



Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université de Carthage

Institut National des Sciences Appliquées et de Technologie



RAPPORT DE PROJET DE FIN D'ANNÉE

Filière : Réseaux Informatiques et Télécommunications

Par

Ahmed Amine AMOUCI , Hamza ROUAISSI , Katr-nada LAHBIB

Prévision de la Criminalité : Vision par Ordinateur pour la prévision et la prévention de la criminalité

SH1

Encadrant académique :

Madame Sana HAMDI

Maître-Assistante

Réalisé au sein de l'Institut National des Sciences Appliquées et de Technologie (INSAT)



Remerciements

C'est avec un immense plaisir que nous réservons ces quelques lignes, au terme de ce modeste travail, en signe de gratitude et de reconnaissance à toutes les personnes qui ont contribué, de près ou de loin, à son bon déroulement.

Nous tenons, particulièrement, à dédier nos remerciements les plus chaleureux et sincères à notre encadrante, Mme Sana HAMDI, maître-assistante à l'Institut National des Sciences Appliquées et de Technologie (INSAT), pour son aide, son soutien, sa disponibilité, ses encouragements et ses précieux conseils qui ont été remarquables tout au long de la réalisation du projet.

Avec le plus grand respect, nous ne manquerons pas également d'exprimer notre profonde gratitude aux membres du jury d'avoir accepté de juger notre humble travail, en espérant qu'ils trouveraient ce qu'ils attendaient.

Et bien sûr, nous avons été très heureux de travailler ensemble sur ce projet. Pour cela, nous tenons à nous remercier mutuellement. C'était une excellente ambiance de travail.

Table des matières

Introduction générale	1
1 Cadre général du projet	3
1.1 Description	4
1.2 Définitions et concepts généraux	4
1.2.1 Apprentissage automatique (Machine Learning)	4
1.2.2 Apprentissage profond (Deep Learning)	5
1.2.3 Apprentissage faiblement supervisé	5
1.2.4 Réseau de neurones profond (DNN)	5
1.2.5 Réseau de neurones convolutif (CNN)	6
1.2.6 Le réseau de neurones récurrent (RNN)	7
1.2.7 Long short terme memory (LSTM)	7
1.2.8 Vision par ordinateur (Computer Vision)	8
1.2.9 Detection d'anomalies	8
1.2.10 C3D	8
1.2.11 I3D two stream (RGBs FLOWS)	9
1.2.12 Internet of Things (IoT)	9
1.3 Objectifs et motivation	9
1.4 Solution existante	10
1.4.1 Architecture	10
1.4.2 Description	10
1.4.3 Limites	10
1.5 Solution proposée	10
1.6 Méthodologie de travail	11
2 Collecte et préparation des données	13
2.1 Compréhension des données	14
2.2 Collecte des données	14
2.3 Préparation des données	16
3 Modélisation	18
3.1 Critères de réussite du modèle	19

3.2	Langages et outils utilisés	19
3.2.1	Langages	19
3.2.2	Frameworks	20
3.2.3	Éditeurs	20
3.2.4	Contrôle de version	21
3.3	Architecture	22
3.4	Description	22
4	Évaluation	24
4.1	Métrique	25
4.2	Tableau comparatif	28
4.3	Interprétation	29
5	Production	30
5.1	Conception IoT	31
5.2	Description	32
	Conclusion générale et perspectives	34

Table des figures

1.1	Intelligence artificielle, Machine Learning et Deep Learning	5
1.2	Architecture d'un réseau DNN	6
1.3	Architecture standard d'un réseau CNN	6
1.4	Architecture standard d'un réseau RNN	7
1.5	Architecture d'un réseau LSTM	7
1.6	Exemple d'application de Computer Vision	8
1.7	Architecture globale de I3D two stream	9
1.8	Architecture du modèle de detection d'anomalies adopté dans la solution existante . .	10
1.9	Modèle CRISP-DM	12
2.1	Nombre total des vidéos par anomalie	15
2.2	Distribution des vidéos normales et anormales entre training et testing sets	16
2.3	Arborescence des données	17
3.1	Sous-apprentissage, apprentissage correct et sur-apprentissage	19
3.2	Logo de <i>Python</i>	20
3.3	Logo de <i>PyTorch</i>	20
3.4	Logo de <i>Pycharm</i>	20
3.5	Logo de <i>Google Colaboratory</i>	21
3.6	Logo de <i>Git</i>	21
3.7	Logo de <i>GitHub</i>	21
3.8	Architecture du modèle de detection d'anomalies adopté dans notre solution proposé .	22
3.9	Arborescence du code	22
4.1	Matrice de confusion	25
4.2	Courbe ROC et Aire sous la courbe AUC	26
4.3	Fonction de perte calculée à chaque epoch lors de l'entraînement du modèle	27
4.4	Précision calculée lors du test à chaque epoch	27
4.5	Rappel calculé lors du test à chaque epoch	28
4.6	Courbe ROC et Aire sous la courbe AUC de notre modèle en comparaison avec la solution existante	28
4.7	Tableau comparatif des résultats obtenus	28

5.1	Conception IOT de la méthode de production du projet	31
5.2	Caméra intelligente	31
5.3	Les services du Cloud	32
5.4	Logo de AWS	32

Liste des abréviations

—	C3D	=	Convolutional 3-Dimensional
—	CNN	=	Convolutional Neural Networks
—	DL	=	Deep Learning
—	DNN	=	Deep Neural Network
—	I3D	=	Inflated 3D ConvNets
—	IoT	=	Internet of Things
—	LSTM	=	Long Short Term Memory
—	ML	=	Mmachine Learning
—	RNN	=	R ecurrent Neural Networks
—	UCF	=	University of C entral F lorida

Introduction générale

Dans le scénario actuel d'augmentation rapide et alarmante du nombre et des formes d'activités criminelles, les techniques traditionnelles de résolution des crimes sont devenues incapables de donner des résultats car elles sont lentes et moins efficaces. Par conséquent, le besoin d'un système prédictif s'est accru.

En effet, la prédiction des crimes devient de plus en plus cruciale, car elle peut potentiellement sauver la vie d'une victime, prévenir un traumatisme à vie, et éviter des dommages aux propriétés publiques et privées, il est donc nécessaire pour nous de fournir à la police et aux organismes autoritaires des moyens de prédire les crimes, en détail, avant qu'ils ne se produisent, ce qui va les aider dans leur processus de résolution de criminalité de manière beaucoup plus précise et rapide. C'est l'introduction de la notion de la police prédictive avec un niveau de précision considérable.

Pour ce faire, nous recourons à la vidéosurveillance qui a pour principale tâche de détecter les événements anormaux tels que les accidents de la route, les crimes et les activités illégales, et vu que les caméras de surveillance sont de plus en plus utilisées dans les lieux publics (tels que les rues, les carrefours, les banques, les centres commerciaux, etc.) afin de renforcer la sécurité publique.

Résumé

Notre projet de fin d'année « *Prévision De La Criminalité : Vision Par Ordinateur Pour La Prévision Et La Prévention De La Criminalité* » consiste à concevoir un modèle de détection profonde des anomalies utilisant des vidéos d'entraînement faiblement étiquetées, dont l'objectif est la détection générale d'anomalies en considérant toutes les anomalies dans un groupe et toutes les activités normales dans un autre groupe et fournir une notification précoce de l'activité qui s'écarte des modèles normaux en identifiant la fenêtre temporelle de l'anomalie qui se produit, et éventuellement, la reconnaissance de chacune des anomalies réalistes (Abus, arrestation, incendie criminel, agression, accident de la route, cambriolage, explosion, bagarre, braquage, fusillade, vol, vol à l'étalage et vandalisme).

Afin d'illustrer l'approche de notre projet, la méthodologie CRISP-DM (Cross-Industry Standards Process for Data Mining) était notre meilleure option, car elle fournit une approche structurée de la planification d'un projet d'extraction de données.

Nous présentons ci-dessous l'organisation générale du rapport, qui s'articule autour de cinq chapitres : Dans le premier chapitre, nous exposerons le cadre général du projet dans lequel nous explicitons le contexte de notre projet et ses objectifs et motivation, quelques définitions et concepts généraux, nous procédons ainsi à une discussion de la solution existante et de notre solution proposée, pour finir par l'exposition de notre méthodologie de travail.

Le deuxième chapitre est intitulé collecte et préparation de donnée dans lequel nous présentons les données nécessaires à la réalisation de ce projet, les sources de ces données et les opérations effectuées lors de la phase de préparation des données et l'extraction des paramètres (features extraction).

Le troisième chapitre fera l'objet de la modélisation de la solution proposée dans lequel nous citons les critères de réussite du modèle, nous détaillons de même les langages et outils utilisés, ainsi que l'architecture du modèle adopté.

Le quatrième chapitre sera consacré à l'évaluation globale du modèle dans lequel nous présentons les métriques calculées permettant de comparer les résultats générés par ce modèle avec les performances de la solution existante, cela nous permet de mieux interpréter les points forts et faibles de notre modèle.

