



الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي  
جامعة وهران للعلوم والتكنولوجيا محمد بوضياف  
كلية الرياضيات و الاعلام الالي

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur Et de la Recherche Scientifique  
Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF  
Faculté des Mathématiques et Informatique

*Département : Informatique*

Mémoire de fin d'études

# Video based human action recognition using deep learning

Pour l'obtention du diplôme  
de **Master**

Domaine : Mathématiques – Informatique

Filière : Informatique

Spécialité : Intelligence artificielle et es applications

Présenté le :

Par :

- Brahmia Mohammed Yacine
- Rouabah Hadj Habib

Jury	Nom et Prénom	Grade	Université
Président:	Mr. TLEMSANI Redouane	MCA	USTO
Encadrant:	Mr. DAIRI Abdelkader	MCA	USTO
Examineur:	Mr. NEMMICHE Amine	MCB	USTO

2022/2023

## *REMERCIEMENTS*

En premier lieu, nous rendons grâce à Dieu, le Tout-Puissant, à qui nous exprimons notre gratitude pour nous avoir donné le courage, la patience et la volonté de réaliser ce travail.

Nous tenons à remercier sincèrement Dr DAIRI Abdelkader pour son encadrement, son soutien et ses conseils précieux tout au long de notre recherche.

Nos remerciements vont également aux membres du jury Dr. TLEMSANI Redouane et Dr. NEMMICHE Amine pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail et de l'enrichir de leurs propositions.

Évidemment, nous n'oublions pas de remercier nos familles, nos amis, nos collègues et tous ceux qui, avec leurs conseils et leurs encouragements, nous ont aidé à mener ce travail à terme.

## *Dédicaces*

A nos chers parents, que nulle dédicace ne puisse exprimer ce que nous leurs devons, pour leur bienveillance, leur affection et leur soutien ... Trésors de bonté, de générosité et de tendresse, en témoignage de notre profond amour.

A tous nos amis et nos enseignants, qui ont contribué de loin ou de près à l'élaboration de ce travail.

# Table des matières

<b>Abstract .....</b>	<b>14</b>
<b>Résumé .....</b>	<b>15</b>
<b>CHAPITRE 1: ETAT DE L'ART .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Problématique.....	1
1.3 Limites de la surveillance vidéo humaine traditionnelle .....	1
1.4 Apport de l'intelligence artificielle dans la surveillance du comportement humain .....	2
1.5 Types de comportements à monitorer.....	3
1.6 Vision par ordinateur et domaine médicale.....	3
1.7 Intelligence artificielle.....	4
1.7.1 C'est quoi une intelligence artificielle ?.....	4
1.7.2.1 L'apprentissage supervisé .....	5
Algorithmes de classification .....	5
Algorithmes de régression .....	6
1.7.2.2 L'apprentissage non-supervisé.....	6
Clustering.....	7
1.7.2.3 L'apprentissage Par Renforcement .....	7
1.7.3 Apprentissage Profond .....	8
1.7.4 Vision par Ordinateur .....	9
1.6.6 Applications divers de la vision par ordinateur.....	10

1.7 Conclusion.....	11
<b>CHAPITRE 2: Détection de chutes basée sur la vision par ordinateur .....</b>	<b>12</b>
2.1 Introduction .....	12
2.2 Reconnaissance des activités humaines (HAR) .....	12
2.3 Détection d'actions Humaine.....	13
2.3.1 Définition.....	13
2.3.2 Domaines d'applications .....	14
2.3 Détection de chutes (FALL Detection) .....	15
2.3.2 Revue de littérature et travaux au sujet de la détection de chutes.....	17
2.4 Axes relié à la détection de la chute .....	20
2.4.1 Détection par Gait (la démarche) .....	20
2.4.2 Analyse Comportementale .....	21
2.4.3 Les détections d'actions humaines basées sur le squelette et sur l'estimation de la pose.....	22
2.4.4 Détection d'anomalies .....	23
2.5 Conclusion.....	24
<b>CHAPITRE 3: Implémentation .....</b>	<b>25</b>
3.1 Introduction .....	25
3.2 Environnement d'exécution .....	25
3.3 Modèles d'apprentissage automatique utilisés.....	27
3.3.1 YOLO .....	27
C'est quoi YOLO ?.....	27

Architecture du modèle YOLO .....	27
Fonctionnement du modèle YOLO .....	27
Chronologie du modèle YOLO .....	29
3.3.2 Les Classificateurs Supervisés utilisés .....	31
KNN .....	32
Gradient Boosting classifier.....	32
eXtreme Gradient Boosting .....	32
Histogram Gradient Boosting .....	32
ADA Boost.....	33
Random Forest Decision Tree .....	33
3.3.3 Classificateur Non-Supervisé utilisés .....	33
K-means .....	34
BIRCH.....	34
Agglomerative Clustering .....	34
3.4 METHODOLOGIE .....	35
3.4.1 Collection des données (choix de l'ensemble de données) .....	36
C'est quoi un ensemble de données (dataset) ? .....	36
Pourquoi utiliser un dataset ?.....	36
CAUCAFall .....	37
Structure du dataset CAUCAFall .....	37
3.4.2 Data Processing.....	39

Frame Extraction .....	39
Détecter la personne en utilisant YOLO .....	39
Limitations lors de l'utilisation de YOLO .....	41
Contexte Temporel .....	41
Certain frames ont été ratés.....	41
Chevauchement/Occlusion des Objets .....	42
Conditions de qualité d'image, d'éclairage et de colorisation.....	42
Application de notre approche .....	42
Application du Window Sliding .....	45
3.4.3 Préparation des données pour l'entraînement .....	45
Le ratio r .....	45
Détermination de la valeur de Y .....	47
3.4.4 Reconnaissance d'actions Supervisé et Non Supervisé .....	48
3.5 Résultats et Discussion .....	48
3.5.1 Métriques d'évaluation .....	48
3.5.2 Résultats des classificateurs supervisés .....	49
Avec Shuffle=vrai (true) .....	49
Avec Shuffle=faux (false) .....	52
3.5.3 Résultats des classificateurs non-supervisés .....	55
Avec Shuffle=vrai (true) .....	55
Avec Shuffle=faux (false) .....	56

3.6 Perspectives pour le future .....	58
3.7 Conclusion.....	59
<b>Conclusion Générale</b> .....	60
<b>Bibliographies</b> .....	61



## **Mots clés**

Activity recognition, Action recognition, datasets, Deep learning, YOLO, Computer vision, FALL Detection.

# Listes des figures

Figure 1: Composantes de L'apprentissage automatique [4] .....	5
Figure 2: Classification et Régression [6] .....	6
Figure 3: Représentation graphique de la manière dont le clustering peut être utilisé pour identifier des modèles dans un ensemble de données [9]. .....	7
Figure 4: Structure d'un réseau de neurones [11] .....	8
Figure 5: Sous-Activités de la Human Activity Recognition (HAR) [18], .....	13
Figure 6: Taux d'intérêt pour la détection des chutes, d'avril 2004 à décembre 2021. Les données expérimentales sont obtenues à partir de "Google Trends" en utilisant "détection des chutes" comme objet de recherche. Les résultats sont normalisés en utilisant les taux les plus élevés ; ainsi, le taux d'intérêt le plus élevé a une valeur de 100 .....	15
Figure 7: Taxonomie des méthodes de détection de chutes [20]. .....	17
Figure 8: Représentation d'un cycle complet de Gait .....	21
Figure 9: Exemples d'estimation de pose humaine sur des données réelles basée sur le squelette [36]. .....	23
Figure 10: Chronologie des versions de YOLO [45] .....	29
Figure 11: Sommaire des architectures de YOLO [49] .....	31
Figure 12: Organigramme de notre Approche .....	35
Figure 13: CAUCAFall Dataset Folder Structure [59] .....	38
Figure 14: Image (frame) avant application de YOLO .....	39

Figure 15: Image (frame) après application du modèle YOLO .....	40
Figure 16: Exemple d'instance où l'application des rotations sur le même frame a permis de détecter la personne.....	43
Figure 17: exemple de vecteur de caractéristiques (Features) .....	44
Figure 18: Personne en position Normale (NOFALL) .....	46
Figure 19: Personne en position Anormale (Accroupi) .....	46
Figure 20: Personne en position Anormale (Chute) .....	47

# Liste des Tableaux

Tableau 1: Résultats 1 KNN .....	49
Tableau 2: Résultats 1 GBC .....	49
Tableau 3: Résultats 1 XGBC .....	50
Tableau 4: Résultats 1 HGBC .....	50
Tableau 5: Résultats 1 RFC .....	50
Tableau 6: Résultats 1 DTC .....	51
Tableau 7: Résultats 1 ABC .....	51
Tableau 8 : Résultats 2 KNN .....	52
Tableau 9: Résultats 2 GBC .....	52
Tableau 10: Résultats 2 XGBC .....	53
Tableau 11: Résultats 2 HGBC .....	53
Tableau 12: Résultats 2 RFC .....	53
Tableau 13: Résultats 2 DTC .....	54
Tableau 14: Résultats 2 ABC .....	54
Tableau 15: Résultats 1 BIRCH .....	55
Tableau 16: Résultats 1 K-means.....	55
Tableau 17: Résultats 1 AggC.....	56

Tableau 18: Résultats 2 BIRCH .....	56
Tableau 19: Résultats 2 K-means.....	57
Tableau 20: Résultats 2 AggC.....	57
Tableau 21: Résumé des résultats .....	58

# Glossaire

**ADL:** activities and daily life

**HAR:** Human activity recognition.

**RF:** radio frequencies.

**FMCW:** Frequency-modulated continuous-wave.

**TP:** True Positive.

**FP:** False Positive.

**TN:** True Negative.

**FN:** False Negative.

**mAP:** mean average precisions.

**KNN:** K-Nearest Neighbors

**RGB:** Red Green Blue

**RGBD:** Red Green Blue Depth

**NMS :** Non-Max Suppression

**IOU :** Intersection sur les syndicats ou

# Abstract

Falls are one of the most critical health risks for the elderly, potentially being an indirect cause of death in certain circumstances. Additionally, demographic projections indicate a global increase in the elderly population. In this context, automated fall detection and prediction models are of paramount importance, particularly in AI applications that utilize ambient sensors or computer vision. In this thesis, we focus on human action recognition based on video recordings from surveillance cameras using computer vision techniques. More specifically, we concentrate on fall detection based on deep learning. In our approach, we first apply the YOLO model to detect individuals in each image of the CAUCAFALL dataset. We then extract a feature vector that we use to calculate the aspect ratio of the bounding boxes around individuals to annotate their state. In the final step, we employ various supervised and unsupervised classifiers to train our model to classify fall and non-fall events. Our training achieved an accuracy of 97% using the KNN classifier with neighboring values for other classifiers.

# Résumé

Les chutes sont l'un des risques les plus critiques pour la santé des personnes âgées, pouvant être, dans certaines circonstances, une cause indirecte de décès. De plus, les prévisions démographiques pour l'avenir montrent une augmentation de la population âgée dans le monde entier. Dans ce contexte, les modèles de détection et de prédiction automatiques des chutes revêtent une importance capitale, en particulier les applications d'intelligence artificielle qui utilisent des capteurs ambiants ou la vision par ordinateur. Dans ce mémoire, nous nous concentrons sur la reconnaissance des actions humaines à partir des enregistrements vidéo des caméras de surveillance basée sur la vision par ordinateur. Plus précisément, sur la détection des chutes basée sur l'utilisation de l'apprentissage profond. Dans notre approche, nous appliquerons d'abord le modèle YOLO pour détecter les personnes dans chaque image du dataset CAUCAFALL, ensuite nous extrayons un vecteur de caractéristiques, qu'on utilisera pour calculer le rapport entre la largeur et la longueur des boîtes englobantes des personnes, afin d'annoter l'état de la personne, dans la dernière étape, nous utilisons différents classificateurs supervisé et non-supervisé afin d'entraîner notre modèle à classifier les événements de chute et de non-chute, la précision de notre entraînement a atteint 97% en utilisant le classifieur KNN et des valeurs avoisinantes en utilisant les autres classificateurs.



# Introduction générale

La vidéo-surveillance (aussi appelé la surveillance à distance), est le procédé qui consiste à veiller sur des lieux, des biens ou des personnes à l'aide des caméras, cette technologie est largement utilisée dans divers domaines tels que la sécurité, la prévention des vols, la gestion du trafic et même dans le domaine médicale. Cependant, la surveillance humaine traditionnelle présente des limites en termes de disponibilité, de coût, d'efficacité et de précision. Les opérateurs humains peuvent être limités dans leur capacité à surveiller en permanence les flux vidéo et à détecter rapidement des incidents ou des comportements suspects. De plus, l'analyse manuelle des énormes quantités de données vidéo peut être couteuse, fastidieuse et sujette à des erreurs humaines.

Les solutions informatiques se sont avérées comme les meilleures solutions à élaborer face à ces limites, surtout avec l'avancement des techniques divers du de l'intelligence artificielle, de l'apprentissage profond et de la vision par ordinateur.

Ainsi, l'intégration de la vision par ordinateur dans la surveillance à distance permet l'ouverture de nouvelles perspectives afin d'améliorer l'efficacité et la précision de la vidéo-surveillance. En utilisant des algorithmes d'apprentissage automatique et de traitement d'image avancés, les systèmes de surveillance basés sur la vision par ordinateur peuvent analyser en temps réel les flux vidéo, détecter automatiquement des objets, des actions, des événements ou même des comportements spécifiques, et déclencher des alertes en cas d'incident ou de situation anormale.

Dans le domaine de la santé, l'intégration de la vision par ordinateur dans les systèmes de surveillance à distance offre des avantages considérables, Ces systèmes peuvent alerter les professionnels de la santé en cas de comportement anormal ou de situation d'urgence, ce qui permet une intervention rapide et une prise en charge adaptée.

Un des comportements anormaux à détecter est la chute des personnes, d'après les estimations de l'Organisation mondiale de la santé (OMS), environ 684 000 chutes mortelles et 37,3 millions de chutes nécessitant des soins médicaux surviennent chaque année dans le monde [1], les plus touchés étant les adultes de plus de 65 ans, selon la même source, Environ un tiers des personnes de ce groupe d'âge font une chute chaque année et 5 % de ces dernières entraînent

au moins une fracture [2]. Dans un tel contexte, il n'est pas surprenant que la conception de systèmes de détection des chutes rapides et automatiques ait suscité l'intérêt des chercheurs durant ces dernières années.

On parle de chute de personnes lorsqu'un individu réalisant une activité normale (marcher, manger, cuisiner,...) se retrouve accidentellement et soudainement au sol. Il peut avoir la force de se relever ou non. Dans le cas où il parvient à se relever, il parviendra lui-même à appeler les secours pour traiter les éventuels problèmes engendrés par sa chute. Mais dans le second cas, cette chute peut occasionner de nombreuses conséquences néfastes si l'individu ne peut pas appeler le service d'urgences ou un centre médical pour lui venir en aide à temps.

La gravité des conséquences de chute dépend en grande partie du temps mis pour aider la personne qui est tombée accidentellement, spécialement chez les personnes âgées. Ainsi plus la réponse à l'accident est rapide, moins lourdes seront les conséquences sur la vie du sujet.

Dans ce mémoire, nous nous intéressons plus précisément à l'utilisation de la vision par ordinateur dans la détection d'actions humaines. Nous explorerons les méthodes et les techniques basées sur l'apprentissage en profondeur (deep learning) pour la reconnaissance d'actions dans les vidéos plus précisément celles de la chute humaine. Nous examinerons également les avantages et les défis de l'intégration de cette technologie dans les systèmes de surveillance à distance dans le domaine médical.

# CHAPITRE 1: ETAT DE L'ART

## 1.1 Introduction

Dans ce premier chapitre, nous aborderons différents aspects de la surveillance du comportement humain et l'importance d'intégrer l'intelligence artificielle dans ce domaine, ensuite, nous introduirons les concepts clés de l'intelligence artificielle et de la vision par ordinateur, Finalement, nous soulignons diverses applications de la vision par ordinateur dans différents domaines. Cette introduction prépare le terrain pour la suite du mémoire, abordée dans le chapitre deux et trois.

## 1.2 Problématique

Avec le vieillissement de la population et l'augmentation des maladies chroniques, la surveillance du comportement des patients est devenue un enjeu majeur dans le domaine de la santé. L'objectif est d'assurer leur bien-être, de prévenir tout incident et événement non désiré. Dans ce contexte, la détection d'actions et des comportements joue un rôle crucial. Il s'agit d'identifier les actions spécifiques réalisées par les patients, telles que se lever, marcher, prendre des médicaments, ainsi que de détecter les comportements anormaux ou les situations d'urgence, comme les chutes. Cependant, la surveillance manuelle et continue de chaque patient est souvent difficile à réaliser pour le personnel médical, en raison des contraintes de temps et des ressources limitées. C'est là que le rôle des techniques de l'intelligence artificielle (IA) et de la vision par ordinateur devient crucial, donc la question qui se pose :

**Comment peut-on automatiser la surveillance des actions et du comportement des patients ?**

## 1.3 Limites de la surveillance vidéo humaine traditionnelle

La surveillance vidéo humaine traditionnelle présente plusieurs limites qui peuvent affecter son efficacité et son coût. Voici quelques exemples de limites :

- **Coût** : La surveillance vidéo humaine nécessite l'embauche de personnel pour surveiller les caméras en temps réel ou pour effectuer la vérification des enregistrements vidéo. Cela peut

être coûteux pour les entreprises ou les organisations qui cherchent à surveiller des zones surpeuplées, étendues ou des zones à risque élevé.

