الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد – تلمسان –
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Génie biomédical

Spécialité : Informatique Biomédicale et Hospitalière

Par :

- Korchi fatima

- Ouadi imane

Sujet :

Système de surveillance pour malades semi-alités

Soutenu publiquement, le 27/06/2026, devant le jury composé de :

Mr Soulimane Sofiane	Université de Tlemcen	Président
Mme Rerbal Souhila	Université de Tlemcen	Examinatrice
Mr Bechar Hassane	Université de Tlemcen	Encadrareur
Mme Baba Hamed née Bensemain Amel	Ecole Nationale Polytechnique d'Oran ENP	Co-Encadrareur

Année universitaire :2025 /2026

بسم الله الرحمان الرحيم

قال الله تعالى:

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ﴾

(سورة المجادلة، الآية 11)

Remerciements

Avant tout, nous tenons à remercier Allah le Tout-Puissant de nous avoir accordé la force, la patience et la persévérance nécessaires pour mener à bien ce travail.

Nous adressons nos plus sincères remerciements à notre encadreur **M. Bechar Hassane** pour son accompagnement précieux, sa disponibilité constante et ses conseils avisés tout au long de la réalisation de ce projet. Son encadrement rigoureux et son soutien ont été déterminants dans l'aboutissement de ce travail.

Nous exprimons également notre profonde gratitude à notre Co-encadreur **Mme Baba-Hamed Amel**, pour son suivi attentif, ses orientations pertinentes et son aide précieuse, qui ont largement contribué à l'amélioration de la qualité de ce mémoire.

Nous remercions vivement les membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail, ainsi que pour leurs remarques et suggestions constructives.

Nos remerciements s'adressent aussi à l'ensemble des enseignants du département de génie biomédical pour les connaissances et la formation qu'ils nous ont apportées durant notre parcours universitaire.

Nous tenons également à remercier toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce projet.

Enfin, nous exprimons notre profonde reconnaissance à nos familles pour leur soutien indéfectible, leurs encouragements et leurs sacrifices, qui ont été une source de motivation inestimable.

Dédicace

« La reconnaissance est la mémoire du cœur. »

Au terme de ce travail, je dédie le fruit de ces années d'efforts...

À la lumière de ma vie, à celle dont les prières m'ont accompagné à chaque pas,

À ma très chère mère.

Tu as été ma première école, mon refuge dans les moments de doute et la source intarissable de ma force. Aucun mot ne saura exprimer ce que je te dois. Que Dieu te garde et te prête longue vie.

À mon cher père,

Mon modèle et mon pilier, Toi qui t'es sacrifié en silence pour que rien ne me manque, et dont les conseils ont éclairé mon chemin.

À la mémoire de ma très chère grand-mère, que Dieu lui accorde Sa miséricorde, Et à mon très cher grand-père, Pour votre tendresse infinie et vos bénédictions qui m'ont toujours accompagné.

À ma sœur bien-aimée Hayam,

Ma confidente et mon soutien,

Et à mon petit frère Ali,

Ma joie de vivre et ma plus belle fierté.

À ma chère tante et à ses enfants,

Meryem, Younes et Amine, Pour leur présence chaleureuse et leur affection sincère.

À la mémoire de mon cher oncle Mohamed Meliani,

Que Dieu l'accueille en Sa vaste miséricorde — ton souvenir reste gravé dans mon cœur, et j'aurais tant aimé partager cette joie avec toi. À sa famille, à qui je pense avec toute mon affection.

À toute ma famille, proche et lointaine, Pour leurs encouragements et leur amour inconditionnel.

À ma chère grande mère paternelle et mon cher oncle paternel (Kheiredin) et à toute la famille de mon père, pour leurs encouragements.

À toutes mes amies (Fatima, Racha, Aicha, Romaissa et Aya) Avec qui j'ai partagé les rires, les peines et les plus beaux souvenirs de ce parcours.

À l'ingénieur Yassine Touati, Pour son aide précieuse et ses encouragements constants.