Le dernier chapitre va traiter la partie production dans lequel nous dévoilons l'idée générale de la méthode de production imaginée de notre projet ainsi que les technologies et outils à utiliser. Nous clôturons notre rapport par une conclusion générale résumant les points essentiels de notre travail accompagnés des perspectives du projet et ses pistes d'amélioration.

CADRE GÉNÉRAL DU PROJET

Plan

1	Description	4
2	Définitions et concepts généraux	4
3	Objectifs et motivation	9
4	Solution existante	10
5	Solution proposée	10
6	Méthodologie de travail	11

Introduction

Ce chapitre est consacré à la compréhension du cadre général du projet, c'est une étape très importante dans la mesure où elle met en évidence la motivation derrière le projet et donne une vision claire de ce que nous essayons d'accomplir et la valeur ajoutée de ce projet en tenant compte de tous les facteurs qui pourraient avoir le potentiel d'influencer le résultat final. Tout d'abord, nous entamons par la description du projet et de son contexte et bien sûr nous avons quelques définitions et concepts généraux à présenter, puis nous cédon place aux objectifs et motivation de ce travail, ensuite nous discutons la solution existante et notre solution proposée, pour finir par l'exposition de notre méthodologie de travail.

1.1 Description

Les caméras de surveillance sont capables de capturer une grande quantité de données sur la criminalité qui pourraient être utilisées pour nous informer sur les tendances et les modèles de criminalité actuels et futurs qui s'améliorent d'année en année grâce aux progrès de la technologie. L'analyse prédictive vise à optimiser l'utilisation de ces données pour anticiper les événements criminels et par suite fournir à la police et au gouvernement les moyens d'une nouvelle et puissante machine qui peut les aider dans leur processus de résolution des crimes. Notre projet consiste à étudier l'impact de la combinaison des algorithmes de l'apprentissage profond et automatique avec les méthodes de vision par ordinateur pour la détection et la prédiction des anomalies et éventuellement la classification de ces anomalies selon la nature des activités criminelles.

1.2 Définitions et concepts généraux

1.2.1 Apprentissage automatique (Machine Learning)

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des "patterns", à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques... Tout ce qui peut être stocké numériquement peut servir de données pour le Machine Learning. En décelant les patterns dans ces données, les algorithmes apprennent et améliorent leurs performances dans l'exécution d'une tâche spécifique. En bref, les algorithmes de Machine Learning apprennent de manière autonome à effectuer une tâche ou à réaliser des prédictions à partir de données et améliorent leurs performances au fil du temps. Une fois entraîné, l'algorithme pourra retrouver les patterns dans de nouvelles données.

1.2.2 Apprentissage profond (Deep Learning)

Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées. L'apprentissage profond s'appuie sur un réseau de neurones artificielles s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de «couches» de neurones, chacune recevant et interprétant les informations de la couche précédente.

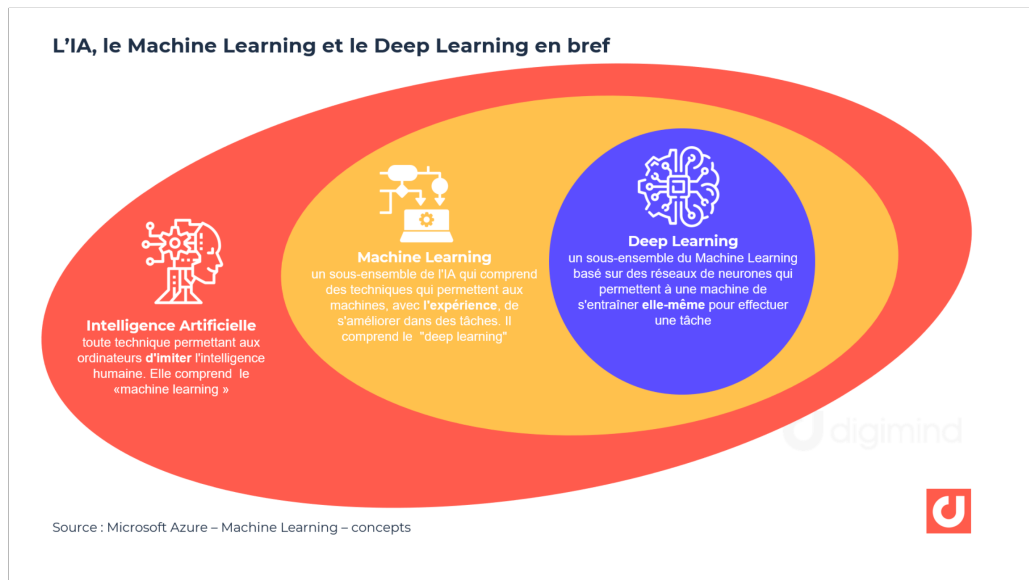


FIGURE 1.1 : Intelligence artificielle, Machine Learning et Deep Learning

1.2.3 Apprentissage faiblement supervisé

La supervision faible est une branche de l'apprentissage automatique où des sources bruyantes, limitées ou imprécises sont utilisées pour fournir un signal de supervision pour l'étiquetage de grandes quantités de données d'apprentissage dans un cadre d'apprentissage supervisé. Cette approche évite d'avoir à obtenir des ensembles de données étiquetées manuellement, ce qui peut être coûteux ou peu pratique.

1.2.4 Réseau de neurones profond (DNN)

Un réseau de neurones est un ensemble d'algorithmes inspirés par le cerveau humain. Le but de cette technologie est de simuler l'activité du cerveau humain, et plus spécifiquement la reconnaissance de motifs et la transmission d'informations entre les différentes couches de connexions neuronales. Un Deep Neural Network, ou réseau de neurones profond, se distingue par une particularité : il est composé d'au moins deux couches. Ceci lui permet de traiter les données de manière complexe, en employant

des modèles mathématiques avancés.

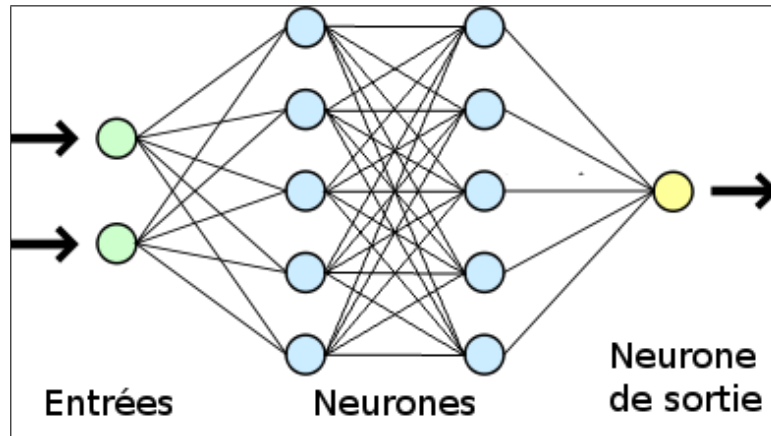


FIGURE 1.2 : Architecture d'un réseau DNN

1.2.5 Réseau de neurones convolutif (CNN)

Les réseaux de neurones convolutifs sont une forme particulière de réseau neuronal multicouche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères. Leur conception suit la découverte de mécanismes visuels dans les organismes vivants. Ces réseaux de neurones artificiels sont capables de catégoriser les informations des plus simples aux plus complexes. Ils consistent en un empilage multicouche de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui pré-traitent de petites quantités d'informations. Une couche convolutive, est basée comme son nom l'indique sur le principe mathématique de convolution, et cherche à repérer la présence d'un motif (dans un signal ou dans une image par exemple). Une phase d'apprentissage sur des objets connus permet de trouver les meilleurs paramètres en montrant par exemple à la machine des milliers d'images d'un chien, d'une voiture ou d'un sport. L'un des enjeux est de trouver des méthodes pour ajuster ces paramètres le plus rapidement et le plus efficacement possible. Les réseaux neuronaux convolutifs ont de nombreuses applications dans la reconnaissance d'images, de vidéos ou le traitement du langage naturel.

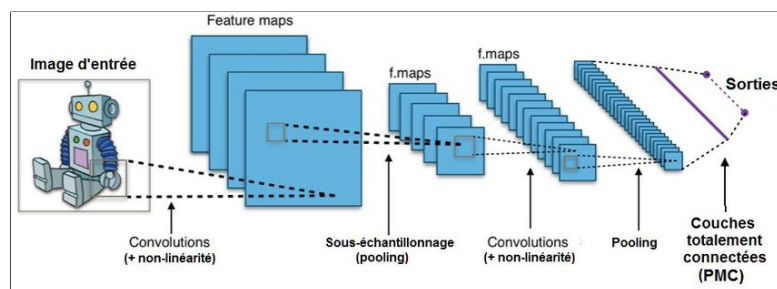


FIGURE 1.3 : Architecture standard d'un réseau CNN

1.2.6 Le réseau de neurones récurrent (RNN)

Le réseau neuronal récurrent, ou RNN, est un réseau neuronal multicouche utilisé pour analyser des entrées séquentielles, telles que du texte, de la parole ou des vidéos, à des fins de classification et de prédiction. Il traite les données d'une manière comparable à la perception d'un être humain. En effet, nous, en tant qu'humains, nous avons des pensées persistantes, ce qui signifie que nous traitons chaque mot en fonction du contexte des mots précédents. Les réseaux neuronaux récurrents abordent cette question de la persistance avec des boucles intégrées qui permettent la persistance de l'information.