- Erreurs humaines : Les opérateurs de surveillance vidéo peuvent manquer des événements importants s'ils ne sont pas attentifs ou s'ils sont distraits. De plus, les opérateurs peuvent avoir des biais inconscients qui peuvent affecter leur interprétation des images vidéo.
- Disponibilité : Les opérateurs de surveillance vidéo ne peuvent pas être disponibles 24 heures sur 24 et 7 jours sur 7 pour surveiller les caméras. Ils ont besoin de pauses, de vacances et peuvent tomber malades. Cela peut créer des périodes de temps où la surveillance est faible ou absente.
- Précision : La surveillance vidéo humaine peut être affectée par des conditions environnementales telles que l'éclairage, la distance ou la qualité des images vidéo. Cela peut rendre difficile la reconnaissance des objets, ou personnes.
- Fatigue : Les opérateurs de surveillance vidéo peuvent être soumis à une fatigue mentale en raison du travail prolongé de surveillance. Cela peut affecter leur capacité à être attentifs et à répondre efficacement aux incidents.

## **1.4 Apport de l'intelligence artificielle dans la surveillance du comportement humain**

L'intelligence artificielle (IA) joue un rôle de plus en plus important dans la surveillance du comportement humain. Grâce aux avancées en vision par ordinateur et en apprentissage automatique, les systèmes de surveillance peuvent désormais analyser les flux vidéo en temps réel et détecter automatiquement des actions, des comportements ou des situations suspectes. L'IA permet d'améliorer l'efficacité de la surveillance en réduisant les erreurs humaines et en permettant une détection rapide des incidents, elle permet donc de remédier les imperfections de la détection humaine. Elle offre également la possibilité d'analyser de grandes quantités de données vidéo et de fournir des informations précieuses pour la sécurité et la prévention des incidents. L'intégration de l'IA dans la surveillance du comportement humain ouvre de nouvelles perspectives pour assurer la sécurité et la protection dans divers domaines tels que la sécurité publique, la gestion des infrastructures critiques et la surveillance médicale.

## 1.5 Types de comportements à monitorer

- **Chutes** : La détection des chutes est l'un des comportements les plus importants à surveiller, en particulier chez les personnes âgées ou les individus à risque de chutes fréquentes. La surveillance peut permettre de détecter rapidement une chute et de prendre des mesures d'urgence appropriées.
- **Mouvements anormaux** : Cela inclut les mouvements saccadés, les tremblements ou les convulsions, qui peuvent être des signes de problèmes de santé ou de crises épileptiques. La surveillance de ces comportements peut aider à alerter les soignants ou les professionnels de la santé en cas d'urgence.
- **Somnolence ou endormissement** : La surveillance peut détecter les signes de somnolence excessive ou d'endormissement, ce qui est particulièrement important dans les environnements de conduite ou de travail où l'attention est primordiale.
- **Comportement de désorientation** : La surveillance peut être utilisée pour détecter les comportements de désorientation ou de confusion, qui peuvent être des symptômes de certaines conditions médicales, telles que la démence ou la maladie d'Alzheimer.
- **Comportement d'automutilation** : Dans les cas où les individus peuvent se blesser délibérément, la surveillance peut être utilisée pour détecter les comportements d'automutilation et déclencher une intervention immédiate.

## 1.6 Vision par ordinateur et domaine médicale

L'application de la vision par ordinateur dans le domaine médical comporte de nombreux avantages. Elle permet par exemple de réaliser des analyses d'images médicales automatisées, telles que la détection de tumeurs, la segmentation d'organes ou la mesure de paramètres anatomiques. Ces analyses automatisées aident les médecins à prendre des décisions éclairées et précises, elle peut également être utilisée pour surveiller en temps réel les signes vitaux des patients, tels que la fréquence cardiaque, la respiration ou la saturation en oxygène. Cela permet une surveillance continue et non intrusive, ce qui est particulièrement utile pour les patients hospitalisés ou ceux qui nécessitent un suivi à domicile.

En outre, la vision par ordinateur peut être appliquée à la détection d'anomalies ou de comportements suspects dans les vidéos médicales, tels que la surveillance des mouvements des patients pour détecter les chutes, les comportements agités ou les signes de détresse.

---

Mais avant, qu'est-ce que cette fameuse vision par ordinateur ?

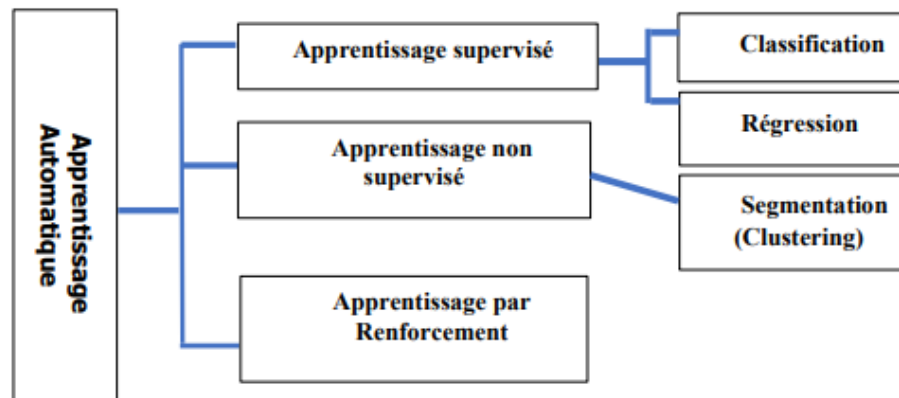
## **1.7 Intelligence artificielle**

### **1.7.1 C'est quoi une intelligence artificielle ?**

Une intelligence artificielle (IA) est une branche de l'informatique qui vise à développer des machines capables d'exécuter des tâches qui requièrent normalement l'intelligence humaine, telles que la reconnaissance vocale, la reconnaissance d'image, la prise de décision et l'apprentissage. L'objectif de l'IA est de créer des systèmes capables de traiter et d'analyser des données de manière autonome, d'apprendre à partir de ces données et de s'adapter pour effectuer des tâches de plus en plus complexes. L'IA peut être utilisée dans de nombreux domaines, tels que la santé, la finance, la sécurité et les transports.

### **1.7.2 Apprentissage Automatique (machine)**

L'apprentissage machine (en anglais Machine Learning) peut être défini comme étant une technologie d'intelligence artificielle permettant aux machines d'apprendre sans avoir été au préalable programmées spécifiquement à cet effet. Elle traite des sujets complexes où la programmation traditionnelle trouve ses limites. Construire un programme qui conduit une voiture par exemple serait très complexe, à cause des nombres infinis des cas possibles à traiter, donc au lieu de décrire quoi faire, le programme apprendra par lui-même comment conduire en "observant" des expérimentations et en prenant des décisions. L'apprentissage machine permet donc au modèle de faire des prédictions ou des décisions basées sur des modèles et des relations dans les données [3].



*Figure 1: Composantes de L'apprentissage automatique [4]*

On peut diviser l'apprentissage automatique en 3 catégories :

### 1.7.2.1 L'apprentissage supervisé

La machine s'appuie sur des classes prédéterminées et sur un certain nombre de paradigmes connus pour mettre en place un système de classement à partir de modèles déjà classés. Il y a deux étapes nécessaires pour compléter le processus, à commencer par l'apprentissage qui consiste à la modélisation des données cataloguées (étiquetées). Ensuite, il s'agira au second stade de se baser sur les données ainsi définies pour attribuer des classes aux nouveaux modèles introduits dans le système, afin de les cataloguer eux aussi.

Il y a deux types d'algorithmes dans l'apprentissage supervisé : les algorithmes de régression et les algorithmes de classification [5].

#### Algorithmes de classification

En apprentissage automatique, la classification est une forme d'apprentissage supervisé dans laquelle vous entraînez un modèle à utiliser les caractéristiques (les valeurs  $x$  dans notre fonction) pour prédire une étiquette ( $y$ ) qui calcule la probabilité que le cas observé appartienne à chacune d'un certain nombre de classes possibles et prédire une étiquette appropriée et l'attribuer à la classe, parmi les algorithmes de classification les plus utilisés, nous citons les arbres de décision, les  $k$  plus proches voisins ( $k$ -NN), les machines à vecteurs de support (SVM) et les réseaux de neurones [3].

## Algorithmes de régression

La régression est une méthode statistique utilisée pour étudier la relation entre une variable dépendante continue et une ou plusieurs variables indépendantes. Elle vise à estimer les coefficients qui décrivent cette relation en utilisant un modèle mathématique approprié. Le modèle de régression suppose généralement une relation linéaire entre les variables, bien que des formes non linéaires puissent également être explorées.

En général, l'objectif de la régression est de créer un modèle capable de prédire avec précision la sortie en fonction des caractéristiques d'entrée et de comprendre la relation sous-jacente entre les caractéristiques d'entrée et la sortie. Les techniques de régression les plus utilisées comprennent la régression linéaire, logistique et polynomiale [3].

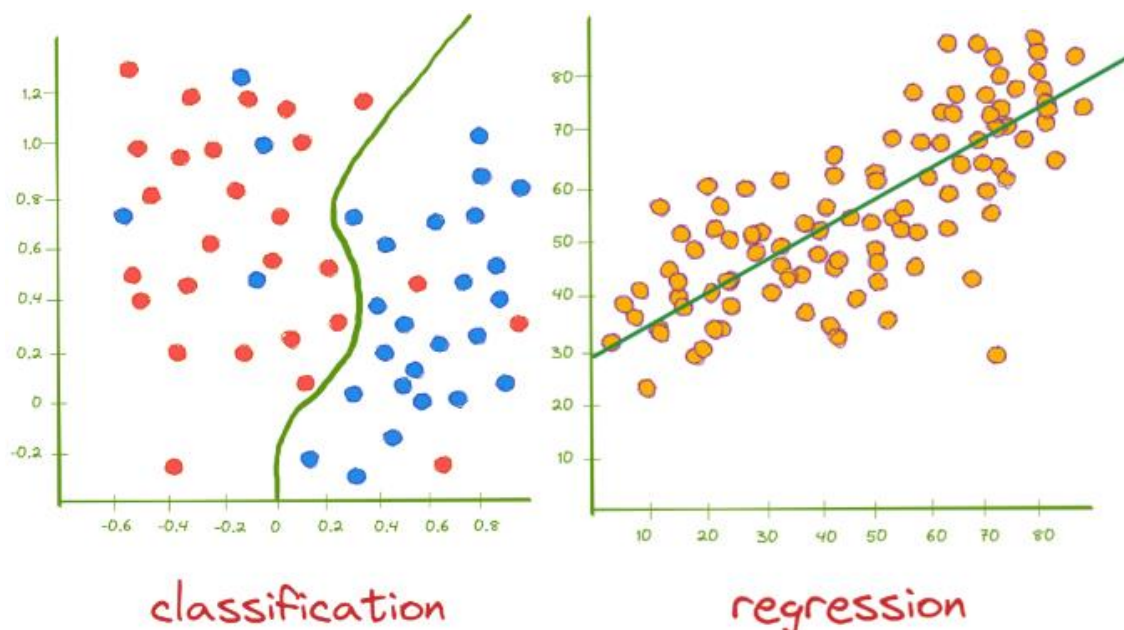


Figure 2: Classification et Régression [6]

### 1.7.2.2 L'apprentissage non-supervisé

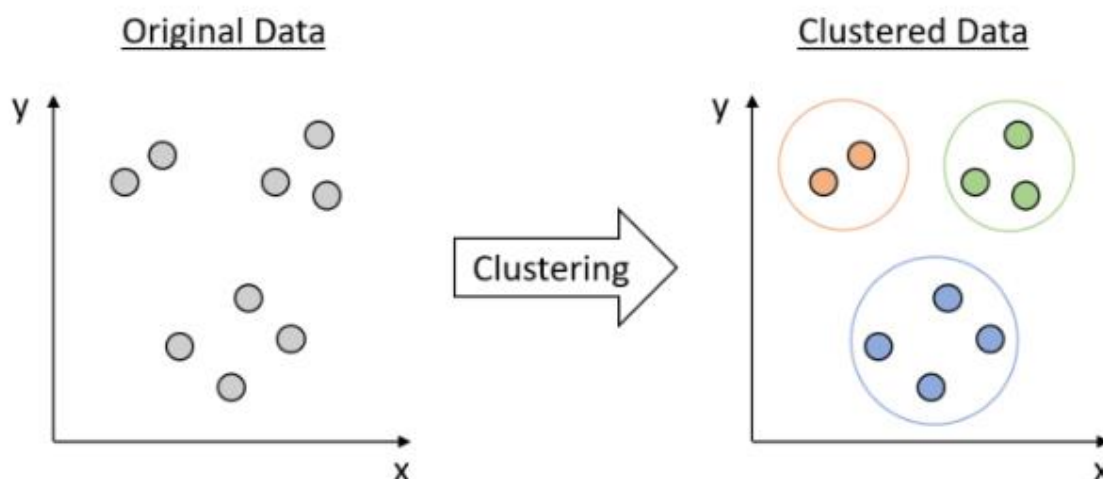
L'apprentissage non supervisé est une approche de l'apprentissage automatique où les données ne sont pas étiquetées par des classes ou des catégories prédéfinies. L'objectif de l'apprentissage non supervisé est de découvrir des structures ou des motifs intrinsèques dans les données, tels que des regroupements, des similarités ou des relations entre les variables. Les algorithmes d'apprentissage non supervisé cherchent à organiser et à représenter les données de



manière significative, sans la nécessité d'une supervision externe. Cela peut inclure des méthodes telles que la clustering, la réduction de dimension, l'analyse de motifs ou la détection d'anomalies [7].

## Clustering

Le regroupement (ou clustering) est une technique en apprentissage automatique qui consiste à regrouper des points de données en fonction de leurs similarités. L'objectif du regroupement est d'identifier des motifs dans les données et de regrouper les points de données similaires ensemble. Le regroupement est une technique d'apprentissage non supervisé, ce qui signifie qu'il ne nécessite pas de données étiquetées pour entraîner le modèle. Le regroupement peut être considéré comme le problème le plus important d'apprentissage non supervisé [8].



*Figure 3: Représentation graphique de la manière dont le clustering peut être utilisé pour identifier des modèles dans un ensemble de données [9].*

### 1.7.2.3 L'apprentissage Par Renforcement

L'apprentissage par renforcement est un type d'apprentissage automatique où un agent apprend à prendre des décisions en interagissant avec un environnement dynamique. L'agent est exposé à un ensemble d'états, il effectue des actions dans ces états et reçoit des récompenses(+) ou des sanctions(-) de l'environnement en fonction de ses actions. L'objectif de l'apprentissage par renforcement est d'apprendre une politique, c'est-à-dire une stratégie ou un ensemble de règles, qui maximise les récompenses cumulatives à long terme. Cela implique de trouver un équilibre entre l'exploration de nouvelles actions et l'exploitation des

actions déjà connues. L'apprentissage par renforcement est souvent utilisé pour résoudre des problèmes de prise de décision séquentielle, tels que les jeux, les robots autonomes et la gestion des ressources [10].

### 1.7.3 Apprentissage Profond

L'apprentissage profond (deep learning) est une branche de l'apprentissage automatique qui vise à modéliser et à comprendre des données complexes en utilisant des architectures de réseaux de neurones artificiels profonds. Les réseaux de neurones profonds sont des modèles composés de plusieurs couches de neurones interconnectés qui permettent de représenter et d'apprendre des caractéristiques hiérarchiques à partir des données.