À mon binôme Fatima, compagne de ce travail et de cette belle aventure, Nous l'avons commencée ensemble, et c'est ensemble que nous la couronnons aujourd'hui.

À mon encadrant M. Hassane Bechar et à ma co-encadrante Mme Amel Baba Hamed, Pour leur encadrement bienveillant, leur patience et leur générosité.

À toutes celles et ceux qui m'ont soutenu, de près ou de loin, je dédie ce modeste travail

Ouadi Imane

Dédicace

Je dédie ce modeste travail :

À mon cher papa, Pour ton soutien, tes conseils et ta confiance

À ma chère maman, Symbole de tendresse, de patience et de sacrifice, Merci pour ton amour infini, ton soutien et tes encouragements.

À mes chers frères Hmed et Abdelrahman, Pour leur présence et leur soutien

À mon neveu Youcef, fils de mon frère Hmed, Pour la joie et le bonheur qu'il apporte à notre famille

À ma chère sœur Karima et sa fille Yasmine, Pour leur amour et les moments précieux partagés

À ma chère Imane, Pour sa gentillesse et ses encouragements.

À ses filles Assia et Bouchra, et son fils Adam, Que je leur souhaite un avenir plein de bonheur et de réussite

À toute ma famille, Pour leur amour et leurs encouragements

À mes chères amies, Pour leur soutien et leur amitié sincère.

À ma merveilleuse binôme Imane, Avec qui j'ai partagé les efforts et les défis, merci pour ta patience, ton sérieux et ton esprit d'équipe

À mon encadrant M. Hassane Bechar et à ma co-encadrante Mme Amel Baba Hamed Je leur exprime ma sincère gratitude pour leur accompagnement, leurs précieux conseils et leur disponibilité tout au long de ce travail.

À tous ceux qui me sont chers.

Korchi Fatima

Résumé

Ce travail s'inscrit dans le domaine du génie biomédical et porte sur le développement d'un système intelligent de surveillance des patients semi-alités, basé sur la vision par ordinateur et l'intelligence artificielle. L'objectif principal est d'assurer une surveillance automatique, continue et non intrusive, tout en détectant rapidement les situations à risque telles que les chutes ou les sorties de lit.

Le système proposé repose sur une approche hybride combinant plusieurs techniques. La détection de la présence du patient est réalisée à l'aide du modèle YOLOv8, tandis que l'extraction des points clés du corps humain est assurée via MediaPipe. Les caractéristiques extraites sont ensuite traitées par un modèle d'apprentissage automatique basé sur l'algorithme Random Forest pour classifier les différentes postures.

Une base de données personnalisée a été construite afin de pallier l'absence de datasets adaptés à notre problématique, nécessitant un processus de collecte, d'annotation et de structuration des données. Le système est intégré dans un pipeline temps réel avec une interface de monitoring et un système d'alertes.

Les résultats obtenus montrent une performance satisfaisante, malgré certains défis tels que le problème d'occlusion lié aux couvertures, les contraintes des systèmes embarqués et l'optimisation de la base de données. Ce travail ouvre des perspectives intéressantes pour le développement de systèmes de surveillance médicaux intelligents.

Mots-clés : Surveillance intelligente des patients ; Vision par ordinateur ; Intelligence artificielle ; Détection des chutes ; Estimation de pose ; Yolov8 ;MediaPipe ;Random Forest ; Firebase.

ملخص

يندرج هذا العمل ضمن مجال الهندسة الطبية الحيوية، ويهدف إلى تطوير نظام ذكي لمراقبة المرضى شبه الملازمين للفراش، بالاعتماد على تقنيات الرؤية الحاسوبية والذكاء الاصطناعي. ويتمثل الهدف الرئيسي في توفير مراقبة آلية، مستمرة، وغير تدخّلية، مع الكشف السريع عن الحالات الخطرة، مثل السقوط أو مغادرة السرير.

