. N. Omkar, Real-Time Violence Detection Using Deep Neural Networks and DTW, CVIP 2022, pp. 316–327, 2023

[64] N. Bagga, G. Singh, B. Balusamy, A. Shanker Singh, Violence Detection in Real Life Videos using Convolutional Neural Network, 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022

[65] A. -M. R. Abdali, R. F. Al-Tuma, Robust Real-Time Violence Detection in Video Using CNN And LSTM, 2nd Scientific Conference of Computer Sciences (SCCS), Baghdad, Iraq, 2019

[66] K. -C. Chang, Y. -C. Liao, Design of Violence Event Detection System Based on CCTVs by Human Body Pose Recognition, International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), PingTung, Taiwan, 2023

[67] M. Datta, M. Shah, N. Da Vitoria Lobo, Person-on-person violence detection in video data, Object recognition supported by user interaction for service robots, 2002

[68] Yu Zhao, Rennong Yang, Guillaume Chevalier, Maoguo Gong, Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors, CoRR, vol. abs/1708.08989, 2017

[69] Nicolas Thiery Name. Cours 6 : Traitement d'images. https://nicolas.thiery.name/Enseignement/IntroScienceDonnees/Devoirs/Semaine6/CM6.html, Consulté le 06/06/2024.

[70] Medium, Comprendre les Réseaux de Neurones Convolutifs (CNN). https://yannicksergeobam.medium.com/comprendre-les-rseaux-de-neurones-convolutifs-cnn-d5f14d963714, Consulté le 06/06/2024.

[71] Open Datascience, Understanding the Mechanism and Types of Recurrent Neural Networks. https://opendatascience.com/understanding-the-mechanism-and-types-of-recurring-neural-networks/. Consulté le 08/07/2024.

[72] Medium, Single Layer Perceptron and Activation Function. https://medium.com/codex/single-layer-perceptron-and-activation-function-b6b74b4aae66, Consulté le 07/06/2024.

[73] Irfan, F. A., & Karim, M. (2021). *Detection of Underwater Ships Categorically Using Convolutional Neural Network and Power Efficient Remotely Operated Vehicle.*

[74] Daniel's Blogs, MobileNet Architecture https://danghoangnhan.github.io/MobileNetArchitecture/, Consulté le 02/03/2024.Consulté le 06/06/2024.

[75] AIML.com, What is the vanishing and exploding gradient problem, and how are they typically addressed ? https://aiml.com/what-do-you-mean-by-vanishing-and-exploding-gradient-problem-and-how-are-they-ty le 08/07/2024.

[76] Kaggle, Hockey Fight Vidoes. https://www.kaggle.com/datasets/yassershrief/hockey-fight-vidoes.Consulté le 08/07/2024.

[77] Kaggle, Real Life Violence Situations Dataset. https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset.Consulté le 08/07/2024.

[78] Kaggle, Violence Detection Dataset. https://www.kaggle.com/datasets/mehedihasannirob/violence-detection-dataset/data.Consulté le 08/07/2024.

[79] Kaggle, PUA-DATASET. https://www.kaggle.com/datasets/habibaahmedelshabasy/pua-dataset.Consulté le 08/07/2024.

[80] Marc Sauget. Parallélisation de problèmes d'apprentissage par des réseaux neuronaux artificiels. Application en radiothérapie externe 2007.

[81] Patil, Krishna Kunte, Sanket Shah, Preetam. (2023). Artificial Intelligence in Pediatric Dentistry A.2. 10.4103/jdrr.jdrr-199-2.

[82] Phiphitphatphaisit, Sirawan Surinta, Olarik. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation. 51-56. 10.1145/3388176.3388179.

[83] minimin2, [] MobileNet V1 , pytorch (depthwise separable convolution) `https://minimin2.tistory.com/42`, Consulté le 08/07/2024.