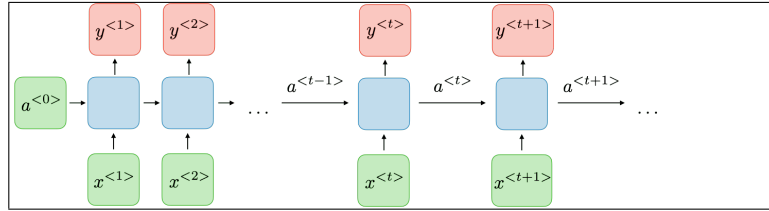


FIGURE 1.4 : Architecture standard d'un réseau RNN

1.2.7 Long short terme memory (LSTM)

C'est une architecture de réseau de neurones récurrents (RNN) utilisé dans le domaine de l'apprentissage profond (deep learning). À la différence des réseaux neuronaux à propagation avant, le LSTM a des connections de feedback. Il peut non seulement traiter des data points uniques (tels que des images), mais également des séquences complètes de données (telles que la parole ou la vidéo). Une unité LSTM de base est composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli. La cellule se souvient des valeurs sur des intervalles de temps arbitraires et les trois portes régulent le flux d'information entrant et sortant de la cellule. Les réseaux LSTM sont bien adaptés à la classification, au traitement et à la prévision sur la base de données chronologiques, car il peut exister des décalages d'une durée inconnue entre les événements importants d'une série chronologique. Les LSTMs ont été développés pour faire face aux problèmes de gradients qui peuvent être rencontrés lors de l'entraînement des RNNs traditionnels.

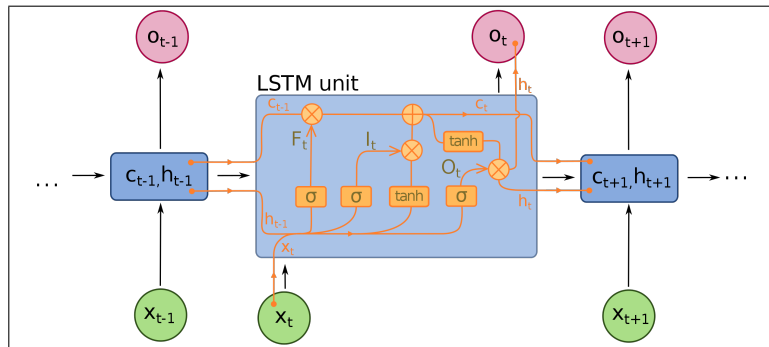


FIGURE 1.5 : Architecture d'un réseau LSTM

1.2.8 Vision par ordinateur (Computer Vision)

La computer vision désigne une technique d'intelligence artificielle permettant d'analyser des images captées par un équipement tel qu'une caméra. Concrètement, la computer vision se présente comme un outil basé sur l'IA capable de reconnaître une image, de la comprendre, et de traiter les informations qui en découlent. Pour beaucoup, la vision par ordinateur est l'équivalent, en termes d'IA, des yeux humains et de la capacité de notre cerveau à traiter et analyser les images perçues. La reproduction de la vision humaine par des ordinateurs constitue d'ailleurs l'un des grands objectifs de la computer vision.

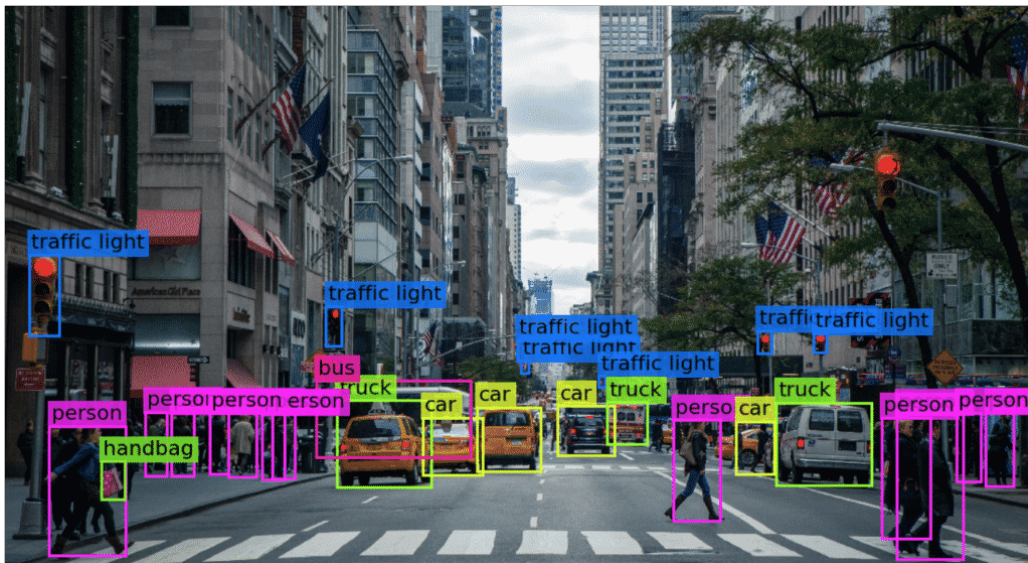


FIGURE 1.6 : Exemple d'application de Computer Vision

1.2.9 Détection d'anomalies

La détection d'anomalies est la technique d'identification d'événements ou d'observations rares qui peuvent soulever des soupçons en étant statistiquement différente du reste des observations. Un tel comportement «anormal» se traduit généralement par une sorte de problème comme une fraude par carte de crédit, une machine défaillante dans un serveur, une cyberattaque, etc.

1.2.10 C3D

Les C3D sont des réseaux neuronaux convolutifs tridimensionnels profonds avec une architecture homogène contenant $3 \times 3 \times 3$ noyaux convolutifs suivis de $2 \times 2 \times 2$ pooling à chaque couche. Le C3D est un descripteur de segment vidéo très courant qui intègre à la fois des caractéristiques de mouvement et des caractéristiques temporelles. Ainsi, il est très intuitif d'utiliser les paramètres C3D comme première étape avant un traitement ultérieur pour diverses applications.

1.2.11 I3D two stream (RGBs FLOWs)

Les paramètres de la 3D gonflée (Inflated 3D : I3D) sont extraits à l'aide d'un modèle pré-entraîné. Ici, les paramètres sont extraits de l'avant-dernière couche de I3D, avant d'être additionnées. Par conséquent, il produit deux tenseurs avec 1024 paramètres : pour les flux RGB et FLOW. Par défaut, il s'attend à entrer 64 images RGB et FLOW (224x224), soit 2,56 secondes de la vidéo enregistrée à 25 images par seconde.

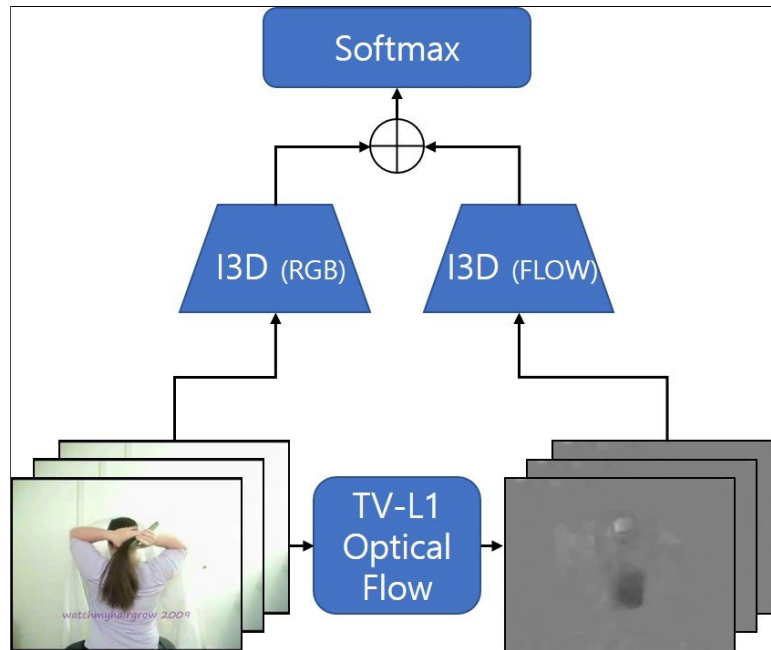


FIGURE 1.7 : Architecture globale de I3D two stream

1.2.12 Internet of Things (IoT)

L'Internet of Things (IoT) décrit le réseau de terminaux physiques, les « objets », qui intègrent des capteurs, des logiciels et d'autres technologies en vue de se connecter à d'autres terminaux et systèmes sur Internet et d'échanger des données avec eux.

1.3 Objectifs et motivation

À la lumière de l'augmentation rapide et alarmante du nombre et des formes d'activités criminelles face à l'incapacité des techniques traditionnelles de résolution des crimes, nous avons pour objectif d'introduire la notion de la police prédictive en offrant l'opportunité de détecter et d'anticiper les actes criminels à travers la mise à disposition d'un outil logiciel permettant l'analyse prédictive des vidéos de surveillance.