يعتمد النظام المقترح على مقاربة هجينة تجمع بين عدة تقنيات متكاملة. إذ يُستخدم نموذج

YOLOv8 للكشف عن وجود المريض

بينما تُستخرج النقاط المرجعية لجسم الإنسان باستخدام

MediaPipe

بعد ذلك، تُعالَج الخصائص المستخرجة بواسطة نموذج تعلّم آلي يعتمد على خوارزمية

Random Forest لتصنيف مختلف وضعيات المريض.

ونظرًا لعدم توفر قواعد بيانات مناسبة للإشكالية التي يعالجها هذا العمل، تم إنشاء قاعدة بيانات خاصة بالمشروع، تطلبت مراحل جمع البيانات

وتنظيمها(Annotation).

كما دُمج النظام ضمن سلسلة معالجة تعمل في الزمن الحقيقي، مع واجهة لمراقبة حالة المرضى

ونظام تنبيه تلقائي.

أظهرت النتائج المتحصل عليها أداءً مُرضيًا للنظام، رغم وجود بعض التحديات، مثل مشكلة حجب أجزاء من جسم المريض بسبب الأغطية، والقيود المرتبطة بالأنظمة المدمجة، إضافة إلى الحاجة إلى تحسين قاعدة البيانات. ويفتح هذا العمل آفاقًا واعدة لتطوير أنظمة ذكية وأكثر كفاءة لمراقبة المرضى في المجال الطبي .

Abstract

This work falls within the field of biomedical engineering and focuses on the development of an intelligent monitoring system for semi-bedridden patients using computer vision and artificial intelligence techniques. The main objective is to provide continuous, automatic, and non-intrusive monitoring while detecting critical situations such as falls or bed-exit events.

The proposed system relies on a hybrid approach combining multiple technologies. Patient detection is performed using the YOLOv8 model, while human body keypoints are extracted using MediaPipe. The extracted features are then processed using a machine learning model based on the Random Forest algorithm to classify patient postures.

Due to the lack of suitable datasets, a custom database was created, involving data collection, annotation, and structuring. The system operates in real time and includes a monitoring interface and alert mechanism.

The obtained results demonstrate satisfactory performance despite several challenges, including occlusion caused by blankets, limitations of embedded systems, and database optimization. This work opens promising perspectives for the development of intelligent medical monitoring systems.

Table des matières :

Introduction générale 15

Chapitre I : Revue de la littérature 18

Introduction à l'état de l'art : 19

1.1. Grandes familles de solutions de surveillance : 19

1.1.1. Solutions de contact (capteurs portés ou capteurs de lit) : 19

1.1.2. Solutions sans contact (caméra et vision par ordinateur) : 19

1.2. Tâches vision principales pour patients alités : 20

1.2.1. Reconnaissance de posture au lit (in-bed pose / posture classification) : 20

Une approche représentative est la classification des postures (ex. sapine/prône/latérale) à partir d'une caméra type "*home security camera*" [NIH] ; par exemple, un système multitâche a montré la détection et la classification simultanées de la tête et du haut du corps avec des performances élevées sur des données difficiles [NIH]. 20

1.2.2. Détection de sortie de lit (bed-exit detection) : 20

1.2.3. Détection de chute / risque de chute près du lit : 20

1.2.4. Analyse du visage : expression, douleur, détresse (optionnel) : 20

1.3. Méthodes et modèles utilisés en vision embarquée : 21

1.3.1. Du "vision classique" au deep learning : 21

1.3.2. Architectures typiques (exemples concrets) : 21

- **Faster R-CNN / multi-tâches : utilisé pour détecter tête/haut du corps puis classifier la pose, avec une approche "tout-en-un" pour réduire latence et simplifier le pipeline [NIH].** 21