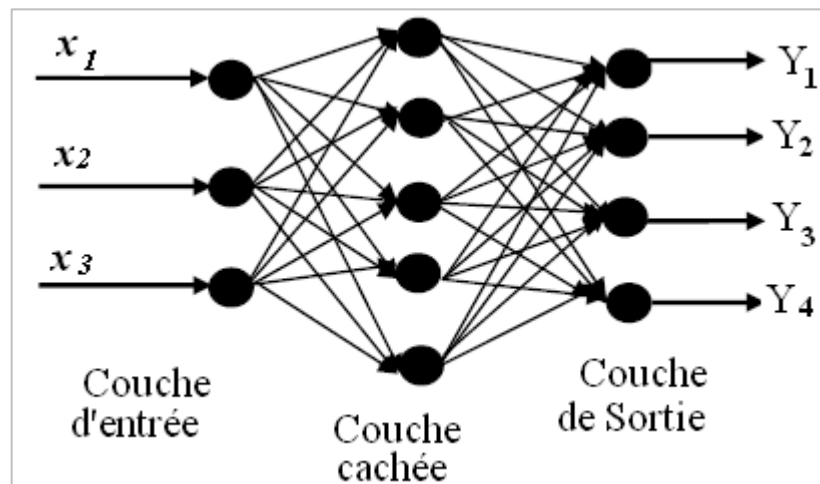


Figure 4: Structure d'un réseau de neurones [11]

Contrairement aux méthodes traditionnelles d'apprentissage automatique, l'apprentissage profond est capable d'apprendre de manière automatique et itérative des représentations de plus en plus abstraites et complexes à partir des données brutes, sans nécessiter d'ingénierie de caractéristiques manuelle. L'apprentissage profond a obtenu des résultats remarquables dans de nombreux domaines, tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance de la parole et bien d'autres [12].

### **1.7.4 Vision par Ordinateur**

La vision par ordinateur est un domaine de l'intelligence artificielle (IA) qui permet aux ordinateurs et aux systèmes de dériver des informations significatives à partir d'images numériques, de vidéos et d'autres entrées visuelles, et de prendre des mesures ou de faire des recommandations sur la base de ces informations. Si l'intelligence artificielle permet aux ordinateurs de penser, la vision par ordinateur leur permet de voir, d'observer et de comprendre, elle fonctionne de la même manière que la vision humaine, mais les êtres humains ont une longueur d'avance. La vue humaine a l'avantage de pouvoir s'entraîner à distinguer les objets, à en déterminer la distance, à savoir s'ils sont en mouvement et si quelque chose ne va pas dans une image.

La vision par ordinateur entraîne les machines à exécuter ces fonctions, mais elle doit le faire beaucoup plus rapidement à l'aide de caméras, de données et d'algorithmes qui remplacent nos rétines, nos nerfs optiques et notre cortex visuel. Parce qu'un système entraîné pour inspecter des produits ou surveiller un actif de production peut analyser des milliers de produits ou de processus à la minute, en remarquant des défauts ou problèmes imperceptibles, il peut rapidement dépasser les capacités humaines [13].

### **1.6.5 Difficultés que rencontrent les systèmes basées sur la vision par ordinateur**

Les systèmes basés sur la vision par ordinateur peuvent rencontrer plusieurs difficultés courantes telles que la mauvaise qualité d'image due à une faible résolution ou du bruit, des conditions d'éclairage médiocres, l'occlusion d'objets, des changements dans l'apparence des objets dus à des changements d'angle de vue, des conditions d'éclairage ou d'autres facteurs, la variabilité de l'apparence des objets due aux différences de forme, de taille, de couleur, de texture, etc., et la difficulté à détecter et à suivre les objets en mouvement [14].

## **1.6.6 Applications divers de la vision par ordinateur**

### **Voitures autonomes**

La vision par ordinateur est utilisée pour reconnaître et catégoriser des éléments (tels que des panneaux de signalisation ou des feux de circulation), construire des cartes 3D et estimer le mouvement, et a joué un rôle essentiel dans la réalisation des voitures autonomes. Les capteurs et les caméras des voitures autonomes recueillent des données sur leur environnement, les analysent et réagissent de manière appropriée [15].

### **Détection du cancer**

Les cliniciens peuvent repérer les anomalies et les altérations en comparant les cellules malades et non cancéreuses sur des photographies. La détection automatisée permet un diagnostic plus rapide du cancer à l'aide de données d'imagerie par résonance magnétique (IRM). Elle est déjà utilisée pour identifier le cancer du sein et de celui de la peau [15].

### **Agriculture**

Le secteur agricole a été témoin de plusieurs contributions des modèles d'intelligence artificielle (y compris la vision par ordinateur) dans des domaines tels que la surveillance des cultures et des rendements, la récolte automatisée, l'analyse des conditions météorologiques, la surveillance de la santé du bétail ou la détection des maladies des plantes [16].

### **Sécurité**

Les systèmes basés sur la vision par ordinateur pour la détection de vol se sont avérés plus précis car ils permettent une analyse en temps réel du mouvement. Les caméras alimentées par l'IA peuvent également être utilisées pour détecter d'autres types de comportements comme les agressions physiques ou les accidents, améliorant ainsi la sécurité.

### **Circulation piétonnière et comptage de foules**

Les caméras en magasin peuvent facilement capturer les données image ou vidéo que les systèmes de vision par ordinateur utilisent pour détecter et compter les personnes entrant et sortant du magasin. Il est particulièrement utile en période de COVID-19 avec des règles de

distanciation sociale appliquées et lorsqu'il n'y a qu'un nombre limité de personnes autorisées dans un magasin à la fois.

Le comptage des foules est également utile pour gérer les changements de personnel, les files d'attente et fournit des données analytiques utiles pour une meilleure gestion du magasin [16].

## **Détection d'actions**

La vision par ordinateur peut être utilisée pour analyser les performances sportives en détectant les actions des athlètes, comme la frappe d'une balle, le mouvement d'un corps ou l'exécution d'une technique spécifique, pour le domaine médicale. Elle peut être utilisée pour analyser des vidéos médicales et détecter des actions spécifiques, telles que des gestes chirurgicaux, des mouvements anormaux du corps ou des signes cliniques particuliers, et même pour détecter des actions qui peuvent être dangereuses, notamment détecter les chutes des patients, en particulier des personnes âgées. En utilisant des caméras ou des capteurs de mouvement, permettant ainsi une intervention médicale plus rapide [17].

## **1.7 Conclusion**

Dans ce chapitre, nous avons vus les différents aspects de la surveillance vidéo, ainsi que les différentes technologies de l'intelligence artificielle.

# CHAPITRE 2: Détection de chutes basée sur la vision par ordinateur

## 2.1 Introduction

Dans ce deuxième chapitre, nous explorerons plus en détail le domaine de la reconnaissance des activités humaines (HAR) et de la détection d'actions humaines, avec un accent particulier sur la détection des chutes.

Nous passerons ensuite en revue les travaux existants et la littérature scientifique sur la détection des chutes, en mettant en évidence les avancées réalisées dans ce domaine.

Enfin nous clôturons le chapitre par aborder brièvement les différents axes de recherche étroitement liés à la détection des chutes.

## 2.2 Reconnaissance des activités humaines (HAR)

La reconnaissance des activités humaines (HAR) est utilisée pour détecter et classifier les activités humaines sous des étiquettes appropriées, c'est le domaine le plus vaste qui regroupe tous les types d'activité humaine, de la plus facile à traiter à la plus pénible, Les activités humaines sont complexes et évoluent dans le temps, ce qui nécessite une division adéquate en sous-activités [18], parmi ces sous-activités, nous trouvons la détection d'action.

Une activité humaine est une tâche en cours composée de gestes, d'actions et d'interactions simples ou multiples, un geste fait référence au mouvement des parties du corps pour souligner un discours, tandis qu'une action désigne le mouvement collectif des parties du corps pour accomplir une tâche, par exemple, bouger la tête en signe de négation est un geste, marcher est une action, et parler bruyamment avec des expressions faciales désagréables est un comportement, celui de la colère.

L'interaction est une collection d'actions généralement réalisées par deux ou plusieurs sujets, par exemple, une conversation à deux personnes, une bagarre, la cuisine, la saisie de données, le lavage de voiture, etc. Les activités de groupe sont réalisées par plusieurs personnes et peuvent inclure une collection de gestes, d'actions et d'interactions, par exemple, un match de football ou une grève [18].

Dans notre travail, nous nous focalisons sur la détection d'actions.



*Figure 5: Sous-Activités de la Human Activity Recognition (HAR) [18],  
Ce travail se concentre sur la détection d'action.*

## 2.3 Détection d'actions Humaine

### 2.3.1 Définition

La détection d'actions ou (action detection) en anglais, est un domaine de recherche du domaine de la vision par ordinateur et de l'intelligence artificielle qui vise à détecter et à localiser les actions ou les activités dans les séquences vidéo.

La détection d'action humaine est le type spécifique de détection d'actions avec le plus d'intérêt, elle se focalise sur toute activité ou comportement fait par l'humain, son objectif principal est de détecter et de reconnaître les actions et les activités humaines qui se déroulent dans une séquence vidéo, elle vise également à classer ces actions détectées dans des catégories prédéfinies. Cela peut inclure des actions simples telles que "marcher", "courir", "sauter", ainsi que des actions complexes telles que "jouer au tennis", "conduire une voiture", "danser", etc., un autre but de la détection d'action humaine est de déterminer le moment précis où une action spécifique commence et se termine dans la séquence vidéo. Cela permet d'obtenir des informations temporelles précises sur les actions qui se déroulent.

### 2.3.2 Domaines d'applications

La détection d'actions humaine dans les séquences vidéo a un large éventail d'applications dans de nombreux domaines. Voici quelques domaines d'actualité :

- **Surveillance et sécurité** : La détection d'actions est largement utilisée dans les systèmes de surveillance et de sécurité pour détecter et reconnaître les activités humaines suspectes ou indésirables. Cela peut inclure la détection de comportements suspects, d'intrusions, de bagarres, de vols, etc.
- **Reconnaissance de gestes** : La détection d'actions est utilisée pour reconnaître et interpréter les gestes réalisés par les individus, cela peut être utile dans des domaines tels que la réalité virtuelle, la robotique, la réadaptation médicale, etc.
- **Domaine médicale** : peut inclure la surveillance, l'évaluation et le suivi des mouvements des patients, l'aide à la rééducation, à l'amélioration des techniques chirurgicales, et aussi à la détection et la reconnaissance des situations où un individu tombe ou perd l'équilibre, particulièrement chez les personnes âgées, afin de fournir une assistance médicale immédiate en cas de chute pour prévenir d'éventuelles complications ou blessures.

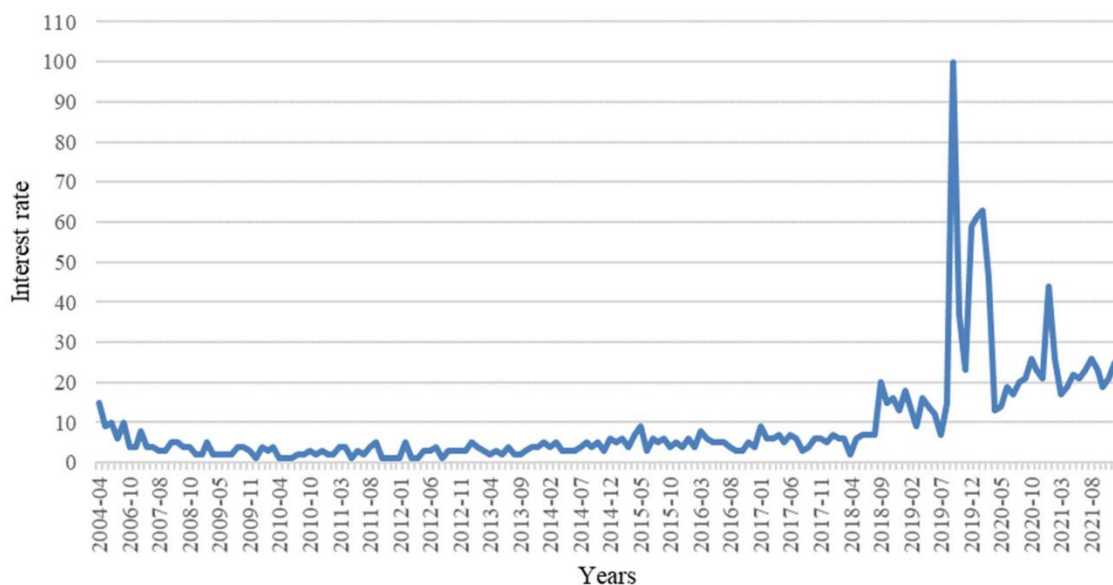
La détection d'action humaine est une sous-activité de la (HAR), mais cette dernière reste très vaste, en raison de l'existence de dizaines de types d'actions, chacune son type et sa nature, et aussi chacune son approche.

Dans notre travail, nous avons choisi une action spécifique qui est d'actualité, notamment dans le domaine médicale, c'est la détection de chute humaine, ou en anglais (FALL detection).



## 2.3 Détection de chutes (FALL Detection)

La détection des chutes fait référence à la capacité de détecter et de reconnaître lorsque qu'une personne tombe ou perd l'équilibre de manière involontaire. Elle utilise généralement des capteurs, des dispositifs ou des algorithmes basée sur la vision par ordinateur pour détecter les changements soudains ou anormaux dans la position, le mouvement ou l'accélération du corps, ce qui indiquent une chute potentielle. L'objectif principal de la détection des chutes est d'alerter rapidement les personnes concernées ou les systèmes de surveillance afin de fournir une assistance médicale ou une intervention appropriée [19].



*Figure 6: Taux d'intérêt pour la détection des chutes, d'avril 2004 à décembre 2021. Les données expérimentales sont obtenues à partir de "Google Trends" en utilisant "détection des chutes" comme objet de recherche. Les résultats sont normalisés en utilisant les taux les plus élevés ; ainsi, le taux d'intérêt le plus élevé a une valeur de 100*

### 2.3.1 L'importance de la détection des chutes

Pourquoi est-il crucial de détecter les chutes ?

Tout d'abord, les chutes sont une cause fréquente de blessures graves, en particulier chez les personnes âgées. Les conséquences d'une chute peuvent être graves, allant des fractures osseuses aux lésions cérébrales traumatiques, et peuvent même entraîner des décès. La détection précoce des chutes permet d'obtenir une intervention médicale rapide et appropriée, ce qui peut réduire les complications et améliorer les résultats pour les personnes concernées.

Deuxièmement, la détection des chutes est essentielle pour assurer la sécurité des personnes vulnérables, en particulier dans les établissements de soins de santé, les maisons de retraite et les environnements domestiques. Les chutes peuvent survenir de manière imprévisible et sans surveillance, et peuvent mettre en danger la vie et le bien-être des individus. La détection des chutes permet de déclencher des alertes et des mesures de sécurité appropriées, comme le déclenchement d'une alarme ou l'envoi d'une assistance médicale immédiate.

En outre, la détection des chutes peut contribuer à la prévention des chutes en identifiant les facteurs de risque et les comportements associés aux chutes. En analysant les données sur les chutes, il est possible d'identifier les modèles et les précurseurs de chute, ce qui peut aider à mettre en place des mesures de prévention et des interventions ciblées pour réduire les risques de chute.

En résumé, la détection des chutes est cruciale pour prévenir les blessures graves, assurer la sécurité des personnes vulnérables et mettre en place des mesures de prévention appropriées. Cela contribue à améliorer la qualité de vie, la sécurité et le bien-être des individus concernés.

### 2.3.2 Revue de littérature et travaux au sujet de la détection de chutes

Xu et al. (2018) [20] ont proposé une taxonomie des méthodes de détection de chutes basée sur le capteur utilisé, (La taxonomie est la science qui consiste à nommer, décrire et classer les organismes en fonction de leurs caractéristiques communes). Trois grandes catégories ont été proposées comme indiqué dans la figure 7 : basées sur l'accéléromètre, basées sur les capteurs de radiofréquence (RF) et basées sur la vision.

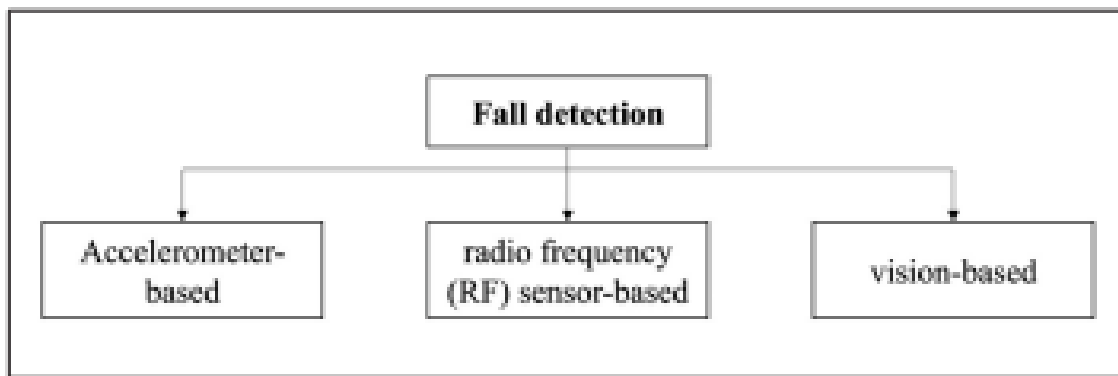


Figure 7: Taxonomie des méthodes de détection de chutes [20].