1.4 Solution existante

1.4.1 Architecture

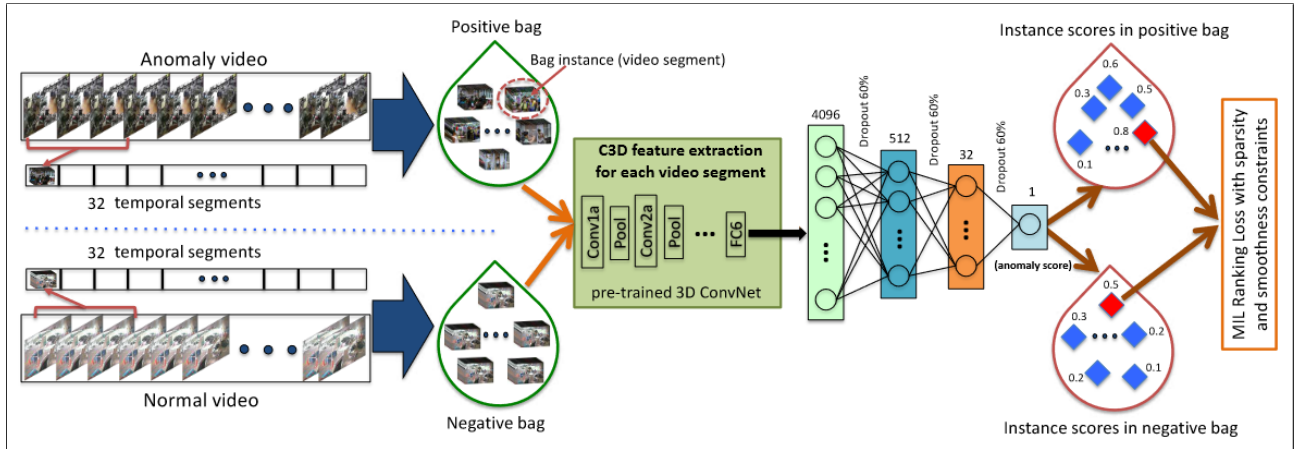


FIGURE 1.8 : Architecture du modèle de detection d'anomalies adopté dans la solution existante

1.4.2 Description

L'approche proposée par cette solution commence par diviser les vidéos de surveillance en un nombre fixe de segments temporels pendant la formation. Ces segments constituent les instances d'un sac : les sacs positifs (contenant l'anomalie quelque part) et négatifs (ne contenant aucune anomalie). Après avoir extraire les paramètres C3D pour les segments vidéo, le modèle est entraîné à détecter des anomalies en se basant sur un réseau neuronal entièrement connecté utilisant une nouvelle fonction de perte de classement qui calcule la perte entre les instances incorrectement classées (en rouge) dans les sacs positif et négatif.

1.4.3 Limites

Cette solution permet d'obtenir des résultats relativement faibles ($AUC = 75$). De plus, elle ne prend pas en considération le séquençement temporel des segments.

1.5 Solution proposée

Notre solution proposée est basée sur la combinaison des réseaux de neurones convolutifs (CNN) et des réseaux de neurones récurrents (RNN) qui produit une architecture puissante pour les problèmes de classification vidéo car elle est à la fois profonde dans la dimension spatiale et temporelle, c.à.d, les informations spatio-temporelles peuvent être traitées simultanément et efficacement.

Le modèle que nous proposons peut être principalement décomposé en deux modules ; Le réseau I3D (Three Dimension Inception) two-stream (FLOWs RGBs), et le réseau LSTM (Long Short Term

Memory) qui est un réseau amélioré basé sur le RNN permettant d'éviter les problèmes de disparition et d'explosion du gradient pendant le processus d'entraînement, pour l'apprentissage des paramètres temporels.

Le réseau I3D est appliqué pour extraire les paramètres spatiaux. Ensuite, le paramètre de sortie appris par le modèle I3D sous-jacent servira d'entrée au réseau LSTM, qui est principalement responsable de la modélisation des paramètres spatiaux de haut niveau. Par conséquent, les paramètres des vidéos d'entrée peuvent être bien appris et représentés, et notre méthode proposée peut très bien apprendre les paramètres de bas niveau et de haut niveau.

1.6 Méthodologie de travail

Pour la méthodologie de travail nous optons pour la méthode CRISP-DM (Cross Industry Standard Process for Data-Mining) développée par IBM et qui permet d'orienter les projets d'exploration de données.

Cette méthodologie contient les phases typiques d'un projet suivantes :

- **La compréhension du cadre général du projet** : Cette phase consiste à bien comprendre les éléments métiers et problématiques qu'on vise à résoudre ou à améliorer.
- **La compréhension et collecte des données** : La phase de compréhension des données consiste à déterminer précisément les données cibles à analyser, à identifier la qualité des données disponibles et à faire le lien entre les données et leur signification d'un point de vue métier.
- **La préparation des données** : la phase de préparation des données regroupe les activités visant à construire l'ensemble de données finales à partir des données brutes initiales.
- **La modélisation** : Cette phase comprend le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle. La modélisation est généralement effectuée en utilisant plusieurs itérations, on exécute plusieurs modèles en utilisant les paramètres par défaut, puis on affine ces derniers ou bien on revient à la phase de préparation des données pour effectuer les manipulations requises par le modèle choisi.
- **L'évaluation** : L'évaluation vise à vérifier le(s) modèle(s) ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus. Elle contribue aussi à la décision de déploiement du modèle ou, si nécessaire, à son amélioration. A ce stade, on teste notamment la robustesse et la précision des modèles étudiés.
- **Le déploiement ou la production** : Cette phase vise à mettre la connaissance obtenue par la modélisation, dans une forme adaptée, et l'intégrer au processus de prise de décision.

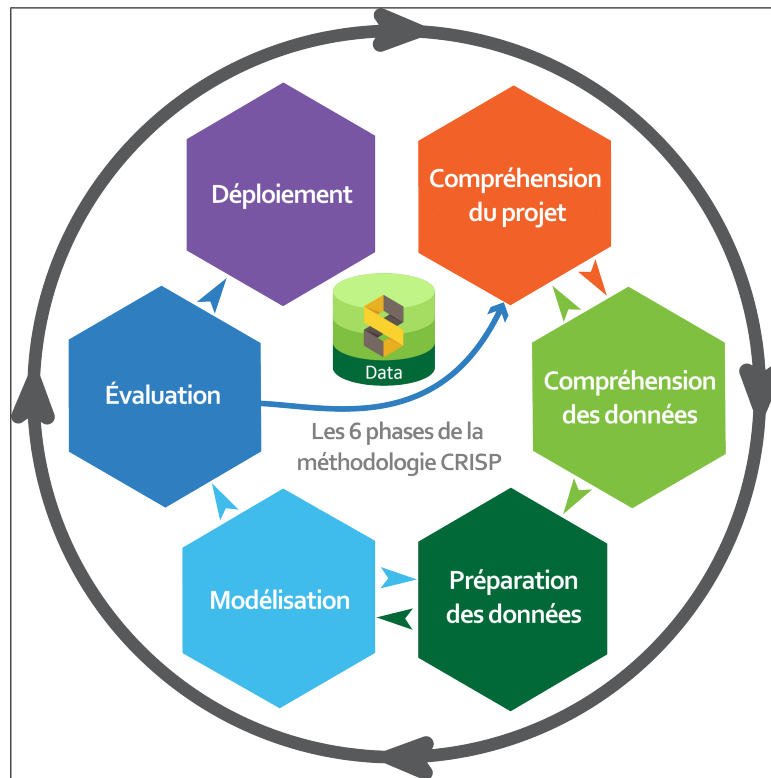


FIGURE 1.9 : Modèle CRISP-DM

Conclusion

Dans ce chapitre nous avons présenté le contexte général de notre projet et les motivations qui nous ont poussé à élaborer ce projet et à s'intéresser à ce sujet innovant et futuriste. Nous avons également discuté la solution existante et ses limites conduisant à penser à notre solution proposée. De plus, nous avons expliqué quelques notions théoriques de base qui nous seront utiles tout au long du projet. Et finalement, nous avons présenté la méthodologie de travail que nous choisissons, qui est la méthode CRISP-DM.

COLLECTE ET PRÉPARATION DES DONNÉES

Plan

1	Compréhension des données	14
2	Collecte des données	14
3	Préparation des données	16

Introduction

La première étape technique du lancement d'un projet de Data Mining est la collecte et la préparation de données. Dans ce chapitre, nous présentons les données nécessaires à la réalisation de ce projet, les sources de ces données et les opérations effectuées lors de la phase de préparation des données et l'extraction des paramètres (features extraction).

2.1 Compréhension des données

L'objectif de notre projet est la détection des crimes à partir de la vidéosurveillance, alors le jeu de données nécessaire doit contenir des vidéos de surveillance étiquetées avec des vidéos normales et d'autres contenant des anomalies.