1.4. Jeux de données (datasets) : 21

1.4.1. Détection de chute (vidéo) : 21

1.4.2. Expression faciale 21

1.4.3. Postures au lit / pose estimation (panorama) : 22

1.5. Déploiement sur Raspberry Pi : edge-AI et contraintes : 22

1.6. Enjeux de recherche et limites actuelles : 22

Chapitre II : Concepts de base et fondements théoriques 23

Introduction 24

2.1. Les systèmes embarqués : 24

2.1.1 Les caractéristiques principales d'un système embarqué : 25

2.1.2 Importance des systèmes embarqués dans le domaine médical : 26

2.1.3 les systèmes embarqués en temps réel : 27
2.2 Le traitement d'images : 28
2.2.1 Image numérique : 29
2.2.2 Types d'images numériques : 29
2.2.3 Numérisation d'une image : 30
2.2.4 Étapes du traitement d'images : 31
2.3. La Vision par Ordinateur et l'IA appliquées à l'image : 33
2.3.1 Définition de la vision par ordinateur : 33
2.3.2 Intelligence artificielle appliquée à l'image : 33
2.3.3 Principales tâches de la vision par ordinateur : 33
2.3.4 Apport de l'IA et de la vision par ordinateur en milieu médical : 35
Conclusion : 36
Chapitre III : Conception et réalisation 37
Introduction : 38
3.1 Patients semi-alités et méthodes de surveillance : 38
3.2. Architecture Globale du Système : 39
3.3 Création de la base de données : 40
3.3.1. Acquisition des données vidéo et protocole expérimental : 40
3.3.2. Extraction des caractéristiques squelettiques avec MediaPipe : 41
3.3.3. Structuration et purification des données (Format CSV) : 42
3.4. Modélisation et Apprentissage Automatique (*Machine Learning*) : 44
3.4.1. Justification du choix de l'algorithme (*Random Forest*) : 44
3.4.2. Phase d'entraînement et séparation des données (*Train/Test*) : 44
3.5. Pipeline du Système de Surveillance en Temps Réel : 45
3.5.1. Détection de la présence du patient avec YOLOv8 (Optimisation) : 45
3.5.2. Classification continue de la posture : 46
3.5.3. Mécanisme de journalisation et d'archivage local (Redondance) : 46
3.6. Architecture IoT et Interface de Monitoring : 46
3.6.1. Synchronisation Cloud via *Firebase Realtime Database:* 47
3.6.2. Développement du tableau de bord multi-chambres : 47
3.6.3. Système de gestion des alertes et notifications sonores : 47
3.7. Environnement de Développement et Outils Utilisés: 48
Conclusion 49
Chapitre IV : Résultats et discussion 50

Introduction : 51
4.1. Évaluation du Modèle d'Intelligence Artificielle : 51
4.1.1. Les métriques d'évaluation : 51
4.1.2. Précision Globale du modèle (*Accuracy globale*) : 52
4.1.3. Analyse de la Matrice de Confusion 53
..... 53
4.1.4 Rapport de classification détaillé par posture 53
Tableau 4.1: Résultats de l'évaluation du classifieur par classe. 54
4.2. Interprétation et Discussion des Résultats 54
4.2.1. Les points forts et performances optimales : 54
4.2.2. Les limites initiales du modèle et résolution du phénomène d'occlusion : 54
4.3. Présentation et Validation de l'Application Mobile (*Dashboard NurseView*) : 59
4.4. Évaluation globale du Système de Bout en Bout (*End-to-End*) : 62
4.4.1. Journalisation et Traçabilité des Événements (Le Journal de Surveillance) : 62
4.4.2. Étude de l'Intégration du Système Embarqué (Raspberry Pi) : 63
Architecture et Contraintes : 63
4.4.3. Perspectives d'Évolution et Améliorations Futures : 64
4.5. Travaux connexes : 65
Conclusion : 67
Conclusion générale et perspectives : 68

Liste des figures

Chapitre II :

Figure 2.1: Applications des systèmes embarqués 25
Figure 2.2:le cycle de fonctionnement d'un système temps réel 28
Figure 2.3 :Les différentes étapes de traitement d'image 31

Chapitre III :