La première catégorie fait référence aux dispositifs d'accéléromètre et aux accéléromètres intégrés aux smartphones (les sensors). Les auteurs Wang, Zhang, Li, Lee & Sherratt (2014) [21], ont utilisé un capteur d'accéléromètre pour détecter une chute. Ils ont calculé le SVMA (Activité Physique Modérée à Vigoureuse) du sujet et l'ont comparé à un seuil prédéfini. Si la valeur du SVMA dépasse le seuil, ils calculent la fréquence cardiaque à l'aide d'un capteur de pression d'impulsion et de l'angle du tronc. Le système contactera les services d'urgence si ces deux valeurs sont anormales. Ils ont obtenu une précision de 97.5%.

Les accéléromètres sont le type de capteur le plus utilisé dans les systèmes de détection de chute avant 2014 (Xu et al. 2018) [20]. Les chercheurs préfèrent utiliser cette technologie en raison de son faible coût, de sa portabilité et parce qu'elle ne soulève pas de problèmes de confidentialité. Les accéléromètres se sont avérés suffisamment efficaces pour détecter les chutes avec une grande précision. Cependant, ils présentent certaines limitations. Cette approche nécessite que le sujet porte le capteur toute la journée, ce qui est non seulement inconfortable, mais affecte également sa vie quotidienne [22].

La deuxième catégorie comprend le WI-FI et le radar. Cette méthode est basée sur le suivi des fluctuations du signal pour détecter les chutes. Tian et al. (2018) [23] ont utilisé la radio à ondes continues à modulation de fréquence (Frequency-modulated continuous-wave FMCW) équipée de deux antennes pour collecter les réflexions de fréquence radio (RF) provenant de l'environnement. Deux cartes thermiques ont été générées à partir de ces réflexions et utilisées pour entraîner un modèle CNN. Cette méthode a atteint une sensibilité de 94% et une précision de 92%.

Les capteurs RF constituent une méthode non intrusive et ont atteint un bon niveau de précision dans les travaux précédents. Cependant, ils sont sensibles aux interférences avec d'autres dispositifs basés sur RF. Dans notre cas, il sera difficile de placer une antenne dans chaque cellule sans risque d'interférence [22].

La dernière catégorie comprend les caméras RGB et les caméras de profondeur. Les caméras RGB ont été adoptées dans les méthodes basées sur la vision précédentes. Dans le travail de Chua et al. (2015) [24], les auteurs ont utilisé des images RGB pour détecter les chutes en utilisant la forme humaine. Après l'extraction de premier plan, ils ont représenté la forme humaine par trois points (tête, corps et jambes) et calculé les caractéristiques liées à la chute. L'événement de chute est détecté en comparant ces caractéristiques à un seuil. Ils ont obtenu une précision de 90,5%. Les méthodes basées sur les caméras traditionnelles ne sont pas robustes et leurs performances sont limitées en raison de la difficulté à distinguer le premier plan de l'arrière-plan dans les images RGB. De plus, elles sont coûteuses en termes de calcul. Le principal inconvénient de ces caméras est leur sensibilité aux variations d'éclairage (Xu et al.) (2018) [20].

Avec les caméras de profondeur, certaines limitations des caméras RGB peuvent être surmontées. En plus d'une image RGB, ces caméras fournissent une image de profondeur dans laquelle chaque pixel représente la distance de l'objet correspondant à la caméra. Différentes caméras de profondeur ont été commercialisées ces dernières années, telles que Kinect de Microsoft, Real Sense d'Intel et Pro Live d'Asus Xtion. Kinect v2 est le successeur de Kinect v1 et a été publié en juillet 2014. Il est devenu le capteur le plus populaire dans les systèmes de détection de chutes Xu et al. (2018) [20]. Il est composé d'une caméra RGB, d'une caméra infrarouge (IR) basée sur la technologie ToF (time-of-flight) et d'un ensemble de microphones. Kinect V2 fournit différents flux de données : image RGB, carte de profondeur, image IR, données de squelette et audio.

La plupart des systèmes basés sur la vision utilisent le squelette et les trajectoires des articulations du corps humain pour détecter les activités quotidiennes des personnes car ils sont robustes aux variations d'éclairage et aux variations de scène (Yan et al.) (2018) [25].

(Gasparrini et al.) (2015) [26] ont opté pour une approche de fusion des données où ils ont utilisé à la fois les données de squelette de Kinect v2 et les données d'accélération d'un accéléromètre portable pour obtenir une meilleure précision. La chute est détectée avec une précision de 99% en exploitant la variation des positions des articulations du squelette, la distance entre l'articulation de la base de la colonne vertébrale et le sol, et l'amplitude de l'accélération mesurée par l'accéléromètre à la taille. Les auteurs ont opté pour une méthode basée sur un seuil. Le jeu de données TST v2 est utilisé pour évaluer cette approche. Les résultats sont très bons sur ce jeu de données, mais la méthode basée sur les seuils n'est pas robuste pour être généralisée car les seuils eux-mêmes dépendent des caractéristiques du sujet telles que le sexe et l'âge. (Ramachandran & Karuppiah) (2020) [27].

L'utilisation à la fois d'une caméra et d'un accéléromètre est plus complexe que l'utilisation d'un seul capteur car ils doivent être synchronisés. De plus, les capteurs doivent être positionnés sur le corps en faisant attention à leur orientation, ce qui les rend peu pratiques dans notre cas [22].

Dans l'article de Le et al. (2014) [28], les données du squelette fournies par Kinect ont été utilisées pour calculer le plan correspondant au sol de la pièce. Ensuite, les auteurs ont calculé la distance et la vitesse de la tête et de la colonne vertébrale par rapport au sol. Enfin, ils ont utilisé SVM pour la classification. Ils ont obtenu un taux de faux positifs de 0% et un taux de faux négatifs de 3,3%, ce qui est un résultat très intéressant. Le jeu de données utilisé dans cette étude n'est pas public, nous ne pouvons donc pas vérifier les résultats, et l'auteur a mentionné avoir supprimé certains échantillons ressemblant à des chutes (s'asseoir et se coucher par terre) pour obtenir de meilleurs résultats.

La plupart des travaux précédents utilisent des caractéristiques prédéfinies (hand-crafted features en anglais), car cela ne nécessite pas un grand jeu de données et permet de se concentrer explicitement sur les caractéristiques qui contribuent largement à la tâche spécifique. Cependant, il peut être difficile de trouver des caractéristiques raisonnables et cohérentes dans certaines tâches qui nécessitent une connaissance approfondie du domaine.

D'autre part, nous avons la méthode de l'apprentissage profond, qui surpasse la méthode d'extraction de features prédéfinies (hand-crafted) car elle peut obtenir de plus en plus de caractéristiques via l'apprentissage, mais elle nécessite beaucoup plus de données, ce qui présente une deuxième limitation [22].

À ce jour, les réseaux de neurones se sont avérés très efficaces pour effectuer la détection de chutes. En particulier, deux approches, y compris les modèles de réseaux de neurones convolutifs et les modèles de réseaux de neurones récurrents, par exemple le travail de A. R. Fansdana. Et al. (2021) [29] qui était le 1<sup>er</sup> travail à utiliser le modèle CNN pour la détection de chute chez les personnes âgées, et qui a atteint de bons résultats [29].

Selon Oussema & Rita (2020) [22], la limitation principale des approches de détection de chute basée sur l'apprentissage profond Xu et al. (2018) [20] Hwang et al. (2017) [30] Fakhruddin et al. [31] (2017), est la limitation de taille des ensembles de données disponibles pour la détection de chutes, ce qui restreint les performances de l'apprentissage profond.

## **2.4 Axes relié à la détection de la chute**

Le domaine de la détection de chutes humaines est étroitement lié à d'autres axes de recherches qui analysent le mouvement humain à partir d'images et de vidéos, nous présenterons brièvement quelque uns.

### **2.4.1 Détection par Gait (la démarche)**

Gait ou la démarche en français, est un terme utilisé pour décrire la manière dont une personne marche. Elle fait référence au schéma de mouvement des jambes et des pieds lors de la marche, ainsi qu'à la posture du corps et à la manière dont le poids du corps se déplace pendant le mouvement.

La reconnaissance du Gait fait référence au processus d'identification d'une personne en fonction de son style de marche ou de sa manière de marcher. Cela implique d'analyser le rythme, la vitesse et le style des mouvements d'une personne lorsqu'elle marche. La reconnaissance du Gait peut être utilisée pour des applications telles que l'identification biométrique, la sécurité et la surveillance de la santé des individus [32].

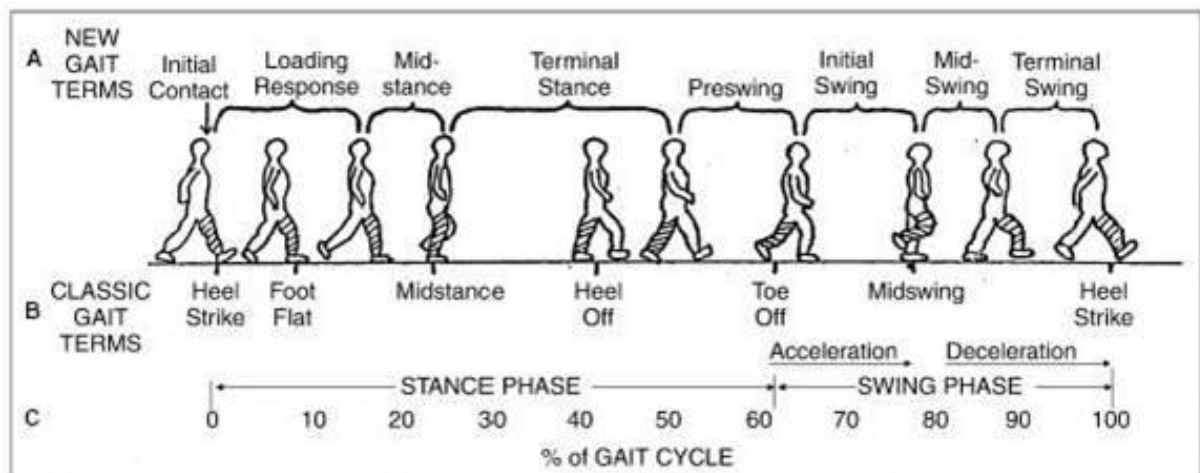


Figure 8: Représentation d'un cycle complet de Gait

La démarche peut fournir des informations importantes sur la santé physique et médicale d'une personne, ainsi que sur sa mobilité et son équilibre général. Les anomalies du Gait peuvent être indicatives de problèmes tels que des douleurs articulaires, une faiblesse musculaire, des troubles neurologiques ou d'autres problèmes de santé [33], et dans certains cas, elle est utilisée pour diagnostiquer et traiter ces affections, ainsi que pour suivre les progrès d'une personne lors de la rééducation.

## 2.4.2 Analyse Comportementale

L'analyse comportementale, également connue sous le nom de "behavioral analysis" en anglais, est une discipline qui vise à étudier et comprendre le comportement humain à travers l'observation et l'analyse de diverses caractéristiques, actions et interactions. Elle utilise des techniques et des méthodes issues des domaines de la psychologie, de la sociologie, de l'éthologie et des sciences cognitives pour analyser les comportements individuels et collectifs.

L'objectif de l'analyse comportementale est de détecter, reconnaître et analyser les comportements humains dans divers contextes et scénarios. Cela peut inclure la détection des actions spécifiques, telles que la marche, la course, l'interaction sociale, ainsi que la

reconnaissance des activités complexes telles que la danse, les sports, les gestes et les expressions faciales.

L'analyse comportementale dans les vidéos consiste à utiliser des techniques d'intelligence artificielle et de vision par ordinateur pour extraire et interpréter les comportements humains à partir de séquences vidéo. Elle vise à détecter, suivre et reconnaître les actions, les interactions sociales, les activités anormales et les schémas de comportement dans les vidéos [34].

### **2.4.3 Les détections d'actions humaines basées sur le squelette et sur l'estimation de la pose**

La détection d'action humaine basée sur le squelette (Skeleton based en anglais) et la détection d'action humaine basée sur l'estimation de pose (Pose estimation based en anglais) sont deux approches de vision par ordinateur qui visent à reconnaître et comprendre les actions effectuées par un individu en utilisant les informations sur les positions et les mouvements du corps.

La détection d'action humaine basée sur le squelette utilise des capteurs tels que des caméras de profondeur (RGBD) ou des capteurs de mouvement pour capturer les positions des articulations du corps. En analysant les schémas de mouvement du squelette, il est possible de reconnaître différentes actions humaines. Cette approche est largement utilisée dans des domaines tels que la surveillance de la sécurité, la réalité virtuelle et la santé, où il est important de détecter et suivre les mouvements humains avec précision, contrairement aux données de caméras normaux (RGB), les données de squelette sont moins sensibles aux variations d'éclairage, aux occlusions et aux changements d'arrière-plan, ce qui permet une détection plus robuste des actions humaines dans différentes conditions. De plus, une séquence de squelette fournit des trajectoires de mouvement humain en n'utilisant qu'un petit nombre de positions articulaires. Par conséquent, elle offre l'avantage d'un faible volume de calcul [35].

De même, la détection d'action humaine basée sur l'estimation de pose repose sur des algorithmes avancés de détection et de suivi des points clés du corps humain à partir d'images ou de vidéos. En analysant les positions et les relations entre les articulations, cette méthode permet de reconnaître et comprendre les actions humaines. Elle est utilisée dans des applications telles que la réalité augmentée, la surveillance vidéo et l'animation 3D.



Ces deux approches offrent des moyens puissants pour comprendre et suivre les mouvements humains dans divers domaines d'application. Elles permettent d'analyser et d'interpréter les actions effectuées par les individus, ouvrant ainsi de nombreuses possibilités pour des applications telles que la surveillance de la santé, l'assistance aux personnes âgées, la sécurité, et bien d'autres.

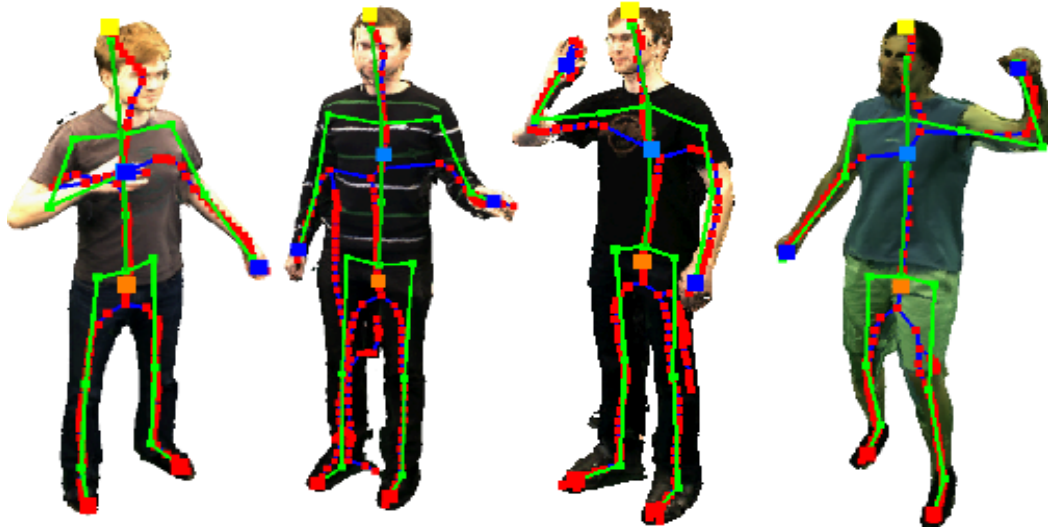


Figure 9: Exemples d'estimation de pose humaine sur des données réelles basée sur le squelette [36].

#### 2.4.4 Détection d'anomalies

La détection des anomalies consiste à identifier des modèles, des comportements ou des événements inhabituels dans un ensemble de données, qui peuvent être indicatifs de problèmes, de changements inattendus ou de situations exceptionnelles. L'objectif est de détecter des points de données qui s'écartent significativement du comportement normal ou attendu, afin de les examiner plus attentivement ou de prendre des mesures appropriées [37].

Il est très difficile de définir explicitement ce que sont des activités anormales ou inhabituelles, car leur définition dépend des contextes et des environnements. De plus, par définition, ils ne sont pas fréquemment observés ; sinon elles seront normales ou habituelles [38].

## **2.5 Conclusion**

Dans ce chapitre, nous avons exploré ce que sont la reconnaissance d'activités humaines, la détection d'actions et la détection de chutes en plus de quelques axes reliés au contexte de notre sujet.

## CHAPITRE 3: Implémentation

### 3.1 Introduction

Nous présenterons dans ce chapitre les différentes étapes de notre travail, en commençant par la récolte des données jusqu'à la discussion des résultats.

### 3.2 Environnement d'exécution

Le Deep Learning est un domaine avec des exigences de la disponibilité des ressources matériel (surtout en GPU) capable de faire des calculs intenses, en totalité, nous avons utilisé deux machines à calculer, dotées d'environnement locale de développement Python en utilisant l'IDE Pycharm Community.