2.2 Collecte des données

Tout d'abord, notre dataset initial est UCF-Crime Dataset qui est un jeu de données à grande échelle comprenant 128 heures de vidéo. Il s'agit de longues vidéos de surveillance non tronquées qui couvrent 13 anomalies du monde réel, notamment des abus, des arrestations, des incendies criminels, des agressions, des accidents de la route, des cambriolages, des explosions, des bagarres, des vols, des fusillades, des vols à l'étalage et du vandalisme. Ces anomalies ont été sélectionnées car elles ont un impact significatif sur la sécurité publique.

- **Abus** : Cet événement contient des vidéos montrant un comportement mauvais, cruel ou violent envers des enfants, des personnes âgées, des animaux et des femmes.
- **Arrestation** : Cet événement contient des vidéos montrant la police arrêtant des individus.
- **Incendie criminel** : Cet événement contient des vidéos montrant des personnes mettant délibérément le feu à des biens.
- **Agression** : Cet événement contient des vidéos montrant une attaque physique soudaine ou violente contre quelqu'un. Notez que dans ces vidéos, la personne agressée ne se défend pas.
- **Cambriolage** : Cet événement contient des vidéos montrant des personnes (voleurs) pénétrant dans un bâtiment ou une maison avec l'intention de commettre un vol. Il ne comprend pas l'usage de la force contre des personnes.
- **Explosion** : Cet événement contient des vidéos montrant des événements destructeurs où quelque chose explose. Cet événement ne comprend pas les vidéos dans lesquelles une personne met le feu ou déclenche une explosion de manière intentionnelle.

- **Bagarre** : Cet événement contient des vidéos montrant deux personnes ou plus en train de s'attaquer l'une à l'autre.
- **Accident de la route** : Cet événement contient des vidéos montrant des accidents de la route impliquant des véhicules, des marcheurs ou des cyclistes.
- **Vol à main armée** : Cet événement contient des vidéos montrant des voleurs prenant illégalement de l'argent par la force ou la menace de la force. Ces vidéos n'incluent pas les fusillades.
- **Fusillade** : Cet événement contient des vidéos montrant l'acte de tirer sur quelqu'un avec une arme à feu.
- **Vol à l'étalage** : Cet événement contient des vidéos montrant des personnes qui volent des marchandises dans un magasin en se faisant passer pour un client.
- **Vol** : Cet événement contient des vidéos montrant des personnes prenant des biens ou de l'argent sans autorisation. Elles n'incluent pas le vol à l'étalage.
- **Vandalisme** : Cet événement contient des vidéos montrant des actions impliquant la destruction ou l'endommagement délibéré de biens publics ou privés. Ce terme inclut les dommages matériels, tels que les graffitis et les dégradations dirigés vers un bien sans l'autorisation du propriétaire.
- **Événement normal** : Cet événement contient des vidéos où aucun crime n'a été commis. Ces vidéos comprennent des scènes d'intérieur (comme un centre commercial) et d'extérieur, ainsi que des scènes de jour et de nuit.

Ce dataset contient 1900 vidéos dont 950 sont des vidéos normales et 950 sont des vidéos anormales distribuées comme indiqué par les graphes ci-dessous.

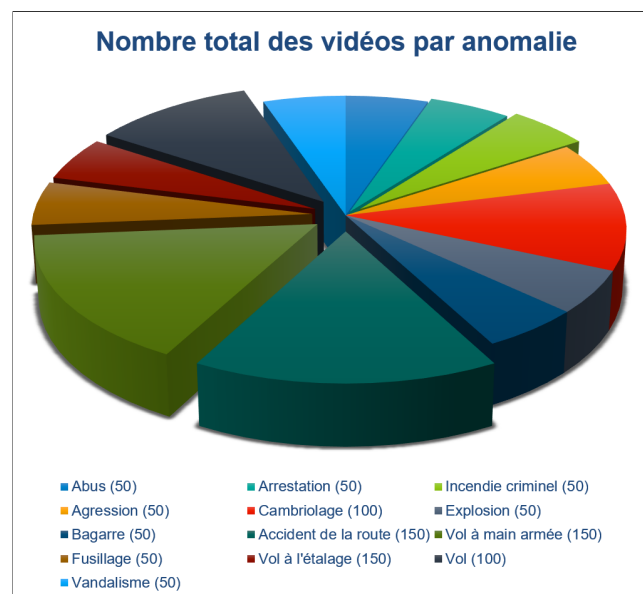


FIGURE 2.1 : Nombre total des vidéos par anomalie

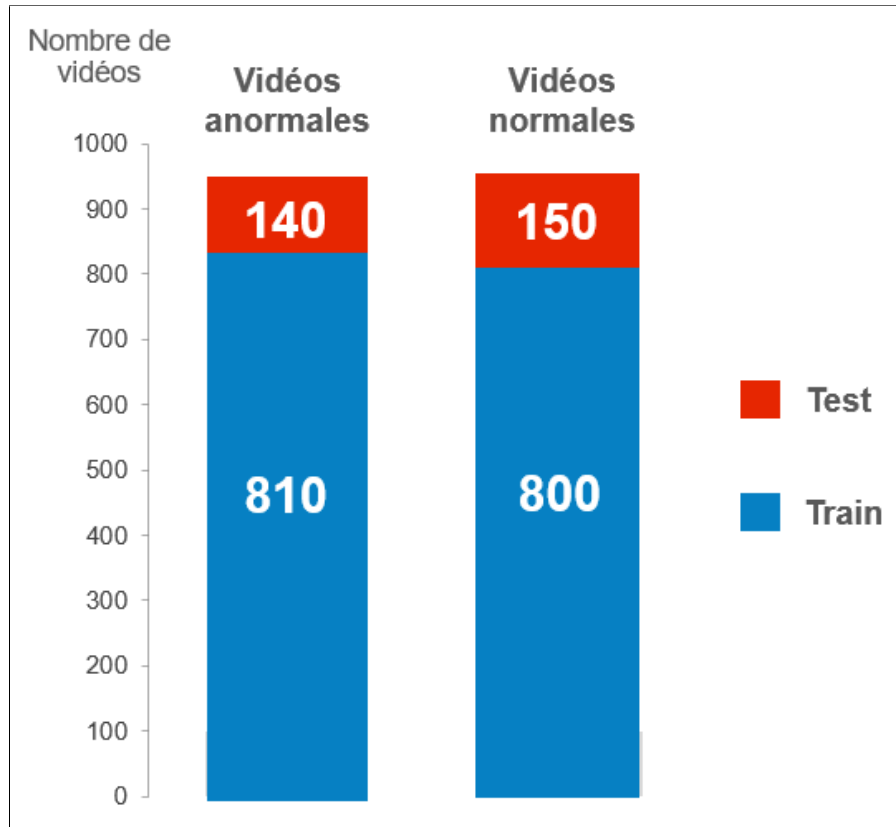
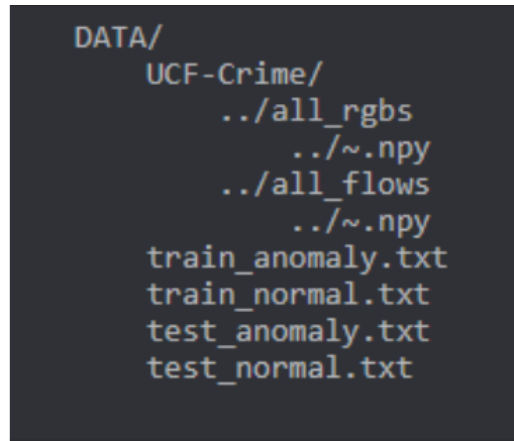


FIGURE 2.2 : Distribution des vidéos normales et anormales entre training et testing sets

2.3 Préparation des données

Les événements anormaux du monde réel sont complexes et variés, il est alors difficile d'énumérer tous les événements anormaux possibles. Par conséquent, il est souhaitable que la détection d'anomalies se fait avec un minimum de supervision. Nous visons traiter la détection générale d'anomalies en considérant toutes les anomalies dans un groupe et toutes les activités normales dans un autre groupe. Pour formuler une approche d'apprentissage faiblement supervisée, nous utilisons l'apprentissage par instances multiples (MIL). Plus précisément, nous proposons d'apprendre les anomalies par le biais d'un cadre MIL profond en traitant les vidéos de surveillance normales et anormales comme des sacs (sacs positifs pour les vidéos anormales et sacs négatifs pour les vidéos normales) et les courts segments de chaque vidéo comme des instances dans un sac (chaque vidéo est divisée en 32 segments temporels). Pour l'extraction des paramètres, on utilise I3D two-stream (RGBs FLOWs). Vu que la taille du dataset initial est très grande (> 100 Go), nous avons procédé à la recherche des paramètres pré-traités au lieu de télécharger le dataset complet et effectuer le processus d'extraction des paramètres, nous utilisons alors des dossiers contenant des fichiers (.npy) composés par des vecteurs de type numpy.