Figure 3.1 : Architecture Globale du Système.....40
Figure 3.2 : Échantillons des séquences vidéo acquises illustrant la diversité des acteurs. 41
Figure 3.3 : Extraction des 33 points clés (Landmarks) sur un sujet de notre base de données. 42
Figure 3.4: Extrait du fichier CSV contenant les coordonnées articulaires et les labels 42

Chapitre IV :

Figure 4.1 : Matrice de confusion du modèle Random Forest. 53
Figure 4.2 : Extraction du squelette et classification en temps réel de la posture "Assis". 56
Figure 4.3 : Capture du flux vidéo montrant la détection correcte de la posture "Dos". 57
Figure 4.4 : Performance du système en situation d'occlusion partielle par la couverture. 58
Figure 4.5 : Détection instantanée d'un événement critique de type "Sortie de lit". 59
Figure 4.6 : Interface principale du Dashboard NurseView en temps réel. 60
Figure 4.7 : Écran d'historique et de traçabilité des postures. 61
Figure 4.8 : interface quand il y'a une alerte de chute 61
Figure 4.9 : Capture d'écran du fichier de journalisation locale (journal_surveillance.txt). 63

Liste des abréviations

- IA: Intelligence Artificielle
- RGB: Red Green Blue
- IR: Infrarouge
- LWIR: Long-Wave Infrared
- BED Net: Bed Exit Detection Network
- YOLO: You Only Look Once
- FER: Facial Expression Recognition
- CNN: Convolutional Neural Network
- ADL: Activities of Daily Living
- FPS: Frames Per Second
- GPU: Graphics Processing Unit
- CPU: Central Processing Unit
- AI : Artificial Intelligence
- IMX500 : Intelligent Vision Sensor IMX500
- HSV : Hue, Saturation, Value
- YCbCr : Luminance, Chrominance Bleue, Chrominance Rouge
- OCR : Optical Character Recognition
- R-CNN: Region-Based Convolutional Neural Network
- GNN : Graph Neural Network
- VLM : Vision Language Model
- GAN : Generative Adversarial Network
- VAE : Variational Autoencoder
- OpenCV : Open Source Computer Vision
- MediaPipe : Media Processing Pipeline
- SSD : Single Shot Detector
- SVM : Support Vector Machine
- RNN : Recurrent Neural Network
- LSTM : Long Short-Term Memory
- MP4 : Format vidéo numérique
- CSV : Comma-Separated Values
- RF : Random Forest
- JSON : JavaScript Object Notation
- UI : User Interface
- UX : User Experience
- HTML : HyperText Markup Language
- CSS : Cascading Style Sheets
- IDE : Integrated Development Environnements
- VS Code : Visual Studio Code
- IoT : Internet of Things

Introduction générale

Contexte :

Dans le domaine du génie biomédical, l'intégration des technologies numériques et des systèmes intelligents a profondément transformé les méthodes de surveillance et de prise en charge des patients. Parmi les problématiques majeures rencontrées en milieu médical, la surveillance des patients à mobilité réduite, notamment les patients **semi-alités**, constitue un enjeu important en termes de sécurité, de confort et de qualité des soins.

Les patients semi-alités sont des personnes dont la mobilité est partiellement limitée, les obligeant à passer une grande partie de leur temps au lit, tout en conservant une capacité réduite à changer de position ou à s'asseoir avec assistance. Cette situation les expose à plusieurs risques, en particulier les chutes, les escarres ainsi que les troubles circulatoires. Les méthodes de surveillance traditionnelles, basées principalement sur l'observation humaine, présentent des limites importantes, telles que la discontinuité du suivi, la charge de travail élevée et le risque d'erreurs.

Avec l'évolution des technologies, les systèmes embarqués associés à la vision par ordinateur et à l'intelligence artificielle offrent de nouvelles solutions pour améliorer la surveillance des patients. Ces approches permettent d'analyser automatiquement des flux vidéo afin de détecter la présence du patient, d'estimer sa posture et d'identifier rapidement les situations à risque. L'intégration d'un système d'alertes intelligentes permet également d'améliorer la réactivité du personnel médical en cas d'événement critique.