**Machine 1** : Pc HP, processeur i5 8<sup>ème</sup> génération, 8gb Ram, carte graphique Intel intégrée.

**Machine 2**: Pc HP, Processeur i7 8<sup>ème</sup> génération, 16 GB Ram, carte graphique: NVIDIA GeForce MX 130.

**Python**: Nous avons opté pour Python comme langage de programmation. Ce langage a été créé par Guido van Rossum et sa première version a été publiée en 1991. Python est un langage interprété, ce qui signifie qu'il n'est pas nécessaire de le compiler avant de l'exécuter [39]. Il est gratuit et peut être utilisé sans restriction dans des projets commerciaux. De plus, il est portable sur diverses variantes d'Unix, mais également sur des systèmes d'exploitation propriétaires tels que Mac OS, BeOS, NeXTStep, MS-DOS et diverses variantes de Windows. Python est caractérisé par sa syntaxe simple et des types de données avancés tels que des listes, des dictionnaires, etc. qui permettent de générer des programmes compacts et lisibles [40].

**IDE Pycharm Community**: Pycharm est un environnement de développement intégré (IDE) conçu spécifiquement pour Python. Il offre des fonctionnalités telles que l'analyse de code, les tests unitaires, l'utilisation de systèmes de contrôle de version (Git, Subversion, Mercurial, Perforce ou CVS), ainsi que la possibilité d'utiliser des frameworks populaires tels

que Django, web2py et Flask. PyCharm dispose également d'un débogueur intégré et d'une fonction de refactoring [41], il permet de modifier rapidement et facilement le code tout en préservant son fonctionnement.

**Keras:** Keras est une bibliothèque open source spécialisée dans les réseaux de neurones. Elle offre des capacités avancées pour entraîner des fonctions d'activation. Keras peut être utilisé avec différents frameworks de Deep Learning en tant que backend. Pour passer d'un framework à un autre, il suffit de modifier le fichier `keras.json`, situé dans le même répertoire que l'installation de Keras [42].

**Pandas:** Pandas est une bibliothèque Python qui offre des structures de données performantes, flexibles et expressives, conçues pour simplifier la manipulation de données relationnelles ou étiquetées. Il est en train de devenir l'outil open source d'analyse et de manipulation de données le plus puissant et le plus flexible disponible dans tous les langages de programmation [43]. Nous avons choisi d'utiliser Pandas dans notre projet pour faciliter la manipulation de l'ensemble de données que nous avons utilisé, qui était au format `.csv`.

**NumPy :** NumPy, abréviation de Numerical Python, est une bibliothèque qui propose des objets de tableau multidimensionnel et une collection de fonctions pour manipuler ces tableaux. L'utilisation de NumPy simplifie les opérations mathématiques et logiques sur les tableaux [43]. Son rôle est similaire à celui de la bibliothèque précédente, Pandas, dans la manipulation et l'analyse des données.

**Scikit-learn :** Scikit-learn est une bibliothèque Python gratuite dédiée à l'apprentissage automatique. Elle offre une variété d'algorithmes accompagnés de leur documentation, ainsi que des fonctions pour calculer les mesures de performance telles que l'exactitude, la précision, etc. Parmi les algorithmes disponibles, on retrouve les K plus proches voisins, les forêts aléatoires, l'analyse en composantes principales (ACP), etc. Scikit-learn prend également en charge les bibliothèques numériques et scientifiques de Python, telles que NumPy, pour faciliter le traitement des données et les calculs nécessaires aux modèles d'apprentissage automatique [44].

## 3.3 Modèles d'apprentissage automatique utilisés

### 3.3.1 YOLO

#### C'est quoi YOLO ?

YOLO (You Only Look Once) est un algorithme de détection d'objets en temps réel largement utilisé dans le domaine de la vision par ordinateur. Contrairement à d'autres méthodes qui utilisent des régions d'intérêt (ROIs) pour détecter les objets, YOLO effectue la détection en une seule passe sur l'image entière en utilisant un réseau de neurones convolutionnels. Cela permet à YOLO d'être rapide et efficace pour la détection d'objets en temps réel [45].

#### Architecture du modèle YOLO

L'architecture YOLO se compose d'un réseau neuronal convolutif (CNN) qui traite l'image complète et génère une carte de caractéristiques. Cette carte de caractéristiques est utilisée pour effectuer des prédictions à différentes échelles, capturant ainsi des objets de différentes tailles. Pendant l'entraînement, YOLO ajuste ses paramètres en minimisant les erreurs entre les boîtes englobantes prédites et les boîtes réelles, ainsi que les erreurs de classification.

Le fonctionnement de l'architecture est le suivant :

- L'image d'entrée est redimensionnée en 448x448 avant d'être traitée par le réseau convolutif.
- Une convolution 1x1 est appliquée pour réduire le nombre de canaux, suivie d'une convolution 3x3 qui génère une sortie cubique.
- La fonction d'activation utilisée est ReLU, à l'exception de la couche finale qui utilise une fonction d'activation linéaire.
- Des techniques supplémentaires telles que la normalisation par lots et l'abandon sont utilisées pour régulariser le modèle et éviter le surajustement.

#### Fonctionnement du modèle YOLO

À la base, YOLO divise l'image d'entrée en une grille de cellules, et pour chaque cellule, il prédit les boîtes englobantes et les probabilités de classe. Ces boîtes englobantes représentent

les emplacements et les tailles des objets potentiels dans l'image. YOLO génère également un score de confiance correspondant indiquant la présence d'un objet dans chaque boîte, ainsi que les probabilités de classe pour différentes catégories d'objets.

L'algorithme se base sur les quatre approches suivantes :

- Residual blocks : Cette étape consiste à diviser l'image originale (A) en une grille de cellules de taille  $N \times N$ , où  $N$  est généralement 4. Chaque cellule de la grille est chargée de localiser et prédire la classe de l'objet qu'elle couvre, ainsi que la valeur de probabilité/confiance.

- Bounding box régression : consiste à déterminer les boîtes englobantes qui correspondent à des rectangles mettant en évidence tous les objets de l'image. Nous pouvons avoir autant de boîtes englobantes qu'il y a d'objets dans une image donnée.

- Intersection sur les syndicats ou IOU: Le processus de sélection de l'IOU commence par la définition par l'utilisateur d'un seuil, généralement fixé à 0,5. YOLO calcule l'IOU pour chaque cellule de la grille en évaluant la zone d'intersection divisée par la zone d'union. Les cellules de la grille dont la valeur IOU est inférieure au seuil sont considérées comme non pertinentes et exclues, tandis que celles dont la valeur IOU est supérieure au seuil sont considérées comme pertinentes.

- Non-Max Suppression ou NMS: Définir un seuil pour l'IOU n'est pas toujours suffisant car un objet peut avoir plusieurs cases avec IOU au-delà du seuil, et laisser toutes ces cases peut inclure du bruit. C'est ici que nous pouvons utiliser NMS pour ne conserver que les boîtes avec le score de probabilité de détection le plus élevé [46].

## Chronologie du modèle YOLO

Depuis sa création, la famille YOLO a évolué à travers plusieurs itérations, chacune s'appuyant sur les versions précédentes pour répondre aux limitations et améliorer les performances (voir Figure 10).



*Figure 10: Chronologie des versions de YOLO [45]*

Le premier YOLO par Joseph Redmon et al. a été publié lors de CVPR 2016 [47]. Il a présenté pour la première fois une approche en temps réel de bout en bout pour la détection d'objets. Le nom YOLO signifie "You Only Look Once", faisant référence au fait qu'il était capable d'accomplir la tâche de détection en une seule passe du réseau, contrairement aux approches précédentes qui utilisaient soit des fenêtres glissantes suivies d'un classifieur qui devait s'exécuter des centaines ou des milliers de fois par image, soit des méthodes plus avancées qui divisaient la tâche en deux étapes, où la première étape détectait les régions possibles avec des objets ou des propositions de régions et la deuxième étape exécutait un classifieur sur les propositions. De plus, YOLO utilisait une sortie plus directe basée uniquement sur la régression pour prédire les sorties de détection, contrairement à Fast R-CNN [48] qui utilisait deux sorties distinctes, une classification pour les probabilités et une régression pour les coordonnées des boîtes.

Depuis 2016, plus de 16 versions de YOLO ont vu le jour, chacune plus rapide et plus efficace que son prédécesseur, YOLO-NAS étant le plus récent (Mai 2023), on peut se baser sur 4 axes principaux pour effectuer la comparaison :

- **Anchor (ancres) :** Le modèle YOLO original était relativement simple et n'utilisait pas d'ancres, tandis que les modèles les plus avancés utilisaient des détecteurs à deux étapes avec des ancres. YOLOv2 a incorporé des ancres, ce qui a entraîné des améliorations de la précision de prédiction des boîtes englobantes. Cette tendance s'est maintenue pendant cinq ans jusqu'à ce que YOLOX (2021) introduise une approche sans ancres qui a obtenu des résultats de pointe. Depuis lors, les versions ultérieures de YOLO ont abandonné l'utilisation d'ancres.
- **Cadre de travail :** Initialement, YOLO a été développé en utilisant le cadre de travail Darknet, et les versions ultérieures ont suivi cette voie. Cependant, lorsque Ultralytics a porté YOLOv3 vers PyTorch, les versions restantes de YOLO ont été développées en utilisant PyTorch, ce qui a entraîné une augmentation des améliorations. Un autre langage d'apprentissage en profondeur utilisé est PaddlePaddle, un cadre de travail open-source initialement développé par Baidu.
- **Architecture de base :** Les architectures de base des modèles YOLO ont subi d'importants changements au fil du temps. À partir de l'architecture Darknet, qui comprenait des couches de convolution simples et des couches de max pooling, les modèles ultérieurs ont incorporé des connexions partielles inter-étapes (CSP) dans YOLOv4, une reparamétrisation dans YOLOv6 et YOLOv7, et une recherche d'architecture neuronale dans DAMO-YOLO et YOLO-NAS.
- **Performance :** Bien que les performances des modèles YOLO se soient améliorées au fil du temps, il convient de noter qu'ils privilégient souvent l'équilibre entre la vitesse et la précision plutôt que de se concentrer uniquement sur la précision. Ce compromis est essentiel au cadre de travail YOLO, permettant la détection en temps réel d'objets dans diverses applications [49].



Version	Date	Anchor	Framework	Backbone
YOLO	2015	No	Darknet	Darknet24
YOLOv2	2016	Yes	Darknet	Darknet24
YOLOv3	2018	Yes	Darknet	Darknet53
YOLOv4	2020	Yes	Darknet	CSPDarknet53
YOLOv5	2020	Yes	Pytorch	YOLOv5CSPDarknet
PP-YOLO	2020	Yes	PaddlePaddle	ResNet50-vd
Scaled-YOLOv4	2021	Yes	Pytorch	CSPDarknet
PP-YOLOv2	2021	Yes	PaddlePaddle	ResNet101-vd
YOLOR	2021	Yes	Pytorch	CSPDarknet
YOLOX	2021	No	Pytorch	YOLOXCSPDarknet
PP-YOLOE	2022	No	PaddlePaddle	CSPRepResNet
YOLOv6	2022	No	Pytorch	EfficientRep
YOLOv7	2022	No	Pytorch	YOLOv7Backbone
DAMO-YOLO	2022	No	Pytorch	MAE-NAS
YOLOv8	2023	No	Pytorch	YOLOv8CSPDarknet
YOLO-NAS	2023	No	Pytorch	NAS

Figure 11: Sommaire des architectures de YOLO [49]

### 3.3.2 Les Classificateurs Supervisés utilisés

Un classificateur supervisé est un algorithme d'apprentissage automatique qui utilise un ensemble de données d'entraînement pour apprendre à classifier les données en fonction de leurs caractéristiques. La classification supervisée consiste à attribuer automatiquement une catégorie (ou une classe) à des données dont on ne connaît pas la catégorie, en utilisant un ensemble de données annoté.

Les méthodes de classification supervisée sont souvent utilisées pour classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets. Les méthodes s'étendent souvent à des variables Y quantitatives (régression).

Voici les classificateurs supervisés utilisés dans notre travail :

## KNN

Le KNN (K-Nearest Neighbors) Classifier est un algorithme de classification supervisée qui attribue une étiquette de classe à un nouvel exemple en se basant sur les classes des  $k$  échantillons les plus proches dans l'espace des caractéristiques. Cet algorithme calcule la distance entre le nouvel exemple et tous les exemples d'apprentissage, puis sélectionne les  $k$  voisins les plus proches. La classe majoritaire parmi ces voisins est attribuée à l'exemple de test. Le choix de la valeur de  $k$  affecte la sensibilité de l'algorithme aux bruits et à la précision de la classification [50].

## Gradient Boosting classifier

Le Gradient Boosting Classifier est un algorithme d'apprentissage supervisé utilisé pour la classification. Il s'agit d'une méthode d'ensemble où plusieurs modèles de faible complexité, appelés arbres de décision, sont entraînés de manière séquentielle pour corriger les erreurs commises par les modèles précédents. À chaque étape, le modèle tente de prédire les résidus ou les erreurs des prédictions précédentes. Ces résidus sont ensuite pondérés et ajoutés aux prédictions actuelles. Ce processus itératif permet d'améliorer progressivement la performance du modèle en combinant les forces des modèles précédents. Le résultat final est une combinaison pondérée des prédictions de tous les modèles [51].

## eXtreme Gradient Boosting

XGBoost est une version améliorée du gradient boosting avec des fonctionnalités et des optimisations supplémentaires. Il offre de meilleures performances, des techniques de régularisation, une gestion des valeurs manquantes et des capacités de traitement parallèle par rapport à GBC [51].

## Histogram Gradient Boosting

L'histogramme de gradient boosting est une variante du gradient boosting qui utilise des histogrammes pour représenter les caractéristiques continues au lieu des seuils, offrant ainsi des avantages en termes de vitesse et de performance par rapport à l'approche traditionnelle [52].

## ADA Boost

AdaBoost (Adaptive Boosting) est un algorithme d'apprentissage automatique utilisé pour la classification binaire. Il fonctionne en combinant plusieurs classificateurs faibles pour former un classificateur fort. L'idée principale derrière AdaBoost est de donner plus de poids aux échantillons mal classés lors de l'entraînement de chaque classifieur faible successif, de manière à se concentrer davantage sur les échantillons difficiles à classer. Cela permet à chaque classifieur faible de se spécialiser dans la classification de certaines instances difficiles. Le classifieur fort final est une combinaison pondérée de tous les classificateurs faibles, où les poids sont attribués en fonction de la performance de chaque classifieur faible. AdaBoost est connu pour sa capacité à gérer des ensembles de données complexes et à obtenir de bons résultats en termes de précision de classification [53].

## Random Forest Decision Tree

Un arbre de décision est un algorithme d'apprentissage automatique qui utilise une structure arborescente pour prendre des décisions basées sur des règles conditionnelles. Chaque nœud de l'arbre représente une caractéristique, et chaque branche représente une décision basée sur cette caractéristique. L'arbre de décision parcourt les branches jusqu'à ce qu'il atteigne une feuille, qui représente une prédiction ou une décision finale. Une forêt aléatoire est une collection d'arbres de décision indépendants, où chaque arbre est entraîné sur un échantillon aléatoire du jeu de données. Les prédictions finales sont obtenues en agrégeant les prédictions de chaque arbre. Cela permet d'obtenir des prédictions plus robustes et de réduire le sur-ajustement [54].

### 3.3.3 Classificateur Non-Supervisé utilisés

Les classificateurs non supervisés sont des algorithmes d'apprentissage automatique qui visent à organiser ou regrouper des données sans l'utilisation de labels ou de catégories prédéfinies. Contrairement aux classificateurs supervisés qui apprennent à partir de données étiquetées, les classificateurs non supervisés explorent la structure inhérente aux données pour trouver des similarités, des relations ou des motifs. Ces algorithmes peuvent être utilisés pour la segmentation des données, la détection d'anomalies, la réduction de dimension, la recommandation et d'autres tâches où la connaissance préalable des catégories n'est pas obligatoire.

## K-means

K-means est un algorithme de regroupement non supervisé qui divise un ensemble de données en clusters ou groupes homogènes. Il cherche à minimiser la distance entre les points de données au sein d'un même cluster. Les étapes principales de l'algorithme consistent à initialiser les centres de cluster, assigner les points de données aux clusters les plus proches, mettre à jour les centres de cluster et répéter jusqu'à convergence. K-means est largement utilisé dans divers domaines pour la segmentation et l'analyse de données [55].

## BIRCH

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) est un algorithme de regroupement hiérarchique basé sur la croissance de l'arbre. Il vise à détecter les clusters dans un ensemble de données en utilisant une structure d'arbre appelée CF-tree (Clustering Feature Tree). L'algorithme BIRCH effectue une réduction progressive de la taille de l'ensemble de données en utilisant des sous-clusters et des vecteurs de centre de gravité, ce qui permet une exploration plus efficace de l'espace des données. Il est efficace pour traiter de grands ensembles de données et est souvent utilisé dans des applications nécessitant une détection rapide de clusters [56].