La figure ci-dessous représente l'arborescence des données :

A screenshot of a terminal window showing a directory tree structure. The root is 'DATA/'. Inside 'DATA/' is a subdirectory 'UCF-Crime/'. Inside 'UCF-Crime/' are two subdirectories: 'all_rgbs' and 'all_flows'. Each of these subdirectories contains a file named '~.npz'. Below the 'UCF-Crime/' directory, there are four files: 'train_anomaly.txt', 'train_normal.txt', 'test_anomaly.txt', and 'test_normal.txt'.

```
DATA/  
  UCF-Crime/  
    ../all_rgbs  
      ../~.npz  
    ../all_flows  
      ../~.npz  
  train_anomaly.txt  
  train_normal.txt  
  test_anomaly.txt  
  test_normal.txt
```

FIGURE 2.3 : Arborescence des données

Conclusion

Dans ce chapitre, nous avons combiné la deuxième et la troisième phase de la méthode CRISP-DM. À ce stade, nous sommes prêts à commencer la phase de modélisation du projet.

MODÉLISATION

Plan

1	Critères de réussite du modèle	19
2	Langages et outils utilisés	19
3	Architecture	22
4	Description	22

Introduction

Ce chapitre est consacré à la phase de construction du modèle adopté. Plusieurs modèles ont été testés afin de sélectionner le meilleur modèle satisfaisant nos critères de réussite.

3.1 Critères de réussite du modèle

Lorsque nous obtenons des bonnes performances sur le jeu de données qui a permis l'apprentissage, cela ne garantit pas de bons résultats sur de nouvelles données. On cherche donc à modéliser une analyse qui reflète la complexité de la nature des données en évitant les écueils du sous-apprentissage et du sur-apprentissage.

- **Sous-apprentissage (underfitting)** : Un phénomène se traduisant par le fait que le modèle n'apprend pas assez et réalise de mauvaise prédiction sur le jeu d'entraînement. Il n'arrive pas alors à déduire des informations du jeu de données et à capter la relation entre les données d'entrées et leurs étiquettes.
- **Sur-apprentissage (overfitting)** : Un phénomène se traduisant par le fait que le modèle a trop appris lors de l'entraînement qu'il devient trop collé aux données d'apprentissage et ne se généralise pas à de nouvelles données qui lui sont inconnues.

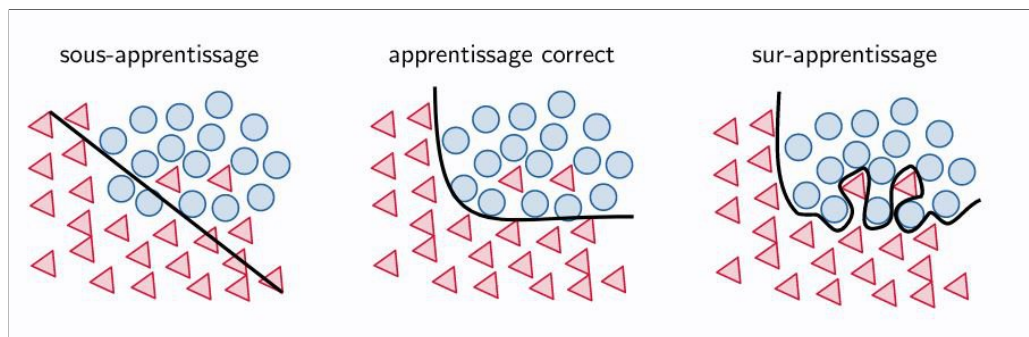


FIGURE 3.1 : Sous-apprentissage, apprentissage correct et sur-apprentissage

Il est également important de définir un modèle qui ait un temps de calcul convenable et une utilisation des ressources en mémoire raisonnable pour que l'algorithme d'analyse soit utilisable.

3.2 Langages et outils utilisés

3.2.1 Langages

- **Python** : C'est un langage de programmation de haut niveau interprété, orienté objet et à sémantique dynamique, développé par Guido van Rossum. Il a été initialement lancé en 1991.



FIGURE 3.2 : Logo de *Python*

3.2.2 Frameworks

- ***PyTorch*** : C'est une bibliothèque d'IA, développée par Meta, écrite en Python pour se lancer dans le deep learning (ou apprentissage profond) et le développement de réseaux de neurones artificiels. À partir de plusieurs variables, elle peut servir à réaliser des calculs de gradients ou à utiliser des tableaux multidimensionnels obtenus grâce à des tenseurs



FIGURE 3.3 : Logo de *PyTorch*

3.2.3 Éditeurs

- ***Pycharm*** : C'est un Environnement de développement intégré (IDE) Python développé et édité par JetBrains basé sur la plateforme IntelliJ.



FIGURE 3.4 : Logo de *Pycharm*

- ***Google Colaboratory*** : C'est un outil de recherche pour l'enseignement et la recherche en apprentissage automatique. Il permet d'écrire et d'exécuter du code Python dans le navigateur sans aucune configuration requise, avec un accès gratuit aux GPU et un partage facile.



FIGURE 3.5 : Logo de *Google Colaboratory*

3.2.4 Contrôle de version

- **Git** : Un outil de versioning et de développement collaboratif (ou *V.C.S. : Version Control System*). Il est devenu le standard du domaine grâce à son fonctionnement décentralisé, la facilité de gestion des branches et l'automatisation grâce à une multitude d'outils comme github.com, gitlab.com, git flow...



FIGURE 3.6 : Logo de *Git*

- **GitHub** : C'est un service d'hébergement Open-Source, permettant aux programmeurs et aux développeurs de partager le code informatique de leurs projets afin de travailler dessus de façon collaborative. On peut le considérer comme un Cloud dédié au code informatique.

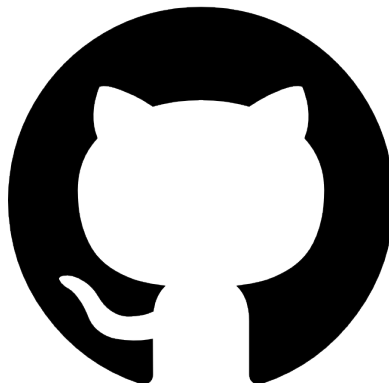


FIGURE 3.7 : Logo de *GitHub*

3.3 Architecture

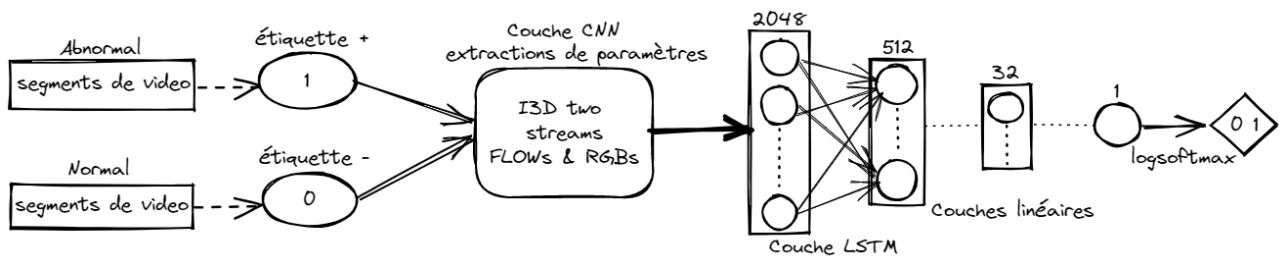


FIGURE 3.8 : Architecture du modèle de detection d'anomalies adopté dans notre solution proposé

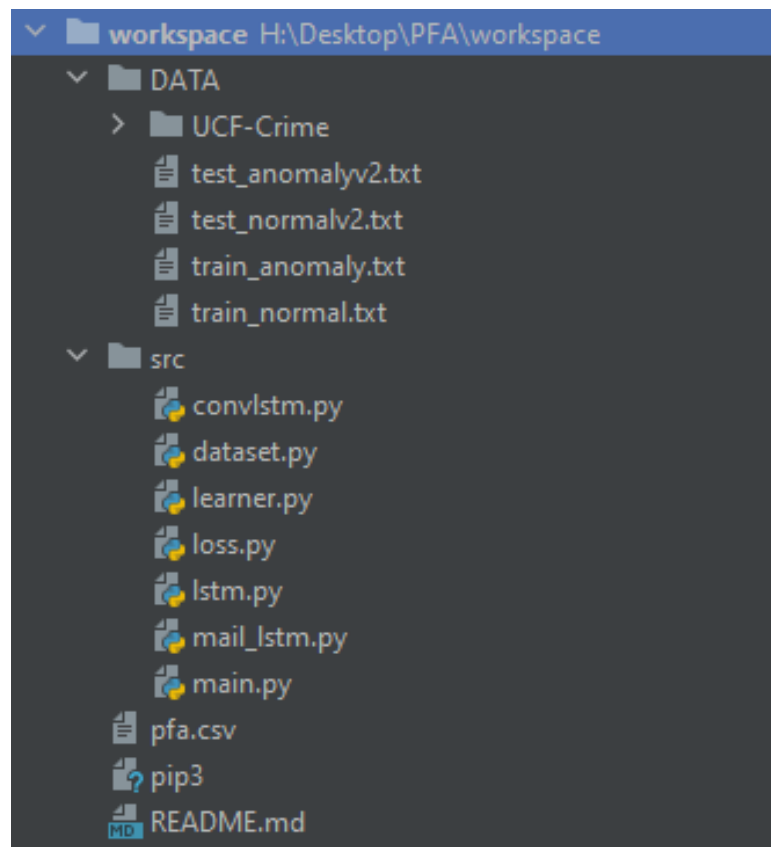


FIGURE 3.9 : Arborescence du code

3.4 Description

L'approche proposée par notre solution commence par diviser chacune des vidéos de surveillance en 32 segments temporels . Ces segments sont les instances des sacs positifs (contenant l'anomalie quelque part) et négatifs (ne contenant aucune anomalie). L'extraction des paramètres est effectuée par un modèle I3D two-stream qui est un modèle CNN prenant comme entrée les sacs de segments temporels et donnant comme sortie deux dossiers RGBs et FLOWS contenant chacun des fichiers de type numpy (.npy) Après avoir extraire les paramètres I3D two-stream, le modèle est entraîné à détecter des anomalies en se basant sur un réseau neuronal composé d'une couche LSTM de taille 2048,

les sorties de cette couche constituent les entrées des deux couches linéaires suivantes, pour finaliser par une fonction d'activation `logSoftmax` nous fournissant la sortie finale.