Problématique, motivation et positionnement :

Dans ce contexte, la problématique de ce travail peut être formulée comme suit :

Comment concevoir un système intelligent basé sur l'analyse vidéo permettant d'assurer une surveillance automatique, continue et fiable des patients semi-alités, tout en détectant rapidement les situations à risque et en réduisant l'intervention humaine ?

Afin de répondre à cette problématique, les hypothèses suivantes sont posées :

- L'utilisation de la vision par ordinateur permet de détecter efficacement la présence du patient.
- L'extraction des points clés du corps humain permet de représenter correctement les postures.
- Les algorithmes d'apprentissage automatique permettent de classifier les postures avec précision.
- L'intégration d'un système d'alertes permet d'améliorer la prise en charge des situations critiques.
- Le déploiement sur une plateforme embarquée (Raspberry Pi) permet un traitement local rapide et autonome.

L'objectif principal de ce projet est de concevoir et de développer un **système intelligent de surveillance des patients semi-alités**, basé sur **l'analyse vidéo**, permettant de détecter automatiquement les postures du patient et de signaler les situations à risque en **temps réel**.

Pour atteindre cet objectif, les actions suivantes ont été réalisées :

- Assurer l'acquisition des données vidéo du patient.
- Détecter la présence du patient à l'aide de YOLO.
- Extraire les points clés du corps via MediaPipe.
- Classifier les postures en utilisant un modèle d'apprentissage automatique.
- Enregistrer les données dans un journal (CSV).
- Développer une application mobile pour la visualisation et les alertes.
- Prévoir le déploiement du système sur une plateforme Raspberry Pi.

Le choix de ce projet est motivé par l'intérêt pour l'application des technologies de l'intelligence artificielle dans le domaine médical. La combinaison de la vision par ordinateur et des systèmes embarqués permet de proposer des solutions innovantes répondant à des besoins réels, notamment en matière de surveillance des patients semi-alités. Ce projet constitue également une opportunité de développer des compétences techniques dans plusieurs domaines complémentaires.

Organisation du mémoire :

Le présent mémoire est structuré comme suit :

- **Chapitre 1 :** Revue de la littérature

Présentation des travaux existants dans le domaine de la surveillance des patients.

- **Chapitre 2 :** Concepts de base et fondements théoriques

Introduction aux systèmes embarqués, au traitement d'images et à la vision par ordinateur.

- **Chapitre 3 :** Conception et réalisation

Description de l'architecture du système et des différentes étapes de son implémentation.

- **Chapitre 4 :** Résultats et discussions

Analyse des résultats obtenus et discussion des performances du système.

En conclusion, ce travail a permis de concevoir un système intelligent de surveillance des patients semi-alités basé sur la vision par ordinateur et l'intelligence artificielle, contribuant ainsi à améliorer la sécurité et la qualité des soins.

Ce projet ouvre également des perspectives intéressantes, notamment l'amélioration des performances du système, son déploiement sur des plateformes embarquées comme le Raspberry Pi, ainsi que son adaptation à des environnements médicaux réels.

Chapitre I : Revue de la littérature

Introduction à l'état de l'art :

Ces dernières années, le secteur de la santé a connu une évolution remarquable grâce à l'intégration des technologies intelligentes, notamment les systèmes embarqués, le traitement d'images et l'intelligence artificielle [Topol, (2019), Aceto Aceto et al. (2020)]. Ces technologies ont contribué à l'amélioration de la qualité des soins, à la réduction de la charge de travail du personnel médical et à l'augmentation de la sécurité des patients. [Haque et al. (2020)]

Ce chapitre présente une revue de la littérature portant sur les systèmes intelligents utilisés en milieu médical, avec un accent particulier sur les systèmes de surveillance des patients semi-alités, basés sur la vision par ordinateur et les plateformes embarquées, les principales approches existantes, leurs avantages ainsi que leurs limites, ainsi qu'un état de l'art des systèmes de surveillance existants pour les malades alités.