## Agglomerative Clustering

AgglomerativeClustering est un algorithme de regroupement faisant partie de la bibliothèque scikit-learn, utilisée couramment en apprentissage automatique avec Python. Il s'agit d'une méthode de regroupement hiérarchique visant à regrouper ensemble des points de données similaires en fonction de leur proximité ou similarité. En commençant par chaque point de données comme un cluster individuel, l'algorithme fusionne de manière itérative les clusters en se basant sur un critère de liaison spécifié [57].

### 3.4 METHODOLOGIE

La figure 12 représente les différentes étapes de notre approche :

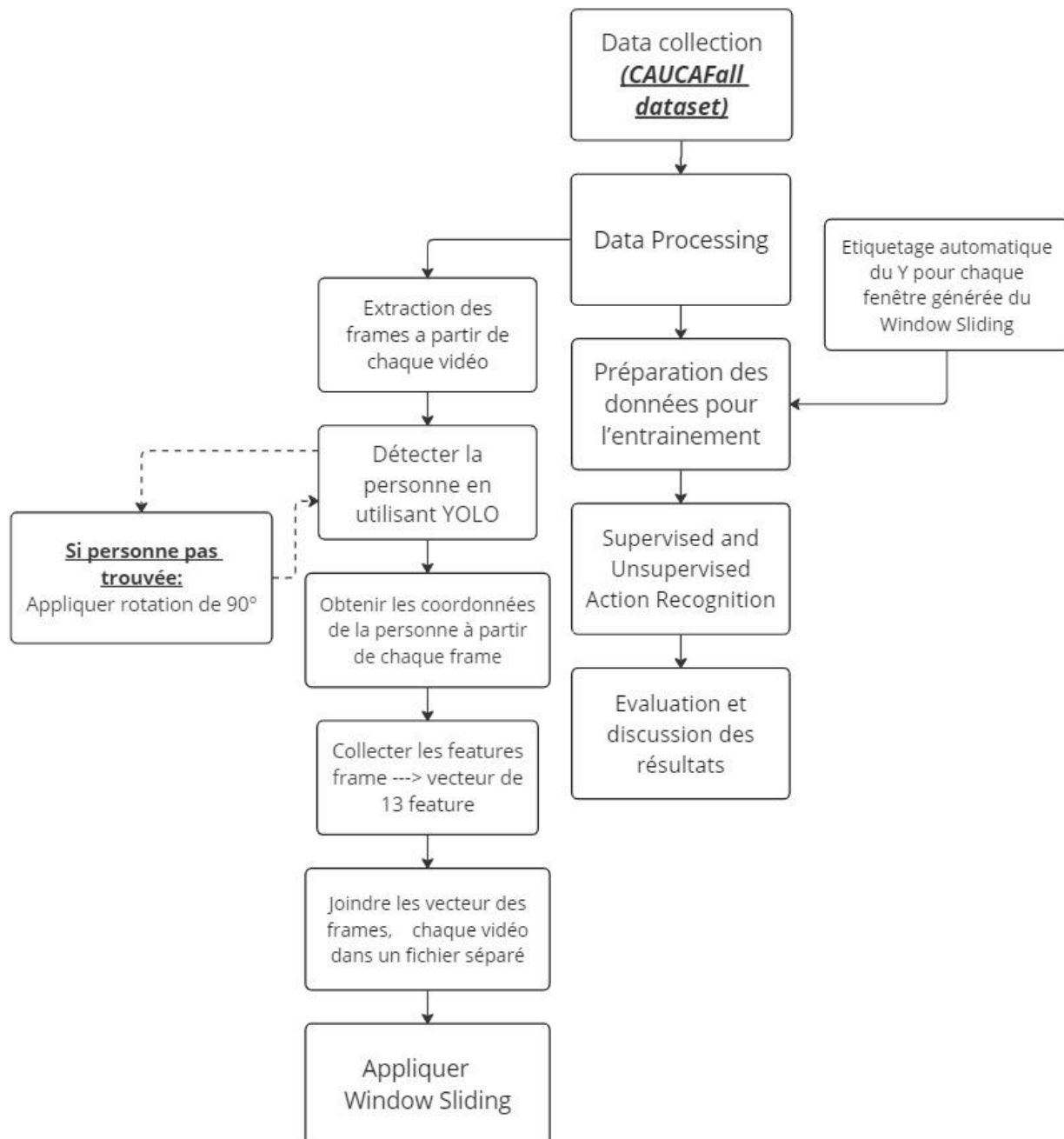


Figure 12: Organigramme de notre Approche

### 3.4.1 Collection des données (choix de l'ensemble de données)

#### C'est quoi un ensemble de données (dataset) ?

Un dataset, ou ensemble de données, est une collection structurée de données organisées dans un format spécifique. Il s'agit d'un regroupement d'informations relatives à un domaine particulier, telles que des observations, des mesures ou des enregistrements. Un dataset peut contenir différents types de données, tels que des textes, des images, des séquences vidéos, etc. [58].

#### Pourquoi utiliser un dataset ?

Les datasets jouent un rôle crucial dans l'apprentissage automatique. Ils fournissent les exemples nécessaires pour entraîner les modèles d'apprentissage, leur permettant d'apprendre à effectuer des tâches spécifiques, ainsi ils servent de base de données à partir desquelles les modèles peuvent apprendre des modèles de données existants et faire des prédictions ou prendre des décisions sur de nouvelles données.

Un bon dataset doit être représentatif et diversifié, fournissant des exemples pertinents et variés du domaine étudié. Il doit également être annoté correctement. De plus, les datasets sont utilisés pour évaluer les performances des modèles sur des données de test, mesurant ainsi leur précision et leur capacité à généraliser.

Le choix du dataset est donc une décision cruciale, car il peut influencer la qualité des résultats obtenus. Il est important de considérer la taille, la qualité et la pertinence du dataset par rapport à la tâche spécifique.

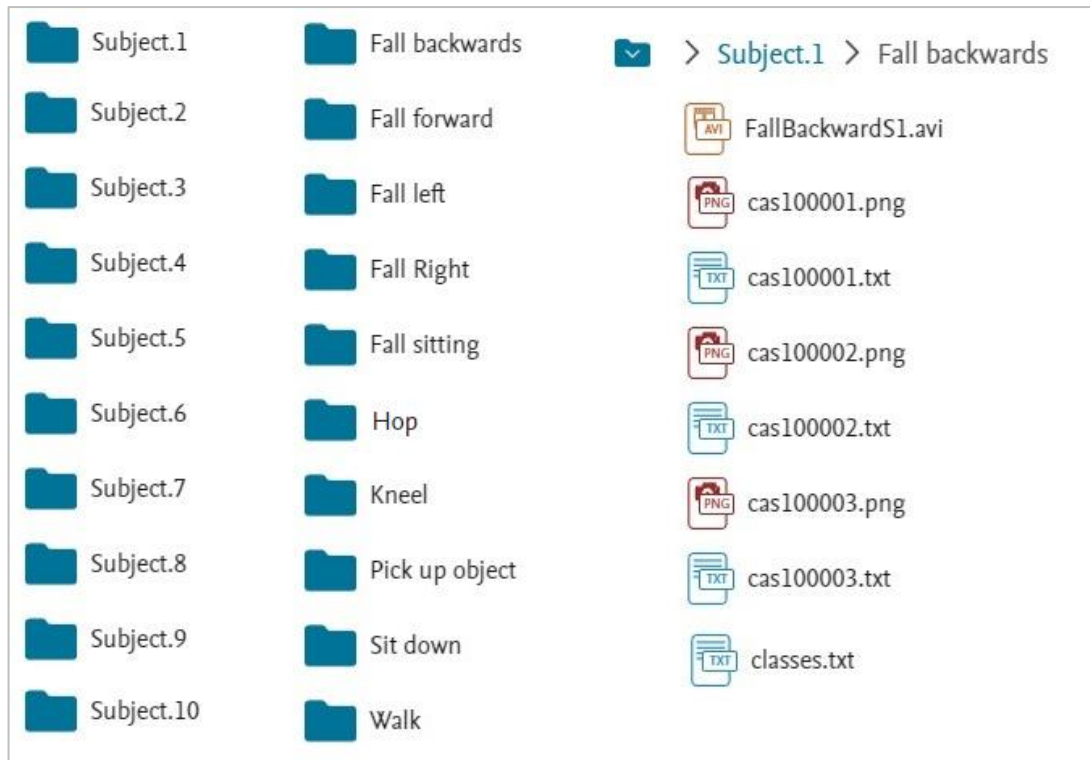
Les séquences vidéo réelles d'instances de chute disponibles au public sont rares, ce qui rend la création de données simulées la seule alternative réalisable. Cependant, il est difficile de simuler de manière réaliste des chutes en respectant les mesures de sécurité, ce qui entraîne une pénurie de données de 'chutes' dans la plupart des datasets simulées disponibles, pour notre travail, nous avons choisi le dataset CAUCAFall.

## CAUCAFall

CAUCAFall est une base de données créée par l'université du Cauca a Popayán, Cauca en Colombie, dans le but de reconnaître les chutes humaines, sa caractéristique distinctive réside dans le fait qu'elle est créée dans des conditions d'un environnement domestique non contrôlé, avec des occultations, des variations d'éclairage (naturel, artificiel et nocturne), une variété de vêtements des participants, des mouvements en arrière-plan, différentes textures sur le sol et dans la pièce, une variété d'angles de chute, différentes distances entre la caméra et la chute, avec des participants de différents âges, poids, taille et même différentes jambes dominantes, ce qui contribue aux progrès réels de la recherche dans la reconnaissance des chutes. Les sujets ont simulé 5 types de chutes et 5 types d'activités de la vie quotidienne (AVQ ou AVL pour Activities of Daily Living). Les données incluent des chutes en avant, des chutes en arrière, des chutes à gauche, des chutes à droite et des chutes survenant en se levant. Les AVQ réalisées par les participants sont : marcher, sauter, ramasser un objet, s'asseoir et enfin s'agenouiller [59].

### Structure du dataset CAUCAFall

Les données sont organisées en 10 répertoires principaux correspondant aux sujets, chacun contenant 10 dossiers avec les différentes activités réalisées par les participants comme montrer dans la figure 13. Dans chaque dossier se trouve une vidéo de l'action au format .avi, les images des actions au format .png, et chacune des étiquettes de segmentation des images au format .txt, Cependant nous n'utiliserons aucune données fournit en dehors des vidéos dans ce travail, car nous extrairons nos propres caractéristiques (features) [59].



*Figure 13: CAUCAFall Dataset Folder Structure [59]*



### 3.4.2 Data Processing Frame Extraction

Nous avons commencé en transformant chaque vidéo en plusieurs frames (images); (20 frames par seconde) en utilisant la bibliothèque OpenCv, que nous passerons après dans le modèle YOLO

1 vidéo = (25\*nombre de secondes) frames.

### Détecter la personne en utilisant YOLO

Nous passons ensuite la totalité de nos frames dans le modèle de détection d'objet YOLO comme input, l'output devrait être de nouveaux frames avec la représentation d'un Bounding Box (rectangle lié) au tour de l'humain, accompagné d'une valeur qu'on appelle valeur de confiance qui représente le degré de certitude que l'objet détecté est belle et bien un humain.

YOLO dispose d'un ensemble de modèles préalablement entraîné à l'aide du Dataset MS COCO [60]. Ces modèles préconstruits sont capables de détecter des objets comme des personnes, des voitures, des vélos, des chiens, des chats, des avions, des bateaux, etc...



*Figure 14: Image (frame) avant application de YOLO*

La figure 14 représente un exemple d'un frame avant de le passer par le modèle YOLO.

Si nous ne précisons pas de classe spécifique à détecter, le modèle YOLO détectera toutes les classes dans l'image (frame), comme le montre la figure 15.



*Figure 15: Image (frame) après application du modèle YOLO*

Après l'exécution du modèle YOLO pour détecter les objets, nous extrayons des caractéristiques spécifiques aux cadres englobants de la classe "Personne", qui est la classe dont l'ID est égale à 0. Ces caractéristiques comprennent les coordonnées des deux points qui délimitent le box, à partir desquelles nous déduisons la longueur, la largeur du cadre, ainsi que les coordonnées x et y de son centre de gravité.

Cependant, c'est à ce stade qu'on a reconcentré notre 1<sup>er</sup> obstacle de cette étude, le modèle de détection d'objet YOLO, a détecté de manière parfaite certaines vidéos (en ne ratant aucune image (frame), donc pas d'existence de frames sans une instance de la classe 'Personne' reconnu), mais d'autre part, pour d'autres vidéos, YOLO ne détecte pas la personne dans quasiment la moitié des frames de la vidéo.

L'échec de YOLO à détecter l'humain dans ces frames peut être dû à de nombreuses raisons, que nous détaillerons dans la section Limitations Lors de l'utilisation de YOLO.

C'est exactement là où notre première approche a vu le jour. Afin de remédier au problème des humains (Personnes) non détectées, nous avons proposé de faire une rotation chaque fois que le frame ne contient pas une instance de la classe 0 (classe 'personne'), si la personne n'est toujours pas détectée, nous appliquons une nouvelle rotation jusqu'à 3 rotations dans le sens anti-horaire.

Les résultats de cette approche se sont avérés très intéressants, et ils nous ont permis de mieux comprendre le mécanisme de fonctionnement de YOLO, ce que nous aborderons dans la prochaine section.

## **Limitations lors de l'utilisation de YOLO**

### **Contexte Temporel**

Cette Limitation réside dans l'incapacité de YOLO à conserver en mémoire le contexte temporel des images précédentes lors de la détection d'objets. YOLO traite chaque image indépendamment et ne prend pas en compte les informations provenant des images précédentes d'une séquence vidéo, cela peut poser des défis dans les scénarios où la continuité temporelle est importante, tels que la détection d'objets en mouvement ou la reconnaissance d'actions humaines. Sans la capacité de mémoriser les informations temporelles, YOLO peut avoir du mal à suivre les objets d'une image à une autre, en particulier lorsque ceux-ci subissent des changements rapides ou des occlusions, cette limitation temporelle peut entraîner des erreurs de détection, des faux positifs ou des faux négatifs, car YOLO ne peut pas exploiter les informations du passé pour améliorer la précision de la détection.

### **Certain frames ont été ratés**

Après le processus d'extraction des images (frames), nous avons obtenu un total de 19 410 images. Parmi celles-ci, nous avons remarqué avec surprise que dans 2 858 images, YOLO n'a pas détecté des objets de la classe « personnes ». Cela représente environ 15 % du nombre total d'images extraites, parmi les raisons pour lesquels cela peut se produire :

## **Chevauchement/Occlusion des Objets**

Lorsqu'un objet est partiellement caché ou occulté par un autre objet, ou lorsque plusieurs objets se chevauchent dans une image, YOLO peut avoir du mal à les détecter individuellement et correctement, cela peut entraîner une fusion des objets détectés en un seul objet ou une localisation incorrecte des objets. Cette limitation peut entraîner une confusion dans l'identification des objets et une baisse de précision dans la détection.

Ces limitations sont principalement dues à la méthode de détection utilisée par YOLO, qui consiste à diviser l'image en une grille et à prédire des boîtes englobantes pour chaque cellule de la grille. Lorsque des chevauchements et des occlusions se produisent, il devient difficile pour YOLO de localiser et de distinguer les objets individuellement

## **Conditions de qualité d'image, d'éclairage et de colorisation**

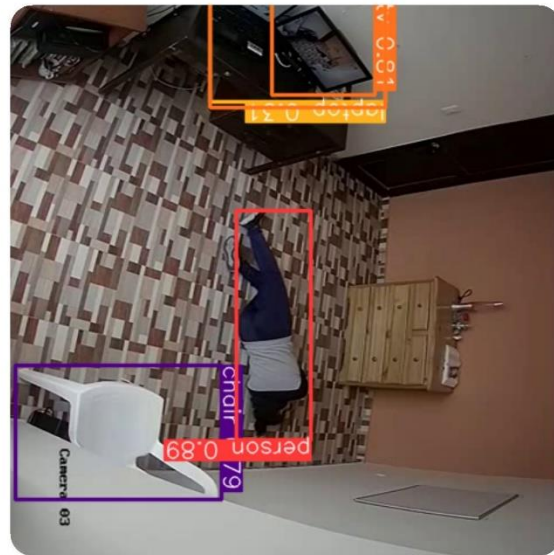
Les limitations de YOLO en termes de qualité d'image, de conditions d'éclairage et de couleurs peuvent affecter la précision de sa détection des objets. Une mauvaise qualité d'image, telle qu'une faible résolution, peut entraîner une perte de détails et altérer la capacité de YOLO à détecter avec précision. Les variations d'éclairage, comme les ombres ou les reflets ou les éclairages inégaux, ainsi que les variations de couleurs importantes, tels que les variations de teints et de saturation peuvent également compromettre la performance de YOLO.