Conclusion

Dans ce chapitre, nous avons couvert la partie modélisation en parlant davantage des langages et des outils utilisés. Nous avons également donné une description détaillée de l'architecture adoptée dans notre solution proposée.

ÉVALUATION

Plan

1	Métrique	25
2	Tableau comparatif	28
3	Interprétation	29

Introduction

Après l'entraînement de notre modèle, il faut effectuer une évaluation globale en testant sur des données autres que celles d'entraînement, et puis construire la matrice de confusion de notre modèle dans le but de calculer quelques métriques. Afin d'évaluer avec succès un modèle d'apprentissage profond ou automatique, il est obligatoire de comparer les résultats générés par ce modèle avec les performances d'autres modèles possibles, cela nous permet de mieux interpréter les points forts et faibles de notre modèle.

4.1 Métrique

La tâche la plus importante dans la construction d'un modèle d'apprentissage profond et automatique est d'évaluer ses performances. Pour effectuer cette évaluation, nous recourons à des nombreuses métriques telles que chacune nous donne des renseignements spécifiques sur la classification. Un des outils utilisés pour le calcul de ces métriques est la matrice de confusion. Il s'agit d'un tableau contenant des combinaisons de valeurs prédites et réelles.

		<i>Reality</i>	
		Negative : 0	Positive : 1
<i>Prediction</i>	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

FIGURE 4.1 : Matrice de confusion

Cette matrice est composée de :

- **TP (True Positives)** : Les cas où la prédiction est positive et la valeur réelle est effectivement positive.
- **TN (True Negatives)** : Les cas où la prédiction est positive et la valeur réelle est effectivement négative.
- **La préparation des données** : Les cas où la prédiction est positive et la valeur réelle est effectivement positive.
- **FP (False Positive)** : Les cas où la prédiction est positive mais la valeur réelle est négative.
- **FN (False Negative)** : Les cas où la prédiction est négative mais la valeur réelle est positive.

Nous citons à titre d'exemples des métriques d'évaluation :

- **Perte (loss)** : C'est la mesure de la perte lors de la phase d'entraînement du modèle. La fonction

de perte est calculée à chaque itération (epoch) de l'entraînement et utilisée pour la mise à jour des poids.

- **Accuracy** : Elle mesure le nombre d'observations, tant positives que négatives, qui ont été correctement classées. Elle est calculée comme le rapport entre le nombre de prédictions correctes (TP+TN) et le nombre total de prédictions (TP+TN+FP+FN).
- **Précision** : Elle nous indique combien de cas correctement prédits se sont avérés positifs. Elle est calculée comme le rapport entre le nombre des observations positives correctement classées (TP) et le nombre total des prédictions positives (TP+FP). Cette métrique est importante pour les systèmes de recommandation où des résultats erronés peuvent conduire à la perte des clients.
- **Rappel (Recall)** : Il nous explique combien de cas positifs le modèle a pu prédire correctement. Il est calculé comme le rapport entre le nombre des observations positives correctement classées (TP) et le nombre total des observations positives (TP+FN). Cette métrique est importante dans les cas où il importe peu que nous déclenchions une fausse alerte, mais où les cas réels positifs ne doivent pas passer inaperçus.
- **F-score** : Il donne une idée combinée des métriques Précision et Rappel. Il est maximal lorsque la précision est égale au rappel.
- **Receiver Operator Characteristic ROC** : C'est une courbe de probabilité qui représente le TPR (taux de vrais positifs) par rapport au FPR (taux de faux positifs)
- **L'aire sous la courbe AUC** : C'est la mesure de la capacité d'un classificateur à distinguer les classes. Il s'agit de aire sous la courbe ROC

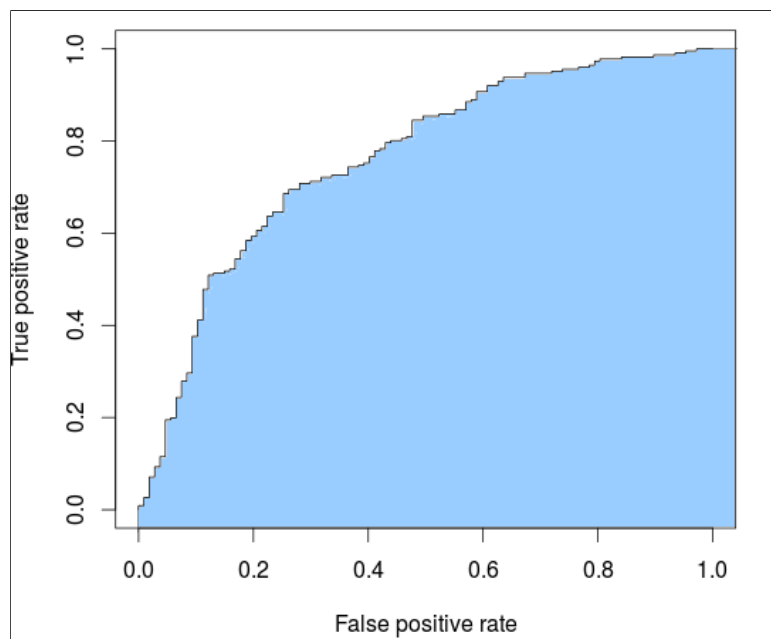


FIGURE 4.2 : Courbe ROC et Aire sous la courbe AUC

Les métriques telles que l'accuracy, la précision et le rappel sont de bons moyens pour évaluer les modèles de classification pour les ensembles de données équilibrés, mais si les données sont déséquilibrées, d'autres méthodes telles que ROC/AUC permettent de mieux évaluer les performances du modèle.

En utilisant différentes métriques pour l'évaluation des performances, nous devrions être en mesure d'améliorer le pouvoir prédictif global de notre modèle avant de le déployer en production sur des données non vues.

Dans le cas de notre projet, nous visons à réduire le nombre de fausses alarmes, c.à.d, le modèle prédit qu'il existe un crime mais en réalité il s'agit d'une vidéo normale, cela revient à maximiser la précision.

Mais le plus important est qu'aucun cas réel positif ne passe inaperçu, c.à.d, les vidéos présentant des crimes ne doivent pas être interprétées comme des vidéos normales, cela revient à maximiser le rappel.

Remarque : Notre modèle effectue le test à chaque epoch de l'entraînement, cela nous permet de calculer la précision et le rappel pour chaque epoch.

Nous présentons par les figures ci-dessous quelques résultats obtenus pour notre modèle :

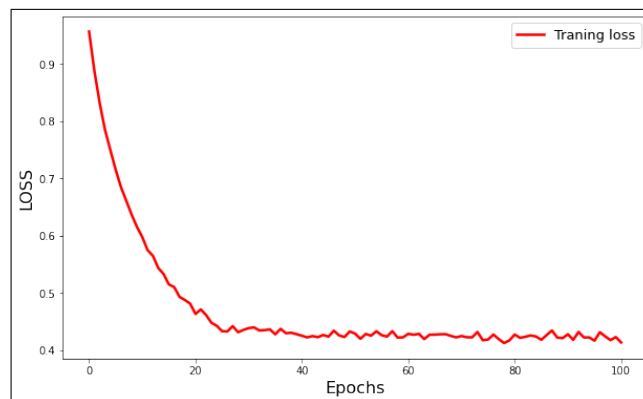


FIGURE 4.3 : Fonction de perte calculée à chaque epoch lors de l'entraînement du modèle