## **Application de notre approche**

Comme solution au problème des frames ratés de la part de YOLO, nous avons proposé une nouvelle approche qui consiste à faire des rotations sur les frames ratées et les repasser dans YOLO, la rotation sur les 4 axes a montré des résultats excellents et très pertinents en ce qui concerne la modalité de fonctionnement du modèle YOLO, comme nous le voyons dans la figure 16, le modèle YOLO n'a pas reconnu l'objet (la personne) dans l'état normal du frame, ce n'est qu'après avoir appliqué une rotation que YOLO l'a détecté, cette instance c'est reproduite sur la plupart des frames ratés.



Frame 1 : sans rotation



Frame 4 : Rotation à 270° CCW



Frame 2 : Rotation à 90° CCW



Frame 3 : Rotation à 180° CCW

*Figure 16: Exemple d'instance où l'application des rotations sur le même frame a permis de détecter la personne.*



Nous avons procédé à appliquer ces rotations sur la totalité des frames, afin d'en extraire encore plus de statistiques, nous avons représenté ces résultats par un vecteur binaire additionnel de 4 valeurs attribuées à chaque frame, les 4 valeurs représentent les 4 rotations (rotation à 0°, rotation à 90°, rotation à 180° et rotation à 270° dans le sens antihoraire respectueusement), les valeurs possibles sont 1 et 0 pour détection de la classe 'Personne' et absence de détection respectueusement.

Par la suite chaque frame est représenté par un vecteur de 13 éléments comme le montre la Figure 17.

1	X1	Y1	X2	Y2	Larg	Long	r (Ratio)	Cx	Cy	N	CCW	I	CW
2	297.73696	41.582904	379.58142	283.81420	81.844451	242.23130	0.3378772	338.65919	162.69855	1	0	1	0
3	296.89855	41.227592	384.75640	283.74005	87.857849	242.51245	0.3622817	340.82748	162.48382	1	0	0	0
4	297.83004	42.746356	382.02542	284.42117	84.195373	241.67481	0.3483828	339.92773	163.58376	1	0	0	0
5	297.74655	43.398403	380.27853	284.13653	82.531982	240.73812	0.3428288	339.01254	163.76746	1	0	0	0
6	300.02319	42.915966	379.67675	284.35784	79.653564	241.44187	0.3299078	339.84997	163.63690	1	1	0	0

Figure 17: exemple de vecteur de caractéristiques (Features)

Avec :

X1 : abscisse du 1<sup>er</sup> point (le plus haut, à gauche) du box de classe1 (humain),

Y1 : ordonnée du 1<sup>er</sup> point (le plus haut, à gauche) du box de classe1 (humain),

X2 : abscisse du 2<sup>ème</sup> point (le plus bas, à droite) du box de classe1 (humain),

Y2 : ordonnée du 2<sup>ème</sup> point (le plus bas, à droite) du box de classe1 (humain),

Larg: largeur du box de classe 1 (humain),

Long : longueur du box de classe 1 (humain),

r (Ratio) : rapport entre largeur et longueur du box englobant la personne  $r = Larg \div Long$ ,

Cx : abscisse du centre de gravité du box de classe1 (humain),

Cy : abscisse du centre de gravité du box de classe1 (humain),

N : 1 si humain est reconnue dans le frame initiale (sans rotation), 0 sinon.

CCW : 1 si humain est reconnue dans le frame après rotation de 90° à gauche, 0 sinon.

I : 1 si humain est reconnue dans le frame après rotation de 180° à gauche, 0 sinon.

CW : 1 si humain est reconnue dans le frame après rotation de 270° à gauche, 0 sinon.

## Application du Window Sliding

L'une des dernières étapes lors de la préparation des données c'est d'effectuer du Window Sliding à notre ensemble de données.

La technique de fenêtre glissante (window sliding) est une méthode utilisée dans le traitement de données vectorielles pour analyser une séquence de données en découpant cette séquence en fenêtres de taille fixe et en les faisant glisser à travers la séquence avec un certain pas de décalage. Chaque fenêtre est ensuite traitée individuellement pour extraire des caractéristiques ou effectuer des opérations spécifiques. Cette approche permet d'explorer l'ensemble de la séquence de données de manière itérative et d'obtenir des informations locales à partir de chaque fenêtre [61].

Pour appliquer Window Sliding, nous devons définir un paramètre (que nous considérons variable dans notre code), ce paramètre est la taille de la fenêtre. Après plusieurs tests et observations, nous remarquons que la longueur moyenne pour que la personne tombe est approximativement 1 seconde, donc nous avons défini la taille de la fenêtre (L) égale à 20 (20 frames approximativement 1 seconde).

### 3.4.3 Préparation des données pour l'entraînement

Pour la classification supervisée, nous avons besoin de définir le Y.

Nous allons proposer une nouvelle approche pour la détermination automatique du Y à partir d'un ratio  $r = \text{largeur} / \text{longueur}$  définit parmi les features (voir figure 17).

#### Le ratio r

Nous allons utiliser les features de la largeur et la longueur du box englobant la personne, lorsqu'elle est debout, sa hauteur est significativement plus importante que sa largeur selon l'équation suivante :  $r = \text{Larg} \div \text{Long}$ .



*Figure 18: Personne en position Normale (NOFALL)*

Nous remarquons que lorsqu'en position debout (pas de chute), la hauteur du box englobant la personne est plus importante que sa largeur, ainsi la valeur de  $r$  sera inférieure à 1.

Lorsque le sujet s'accroupie ou s'assoit, le box rectangulaire devient carré, signifiant ainsi une vraisemblance entre la valeur de la largeur et de la longueur (valeur de  $r$  avoisine 1), comme le montre la figure 19.



*Figure 19: Personne en position Anormale (Accroupi)*



La figure suivante montre un cas évident de chute sur lequel nous remarquons que la valeur de la largeur est maintenant la plus élevée (valeur de  $r$  supérieure à 1).



*Figure 20: Personne en position Anormale (Chute)*

Après plusieurs tests nous avons pu déterminer la valeur de 0.55 comme seuil optimale, ainsi :

Si  $r > 0.55 \rightarrow$  Personne en cas de chute (FALL) sinon pas de chute (NO FALL)

## Détermination de la valeur de Y

Nous proposons une approche de détermination automatique de la valeur de Y, basée sur la valeur du ratio  $r$ .

Chaque fenêtre glissante générée aura 20 valeurs de  $r$  (ratio), pour définir la valeur de Y de chaque fenêtre nous calculons le nombre de frame pour lesquels la valeur de  $r$  est inférieure à 0.55 ( $r < 0.55$ ), si ce nombre excède 75% de la taille de la fenêtre (donc 15 valeurs), la valeur attribuée à Y devient 1 (donc la fenêtre est annotée comme fenêtre qui représente une chute 'FALL'), sinon Y est égale à 0 (fenêtre qui représente le cas de non chute 'NOFALL').

### 3.4.4 Reconnaissance d'actions Supervisé et Non Supervisé

Tout d'abord nous divisons notre enjeu de données en 2 parties, 70% des vidéos de notre dataset seront utilisées pour l'entraînement (training) des modèles et les 30% restantes seront allouées aux tests.

Nous avons utilisé une multitude de modèles tel que KNN, GBC et XGBC, abordés en détaille au début du chapitre.

## 3.5 Résultats et Discussion

### 3.5.1 Métriques d'évaluation

**Accuracy :** L'exactitude (accuracy) est une mesure couramment utilisée pour évaluer les performances d'un modèle de classification. Elle représente le pourcentage de prédictions correctes parmi l'ensemble total des prédictions effectuées par le modèle [62].

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

**Precision:** est le rapport entre les vrais positifs et les vrais positifs sur le terrain (prédictions correctes totales). [62]

$$\text{Précision} = \text{TP} / (\text{TP} + \text{FP})$$

La moyenne de toutes les précisions moyennes est appelée précision moyenne moyenne (mAP).

**Recall :** est le rapport entre les vrais positifs et la prédiction positive totale (correcte ou incorrecte) [62].

**f1-score :** Le F1-score est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus.

$$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

**Support :** nombre d'éléments dans la classe.

### 3.5.2 Résultats des classificateurs supervisés

#### Avec Shuffle=vrai (true)

Lorsque le paramètre "shuffle" est vrai, les échantillons d'entraînement sont réorganisés de manière aléatoire afin d'introduire de la variabilité dans l'ordre de présentation des données. Cela peut être bénéfique pour éviter tout biais potentiel qui pourrait être introduit si les données étaient présentées dans un ordre spécifique.

*Tableau 1: Résultats 1 KNN*

KNN	precision	Recall	f1-score	support
<b>0</b>	0.96	0.97	0.96	2071
<b>1</b>	0.98	0.98	0.98	3182
<b>Accuracy</b>	-	-	0.97	5253

KNN : L'algorithme KNN (K-Nearest Neighbors) a une 'accuracy' de 0.97, une précision de 0,96 pour la classe 0 et de 0,98 pour la classe 1. Le rappel est de 0,97 pour la classe 0 et de 0,98 pour la classe 1. Le score F1 est de 0,96 pour la classe 0 et de 0,98 pour la classe 1. Ces valeurs indiquent une performance élevée de l'algorithme KNN.

*Tableau 2: Résultats 1 GBC*

GBC	Precision	Recall	f1-score	support
<b>0</b>	0.94	0.95	0.95	2071
<b>1</b>	0.97	0.96	0.97	3182
<b>Accuracy</b>	-	-	0.96	5235

GBC : L'algorithme GBC (Gradient Boosting Classifier) a une 'accuracy' de 0.96, une précision de 0,94 pour la classe 0 et de 0,97 pour la classe 1. Le rappel est de 0,95 pour la classe 0 et de 0,96 pour la classe 1. Le score F1 est de 0,95 pour la classe 0 et de 0,97 pour la classe 1. Ces résultats montrent une performance élevée de l'algorithme GBC.

*Tableau 3: Résultats 1 XGBC*

<b>XGBC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.98	0.99	0.98	2071
<b>1</b>	0.99	0.99	0.99	3182
<b>Accuracy</b>	-	-	0.99	5235

XGBC : L'algorithme XGBC (XGBoost Classifier) a une 'accuracy' de 0.99, une précision de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Le rappel est de 0,99 pour la classe 0 et de 0,99 pour la classe 1. Le score F1 est de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Ces résultats indiquent une performance élevée de l'algorithme XGBC.

*Tableau 4: Résultats 1 HGBC*

<b>HGBC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.99	0.98	0.98	2071
<b>1</b>	0.99	0.99	0.99	3182
<b>Accuracy</b>	-	-	0.99	5235

HGBC : L'algorithme HGBC (Histogram Gradient Boosting Classifier) a une 'accuracy' de 0.99, une précision de 0,99 pour la classe 0 et de 0,99 pour la classe 1. Le rappel est de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Le score F1 est de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Ces résultats montrent une performance élevée de l'algorithme HGBC.

*Tableau 5: Résultats 1 RFC*

<b>RFC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.98	0.99	0.98	2071
<b>1</b>	0.99	0.99	0.99	3182
<b>Accuracy</b>	-	-	0.99	5253

RFC : L'algorithme RFC (Random Forest Classifier) a une 'accuracy' de 0.99, une précision de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Le rappel est de 0,99 pour la classe

0 et de 0,99 pour la classe 1. Le score F1 est de 0,98 pour la classe 0 et de 0,99 pour la classe 1. Ces résultats indiquent une performance élevée de l'algorithme RFC.

*Tableau 6: Résultats 1 DTC*

<b>DTC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.9	0.94	0.92	2071
<b>1</b>	0.96	0.93	0.94	3182
<b>Accuracy</b>	-	-	0.93	5253

DTC : L'algorithme DTC (Decision Tree Classifier) a une 'accuracy' de 0.93, une précision de 0,90 pour la classe 0 et de 0,96 pour la classe 1. Le rappel est de 0,94 pour la classe 0 et de 0,93 pour la classe 1. Le score F1 est de 0,92 pour la classe 0 et de 0,94 pour la classe 1. Ces résultats montrent une performance légèrement inférieure de l'algorithme DTC par rapport aux autres.

*Tableau 7: Résultats 1 ABC*

<b>ABC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.94	0.95	0.94	2071
<b>1</b>	0.96	0.96	0.96	3182
<b>Accuracy</b>	-	-	0.96	5253

ABC : L'algorithme ABC (AdaBoost Classifier) a une 'accuracy' de 0.96, une précision de 0,94 pour la classe 0 et de 0,96 pour la classe 1. Le rappel est de 0,95 pour la classe 0 et de 0,96 pour la classe 1. Le score F1 est de 0,94 pour la classe 0 et de 0,96 pour la classe 1. Ces résultats montrent une performance élevée de l'algorithme ABC.

## Avec Shuffle=faux (false)

*Tableau 8 : Résultats 2 KNN*

KNN	Precision	Recall	f1-score	support
<b>0</b>	0.87	0.9	0.89	2991
<b>1</b>	0.87	0.83	0.85	2262
<b>Accuracy</b>	-	-	0.87	5253

KNN : L'algorithme KNN (K-Nearest Neighbors) a une 'accuracy' de 0.87, une précision de 0,87 pour la classe 0 et de 0,87 pour la classe 1. Le rappel est de 0,90 pour la classe 0 et de 0,83 pour la classe 1. Le score F1 est de 0,89 pour la classe 0 et de 0,85 pour la classe 1. Ces résultats indiquent une performance légèrement inférieure de l'algorithme KNN par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 9: Résultats 2 GBC*

GBC	Precision	Recall	f1-score	support
<b>0</b>	0.92	0.92	0.92	2991
<b>1</b>	0.9	0.89	0.89	2262
<b>Accuracy</b>	-	-	0.91	5253

GBC : L'algorithme GBC (Gradient Boosting Classifier) a une 'accuracy' de 0.91, une précision de 0,92 pour la classe 0 et de 0,90 pour la classe 1. Le rappel est de 0,92 pour la classe 0 et de 0,89 pour la classe 1. Le score F1 est de 0,92 pour la classe 0 et de 0,89 pour la classe 1. Ces résultats montrent une performance globalement similaire de l'algorithme GBC par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 10: Résultats 2 XGBC*

<b>XGBC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.96	0.96	0.96	2991
<b>1</b>	0.94	0.95	0.95	2262
<b>Accuracy</b>	-	-	0.96	5253

XGBC : L'algorithme XGBC (XGBoost Classifier) a une 'accuracy' de 0.96, une précision de 0,96 pour la classe 0 et de 0,94 pour la classe 1. Le rappel est de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Le score F1 est de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Ces résultats montrent une performance globalement similaire de l'algorithme XGBC par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 11: Résultats 2 HGBC*

<b>HGBC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.96	0.96	0.96	2991
<b>1</b>	0.95	0.95	0.95	2262
<b>Accuracy</b>	-	-	0.96	5253

HGBC : L'algorithme HGBC (Histogram Gradient Boosting Classifier) a une 'accuracy' de 0.96, une précision de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Le rappel est de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Le score F1 est de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Ces résultats montrent une performance globalement similaire de l'algorithme HGBC par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 12: Résultats 2 RFC*

<b>RFC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.97	0.95	0.96	2991
<b>1</b>	0.93	0.96	0.95	2262

<b>Accuracy</b>	-	-	0.95	5253
-----------------	---	---	------	------

RFC : L'algorithme RFC (Random Forest Classifier) a une 'accuracy' de 0.95, une précision de 0,97 pour la classe 0 et de 0,93 pour la classe 1. Le rappel est de 0,95 pour la classe 0 et de 0,96 pour la classe 1. Le score F1 est de 0,96 pour la classe 0 et de 0,95 pour la classe 1. Ces résultats montrent une performance légèrement inférieure de l'algorithme RFC par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 13: Résultats 2 DTC*

<b>DTC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.87	0.94	0.9	2991
<b>1</b>	0.91	0.82	0.86	2262
<b>Accuracy</b>	-	-	0.89	5253

DTC : L'algorithme DTC (Decision Tree Classifier) a une 'accuracy' de 0.89, une précision de 0,87 pour la classe 0 et de 0,91 pour la classe 1. Le rappel est de 0,94 pour la classe 0 et de 0,82 pour la classe 1. Le score F1 est de 0,90 pour la classe 0 et de 0,86 pour la classe 1. Ces résultats montrent une performance légèrement inférieure de l'algorithme DTC par rapport aux résultats obtenus lorsque "shuffle=True".

*Tableau 14: Résultats 2 ABC*

<b>ABC</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.93	0.93	0.93	2991
<b>1</b>	0.9	0.91	0.91	2262
<b>Accuracy</b>	-	-	0.92	5253

ABC : L'algorithme ABC (AdaBoost Classifier) a une 'accuracy' de 0.92, une précision de 0,93 pour la classe 0 et de 0,90 pour la classe 1. Le rappel est de 0,93 pour la classe 0 et de 0,91 pour la classe 1. Le score F1 est de 0,93 pour la classe 0 et de 0,91 pour la classe 1. Ces



résultats montrent une performance globalement similaire de l'algorithme ABC par rapport aux résultats obtenus lorsque "shuffle=True".