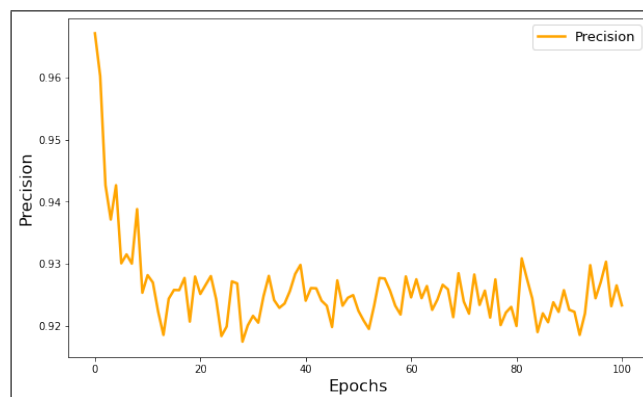


FIGURE 4.4 : Précision calculée lors du test à chaque epoch

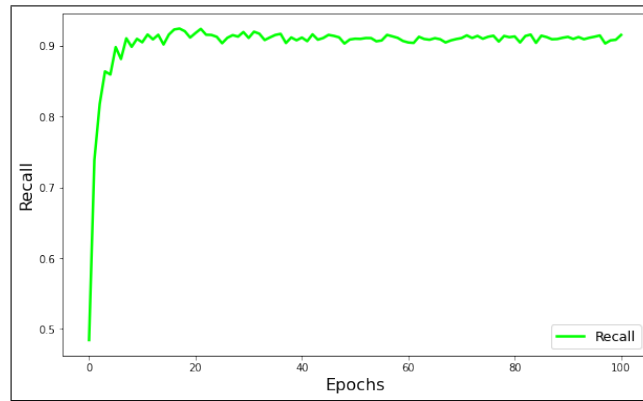


FIGURE 4.5 : Rappel calculé lors du test à chaque epoch

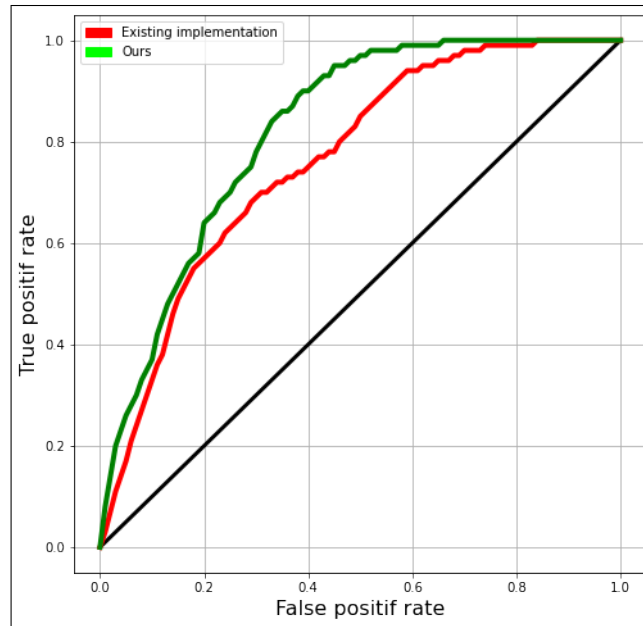


FIGURE 4.6 : Courbe ROC et Aire sous la courbe AUC de notre modèle en comparaison avec la solution existante

Pour une évaluation globale du modèle nous recourons à la métrique F-score qui donne la valeur $\text{F-score} = 0.916784$

4.2 Tableau comparatif

Méthode de l'extraction de paramètres		Type de réseau neuronal		
		<i>DNN</i>	<i>ConvLSTM</i>	<i>LSTM</i>
		(Solution existante)	(Notre essai)	(Notre solution adoptée)
	<i>C3D</i>	75.05		
	<i>I3D</i>	84.03		
	<i>I3D two-stream</i>	84.45	85.30	88.16

FIGURE 4.7 : Tableau comparatif des résultats obtenus

4.3 Interprétation

Nous avons testé deux architectures de réseau neuronal :

- **Une architecture CNN²-RNN : I3D + ConvLSTM :**

Cette architecture a donné un AUC = 85.30

- **Une architecture CNN-RNN : I3D two-stream + LSTM**

Cette architecture a donné un AUC = 88.16

L'architecture CNN-RNN présente un meilleur résultat que l'architecture existante DNN et l'architecture CNN²-LSTM. Ceci est justifié par le fait que non seulement une combinaison de deux couches CNN (aspect spatial) avec une couche RNN (aspect temporel) conduit à la dominance de l'aspect spatial au détriment de l'aspect temporel, mais aussi la résolution des vidéos est très faible (144-240 px).

Conclusion

Dans ce chapitre, nous avons évalué le degré d'adéquation de notre modèle aux objectifs visés, et déterminé la raison des déficiences rencontrées. À ce stade, nous sommes prêts à entamer la phase finale qui est la phase de production de notre solution.

PRODUCTION

Plan

1	Conception IoT	31
2	Description	32

Introduction

Dans ce chapitre nous dévoilons l'idée générale de la méthode de production imaginée de notre projet ainsi que les technologies et outils à utiliser. Cette phase de la méthodologie CRISP-DM est très importante car elle nous permet de tester l'efficacité de notre solution sur le plan réel en tant qu'un produit prêt à utiliser.

5.1 Conception IoT

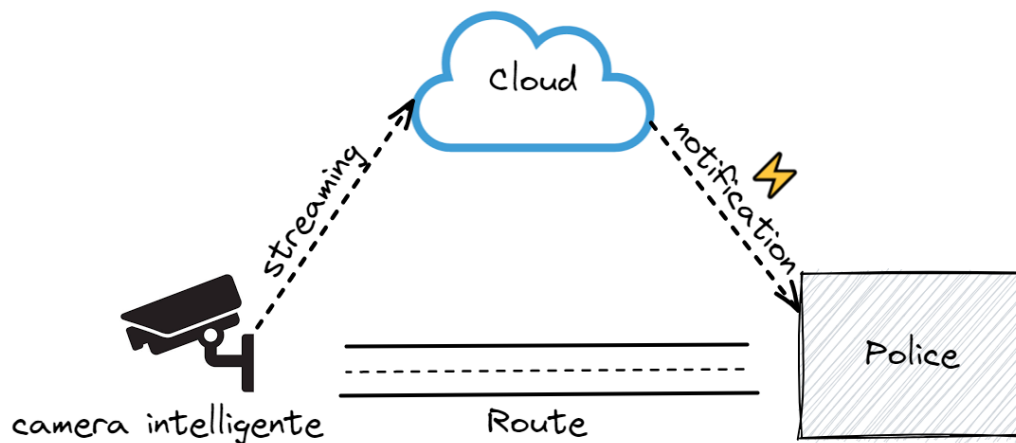


FIGURE 5.1 : Conception IOT de la méthode de production du projet

Les éléments utilisés dans cette conception IOT sont les suivants :

- **Camera intelligente** : Une caméra intelligente est comme son nom l'indique une caméra dans laquelle est ajoutée une électronique permettant d'acquérir et de stocker des images mais aussi de traiter de l'information et de communiquer avec les systèmes environnants (réseau, cloud, automates, opérateurs ...).



FIGURE 5.2 : Caméra intelligente

- **Cloud** : Le cloud computing ou informatique en nuage est une infrastructure dans laquelle les puissances de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée.



FIGURE 5.3 : Les services du Cloud

On peut utiliser par exemple le **AWS Cloud Provider**



FIGURE 5.4 : Logo de *AWS*

5.2 Description

Une caméra intelligence capte des scènes en temps réel et les envoie au service cloud par le recours à des protocoles de connexion IOT. Les unités de calcul au niveau du cloud procèdent à la detection d'anomalies dans ces scènes, si un crime est detecté, localisent ce crime et déclenchent une alerte au poste de police le plus proche.

Conclusion

Dans ce dernier chapitre nous avons abouti au résultat final de la conception de notre projet. La concrétisation de cette étape résulte en un produit de détection des différentes activités criminelles à partir des vidéos de surveillance. Ce produit est finalement prêt à être utilisé par la police et aux organismes autoritaires.

Conclusion générale et perspectives

Nous aimerons tout d'abord souligner que nous avons eu la chance de travailler sur un sujet très intéressant qui nous a permis d'avoir un aperçu sur l'importance de la détection des anomalies comme un problème classique et commun à de nombreux domaines d'activité.

L'objectif de notre projet était de tirer parti de la vidéosurveillance pour concevoir un modèle de prédiction des crimes permettant d'introduire la notion de police prédictive qui servira évidemment à améliorer le processus de résolution des crimes.

Dans ce rapport, nous avons exploré comment un ensemble d'approches d'apprentissage profond et automatique peut être appliqué à cette tâche dans un cadre faiblement supervisé.

Malheureusement, la tentative de reconnaissance de l'anomalie détectée a échoué vu que les vidéos sont de longues vidéos de surveillance non découpées avec de très grandes variations intra-classes.

Ainsi, nous proposons comme perspectives d'amélioration de notre projet un pré-traitement plus profond des vidéos initiales pour adapter le jeu de données au processus de classification des différents types d'anomalies.

Références

- *Real-world Anomaly Detection in Surveillance Videos*

<https://www.crcv.ucf.edu/research/real-world-anomaly-detection-in-surveillance-videos/>

- *Anomaly Detection In Surveillance Videos on UCF-Crime*

<https://paperswithcode.com/sota/anomaly-detection-in-surveillance-videos-on>

- *Crime forecasting : a machine learning and computer vision approach to crime prediction and prevention*

<https://vciba.springeropen.com/articles/10.1186/s42492-021-00075-z>

- *A novel keyframe extraction method for video classification using deep neural networks*

<https://link.springer.com/content/pdf/10.1007/s00521-021-06322-x.pdf>

- *Deep Learning for Anomaly Detection*

<https://ff12.fastforwardlabs.com/>