### 3.5.3 Résultats des classificateurs non-supervisés

#### Avec Shuffle=vrai (true)

*Tableau 15: Résultats 1 BIRCH*

<b>BIRCH</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.57	0.41	0.47	2094
<b>1</b>	0.67	0.79	0.73	3159
<b>Accuracy</b>	-	-	0.64	5253

BIRCH : Lorsque "Shuffle=True", l'algorithme BIRCH a une 'accuracy' de 0.64, une précision de 0,57 pour la classe 0 et de 0,67 pour la classe 1. Le rappel est de 0,41 pour classe 0 et 0,79 pour classe 1. Le score F1 est de 0,47 pour la classe 0 et de 0,73 pour la classe 1. Ces valeurs indiquent une performance modérée de l'algorithme.

*Tableau 16: Résultats 1 K-means*

<b>KMS</b>	<b>Precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>support</b>
<b>0</b>	0.32	0.59	0.41	2037
<b>1</b>	0.44	0.20	0.28	3216
<b>Accuracy</b>	-	-	0.35	5253

K-Means : Avec "Shuffle=True", le K-Means a une 'accuracy' de 0.35, une précision de 0,32 pour la classe 0 et de 0,44 pour la classe 1. Le rappel est de 0,59 pour la classe 0 et de

0,20 pour la classe 1. Le score F1 est de 0,41 pour la classe 0 et de 0,28 pour la classe 1. Ces résultats montrent une performance inférieure de l'algorithme par rapport à BIRCH.

*Tableau 17: Résultats 1 AggC*

AggC	Precision	Recall	f1-score	support
<b>0</b>	0.48	0.74	0.58	2045
<b>1</b>	0.75	0.49	0.28	3208
<b>Accuracy</b>	-	-	0.59	5253

Agglomerative Clustering : Avec "Shuffle=True", l'Agglomerative Clustering a une 'accuracy' de 0.59, une précision de 0,48 pour la classe 0 et de 0,75 pour la classe 1. Le rappel est de 0,74 pour la classe 0 et de 0,49 pour la classe 1. Le score F1 est de 0,58 pour la classe 0 et de 0,60 pour la classe 1. Ces résultats indiquent une performance (notamment la précision) relativement meilleure de l'algorithme par rapport à BIRCH et K-Means.

### Avec Shuffle=faux (false)

*Tableau 18: Résultats 2 BIRCH*

BIRCH	Precision	Recall	f1-score	support
<b>0</b>	0.48	0.57	0.52	2991
<b>1</b>	0.24	0.18	0.20	2262
<b>Accuracy</b>	-	-	0.40	5253

BIRCH : Avec "Shuffle=False", l'algorithme BIRCH a une 'accuracy' de 0.40, une précision de 0,48 pour la classe 0 et de 0,24 pour la classe 1. Le rappel est de 0,57 pour la classe 0 et de 0,18 pour la classe 1. Le score F1 est de 0,52 pour la classe 0 et de 0,20 pour la classe 1. Ces résultats montrent une performance globalement inférieure à celle obtenue lorsque "Shuffle=True".

Tableau 19: Résultats 2 K-means

KMS	Precision	Recall	f1-score	support
<b>0</b>	0.76	0.43	0.55	2991
<b>1</b>	0.52	0.82	0.64	2262
<b>Accuracy</b>	-	-	0.60	5253

K-Means : Avec "Shuffle=False", le K-Means a une 'accuracy' de 0.60, une précision de 0,76 pour la classe 0 et de 0,52 pour la classe 1. Le rappel est de 0,43 pour la classe 0 et de 0,82 pour la classe 1. Le score F1 est de 0,55 pour la classe 0 et de 0,64 pour la classe 1. Ces résultats montrent une performance relativement meilleure de l'algorithme par rapport à BIRCH lorsque "Shuffle=False".

Tableau 20: Résultats 2 AggC

AggC	Precision	Recall	f1-score	support
<b>0</b>	0.50	0.57	0.53	2991
<b>1</b>	0.31	0.25	0.28	2262
<b>Accuracy</b>	-	-	0.43	5253

AgglomerativeClustering : Avec "Shuffle=False", l'AgglomerativeClustering a une 'accuracy' de 0.43, une précision de 0,50 pour la classe 0 et de 0,31 pour la classe 1. Le rappel est de 0,57 pour la classe 0 et de 0,25 pour la classe 1. Le score F1 est de 0,53 pour la classe 0 et de 0,28 pour la classe 1. Ces résultats montrent une performance modérée de l'algorithme, mais inférieure à celle obtenue lorsque "Shuffle=True".

Voici le résumé des résultats :

*Tableau 21: Résumé des résultats*

Supervised & Unsupervised Models	Accuracy with shuffle=True	Accuracy with shuffle=False
<b>KNN</b>	0.97	0.87
<b>GBC</b>	0.96	0.91
<b>XGBC</b>	0.99	0.96
<b>HGBC</b>	0.99	0.96
<b>RFC</b>	0.99	0.95
<b>DTC</b>	0.93	0.89
<b>ABC</b>	0.96	0.92
<b>BIRCH</b>	0.64	0.40
<b>KMS</b>	0.35	0.60
<b>AggC</b>	0.59	0.43

Parmi les algorithmes de classification évalués avec les paramètres "shuffle=True" et "shuffle=False", les deux meilleurs algorithmes en termes de performances générales sont XGBC (XGBoost Classifier) et HGBC (Histogram Gradient Boosting Classifier). Ces deux algorithmes ont montré des performances élevées dans les deux configurations et ont obtenu des scores de précision, de rappel et de score F1 élevés pour les deux classes. Leurs performances globales ont été cohérentes et robustes, ce qui suggère qu'ils sont bien adaptés à la tâche de classification pour les données spécifiques considérées.

### 3.6 Perspectives pour le future

Dans le contexte de la détection de chutes, plusieurs perspectives d'avenir peuvent être envisagées pour améliorer les performances et l'applicabilité de notre travail. Tout d'abord, il serait intéressant d'explorer l'utilisation de capteurs supplémentaires, tels que des capteurs de pression, des capteurs d'inclinaison, des capteurs d'accélération ou des gyroscopes, l'intégration de ces capteurs dans le système de détection permettrait de capturer des informations supplémentaires sur les mouvements et les postures du corps, offrant une meilleure

compréhension du contexte et une détection plus fiable des chutes, ainsi qu'une réduction des faux positifs.

Ensuite, il serait bénéfique d'élargir la portée du système de détection de chutes pour prendre en compte différents environnements, dans notre cas, le dataset CAUCAFALL était collecté pour but de simuler un environnement domestique, ainsi, l'exploration de datasets supplémentaires permettra d'entraîner notre modèle et de tester son efficacité dans différents contextes et dans différents environnements, notamment les établissements de soins et de santé, les lieux de travail ou les espaces publics. Chaque environnement présente ses propres défis et contraintes, et adapter le modèle en conséquence permettrait de répondre aux besoins spécifiques de chaque contexte.

Enfin, une perspective prometteuse pour le futur de notre travail serait la collaboration avec des experts du domaine médical, nous pourrions obtenir des informations précieuses sur les comportements à surveiller, sur les signaux d'alerte à prendre en compte et sur les mesures de prévention appropriées. Leur expertise nous permettra de mieux comprendre les besoins spécifiques des patients et de concevoir des modèles mieux adaptés à leurs situations.

### 3.7 Conclusion

Dans ce chapitre, nous avons exploré plusieurs éléments clés de notre étude. Nous avons présenté l'environnement d'exécution et les modèles d'apprentissage automatique utilisés, nous avons ensuite présenté l'enjeu de données utilisé, et nous avons terminé par détailler la méthodologie de notre travail ainsi que les résultats obtenus.

## Conclusion Générale

Ce travail représente notre propre approche pour la reconnaissance d'actions humaine à partir des vidéos, plus précisément la reconnaissance de la chute humaine pour les personnes âgées, en utilisant les techniques de la vision par ordinateur basées sur l'apprentissage profond.

Nous avons commencé par examiner l'état de l'art de la surveillance du comportement humain, mettant en évidence les limites de la surveillance vidéo traditionnelle et l'apport de l'intelligence artificielle dans ce domaine. Nous avons ensuite abordé les différents concepts clés de l'intelligence artificielle en soulignant son rôle essentiel dans la détection des actions humaines.

Nous avons également effectué une revue de la littérature des travaux existants sur la détection de chutes et nous avons détaillé les différents axes liés au contexte de notre étude.

Nous avons ensuite décrit notre méthodologie d'implémentation, en expliquant notre choix de dataset, et en détaillant chaque étape de notre travail ainsi que les difficultés rencontrées dans chacune des étapes.

Nous avons présenté nos résultats, en utilisant différents classificateurs supervisés et non supervisés et en tenant compte du paramètre shuffle. Nous avons également évalué ces résultats en utilisant plusieurs paramètres d'évaluation pour mesurer l'efficacité de notre approche.

En conclusion, ce mémoire propose une nouvelle approche de reconnaissance des actions humaines basée sur la vidéo, et contribue à l'amélioration de la détection de chutes chez les personnes âgées grâce à l'utilisation de la vision par ordinateur. En continuant à explorer de nouvelles approches et en s'appuyant sur les avancées technologiques, il est possible de développer des solutions plus précises et efficaces pour assurer la sécurité et le bien-être des personnes âgées, avec des implémentations prometteuses dans le domaine médical et social.

# Bibliographies

- [1] l'Organisation mondiale de la santé (OMS), Falls (Key facts- 26 April 2021), disponible sur : <<https://www.who.int/news-room/fact-sheets/detail/falls>>
- [2] Organisation mondiale de la Santé (OMS), Stratégies de prévention et de gestion des chutes tout au long de la vie, Organisation mondiale de la Santé, Genève, 2021.
- [3] <https://geekflare.com/fr/regression-vs-classification/>
- [4] Jaber, Wael. (2018). Oral Probatoire Wael JABER - Detection et diagnostic des défaillances des procedes chimiques a l'aide des reseaux neuronaux artificiels v2.0.
- [5] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501311-apprentissage-supervise/>
- [6] <https://algodaily.com/lessons/ml-interview-questions/using-classification-vs-regression>
- [7], Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.
- [8] Unsupervised Learning and Data Clustering | by Sanatan Mishra | [towardsdatascience.com](https://towardsdatascience.com).
- [9] <https://waterprogramming.wordpress.com/2022/03/16/clustering-basics-and-a-demonstration-in-clustering-infrastructure-pathways/>
- [10] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.
- [11] Ahmed, Salhi & Bouktir, Tarek. (2015). CONTRIBUTION A L'OPTIMISATION DE L'ECOULEMENT DE PUISSANCE EN UTILISANT LA LOGIQUE FLOUE ASSOCIEE AUX RESEAUX DE NEURONES ( NEURO-FLOU).
- [12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [13] <https://www.ibm.com/topics/computer-vision>

- [14] Sarma, D., Bhuyan, M.K. Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review. *SN COMPUT. SCI.* 2, 436 (2021).
- [15] <https://www.xenonstack.com/blog/computer-vision-applications>
- [16] <https://www.v7labs.com/blog/computer-vision-applications>
- [17] <https://www.ibm.com/topics/artificial-intelligence-medicine>
- [18] Saleem, G., Bajwa, U.I. & Raza, R.H. Toward human activity recognition: a survey. *Neural Comput & Applic* 35, 4145–4182 (2023).
- [19] Nooruddin, S., Islam, M.M., Sharna, F.A. et al. Sensor-based fall detection systems: a review. *J Ambient Intell Human Comput* 13, 2735–2751 (2022).
- [20] Xu, T., Zhou, Y. & Zhu, J. (2018). New Advances and Challenges of Fall Detection Systems: A Survey. *Applied Sciences*, 8(3).
- [21] Wang, J., Zhang, Z., Li, B., Lee, S. & Sherratt, R. S. (2014). An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE transactions on consumer electronics*, 60(1), 23–29.
- [22] Keskes, Oussema (2020). Système de détection des chutes en ligne. Mémoire de maîtrise électronique, Montréal, École de technologie supérieure..
- [23] Tian, Y., Lee, G.-H., He, H., Hsu, C.-Y. & Katabi, D. (2018). RF-based fall monitoring using convolutional neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3), 1–24.
- [24] Chua, J.-L., Chang, Y. C. & Lim, W. K. (2015). A simple vision-based fall detection technique for indoor video surveillance. *Signal, Image and Video Processing*, 9(3), 623–633.
- [25] Yan, S., Xiong, Y. & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition.



- [26] Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I. & Lindh, T. (2015). Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion. *International Conference on ICT Innovations*, pp. 99–108.
- [27] Ramachandran, A. & Karuppiyah, A. (2020). A Survey on Recent Advances in Wearable Fall Detection Systems. *BioMed Research International*, 2020.
- [28] Le, T.-L., Morel, J. et al. (2014). An analysis on human fall detection using skeleton from Microsoft Kinect. *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, pp. 484–489.
- [29] A. R. Fansdana, A. K. Heikhmakhtiar and S. Mandala, "Real-time Falling Detection System for Elderly using CNN," *2021 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia, 2021, pp. 194-19.
- [30] Hwang, S., Ahn, D., Park, H. & Park, T. (2017). Maximizing accuracy of fall detection and alert systems based on 3D convolutional neural network. *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 343–344.
- [31] Fakhruddin, A. H., Fei, X. & Li, H. (2017). Convolutional neural networks (CNN) based human fall detection on body sensor networks (BSN) sensor data. *2017 4th International Conference on Systems and Informatics (ICSAI)*.
- [32] Wang, X., Zhang, J. & Yan, W.Q. Gait recognition using multichannel convolution neural networks. *Neural Comput & Applic* 32, 14275–14285 (2020).
- [33] Mason, J.E., Traoré, I., Woungang, I. (2016). Applications of Gait Biometrics. In: *Machine Learning Techniques for Gait Biometric Recognition*. Springer, Cham.
- [34] C. Chen et al., "A survey on human behavior analysis in video surveillance," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 180-201, 2019.
- [35] Gavril, D. M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1), 82-98.

- [36] Straka, M., Hauswiesner, S., Rüther, M., & Bischof, H. (2011). Skeletal Graph Based Human Pose Estimation in Real-Time. British Machine Vision Conference.
- [37] Hodge, Victoria, and Jim Austin. "A survey of outlier detection methodologies". Artificial intelligence review 22.2 (2004): 85-126.
- [38] Ke S-R, Thuc HLU, Lee Y-J, Hwang J-N, Yoo J-H, Choi K-H. A Review on Video-Based Human Activity Recognition. Computers. 2013; 2(2):88-131.
- [39] <https://www.futura-sciences.com/tech/definitions/informatique-python-19349/>
- [40] : <https://www.techno-science.net/glossaire-definition/Python-langage.html>
- [41]: « Pycharm : une définition de Ma petite encyclopédie ». [https://ma-petite-encyclopedia.org/accueil?lex\\_id=1840](https://ma-petite-encyclopedia.org/accueil?lex_id=1840).
- [42]: A. Nandy et M. Biswas, « Reinforcement Learning Basics », in Reinforcement Learning : With Open AI, TensorFlow and Keras Using Python, A. Nandy et M. Biswas, Éd., Berkeley, CA: Apress, 2018, p. 1-18.
- [43]: « Python Data Analytics: Data Analysis and Science using pandas, matplotlib and the Python Programming Language 1, Nelli, Fabio, eBook - Amazon.com ». <https://www.amazon.com/Python-Data-Analytics-matplotlib-Programming-ebook/dp/B01HXXH9ZSK>
- [44] <https://www.techopedia.com/definition/33860/scikit-learn>.
- [45] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.
- [46] YOLO (Partie 3) Non Maxima Suppression (NMS) - datacorner par Benoit Cayla
- [47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

- [48] <https://machinelearningknowledge.ai/a-brief-history-of-yolo-object-detection-models/>
- [49] Terven, Juan. "A Comprehensive Review of YOLO: From YOLOv1 and Beyond." *ArXiv*, (2023).
- [50]:<https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
- [51] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>
- [52] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html>
- [53] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>
- [54] <https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991>
- [55] <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
- [56] <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html>
- [57] <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html>
- [58] <https://datascientest.com/dataset-definition>
- [59] José Camilo Eraso Guerrero, Elena Muñoz España, Mariela Muñoz Añasco, Jesús Emilio Pinto Lopera, Dataset for human fall recognition in an uncontrolled environment, Data in Brief, Volume 45, 2022, 108610, ISSN 2352-3409.
- [60] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014.

[61] M. Zeiler et R. Fergus, "Visualizing and Understanding Convolutional Networks," in European Conference on Computer Vision (ECCV), 2014.

[62] C. Ferri, J. Hernández-Orallo, et R. Modroiu, « An experimental comparison of performance measures for classification », Pattern Recognit. Lett., vol. 30, no 1, p. 27-38, janv. 2009.