1.1. Grandes familles de solutions de surveillance :

1.1.1. Solutions de contact (capteurs portés ou capteurs de lit) :

Les approches non *vision-based* incluent des tapis de pression, capteurs de charge, ou capteurs portés (accéléromètres), capables de détecter des changements de poids ou des mouvements. Cependant, les tapis de pression peuvent exiger une maintenance (positionnement, nettoyage, désinfection) et générer des fausses alarmes, ce qui réduit l'acceptabilité au quotidien.

Les capteurs portés posent des contraintes d'usage (oubli de port, charge batterie, inconfort), ce qui limite parfois leur déploiement en contexte réel.

1.1.2. Solutions sans contact (caméra et vision par ordinateur) :

Les systèmes basés sur caméra sont non intrusifs et peuvent fournir un contexte riche : posture, mouvements, trajectoires et franchissement du bord du lit [Haque et al., (2020)]. Les approches de vision peuvent être basées sur des caméras RGB, infrarouges (IR/LWIR), des capteurs de profondeur ou des systèmes multimodaux [arXiv]. Leur objectif est généralement de résister aux occultations causées par les couvertures ou les draps, ainsi qu'aux variations d'éclairage.

Un enjeu majeur reste la confidentialité. Pour limiter l'exposition des données personnelles, certains travaux recommandent l'utilisation de modalités alternatives telles que l'infrarouge, la profondeur ou les capteurs de pression, ainsi que le traitement local des données. [arXiv].

1.2. Tâches vision principales pour patients alités :

1.2.1. Reconnaissance de posture au lit (in-bed pose / posture classification) :

Une approche représentative est la classification des postures (ex. sapine/prône/latérale) à partir d'une caméra type "*home security camera*" [NIH] ; par exemple, un système multitâche a montré la détection et la classification simultanées de la tête et du haut du corps avec des performances élevées sur des données difficiles [NIH].

Les revues récentes sur "*in-bed pose estimation*" distinguent des méthodes unies modales (souvent RGB) et multimodales (RGB + LWIR + profondeur + pression), la multi modalité aidant à gérer l'occlusion par couverture [ArXiv]. La littérature insiste aussi sur la rareté des datasets annotés en contexte réel (problèmes de coût et de vie privée), ce qui freine l'évaluation comparative et la généralisation des modèles **[**arXiv**].**

1.2.2. Détection de sortie de lit (bed-exit detection) :

Les méthodes modernes exploitent souvent l'estimation de pose (squelette) : l'idée est de détecter l'action "*exit*" à partir d'une séquence de poses et de la géométrie du lit. Un exemple académique propose un réseau *BED Net* composé de deux sous-réseaux : un module de détection de posture (*keypoints*) et un module de reconnaissance d'action sur séquence pour décider "*stay*" vs "*exit*" **[**Imaging**].**

Côté implémentation pratique, des projets utilisent YOLOv8-Pose (modèle d'IA rapide pour la détection humaine et l'estimation de posture en temps réel) et déclenchent une alerte dès que les jambes franchissent une ligne représentant le bord du lit, ce qui illustre une logique simple et déplorable **[**GitHub**].**

1.2.3. Détection de chute / risque de chute près du lit :

Des approches webcam et IA évaluent le risque en vérifiant la présence/absence de keypoints et en déclenchant une alarme si des zones clés (tête/épaules/hanches) sortent de la zone "lit" **[**ACM**]**. Sur edge-AI, des démonstrations sur Raspberry Pi utilisent la détection de *keypoint*s (ex. HigherHRNet) et des critères géométriques (angles tête-genou-pied) pour classifier une posture de chute en temps réel **[**Hackster**].**

Enfin, des plateformes industrielles/partenariales proposent des pipelines vidéo "*continuous monitoring*" axés sur la sécurité (présence, franchissement de limites, patient seul), ce qui montre l'intérêt croissant de l'analyse vidéo en environnement clinique **[**arXiv**].**

1.2.4. Analyse du visage : expression, douleur, détresse (optionnel) :

Pour des scénarios où la caméra voit le visage (patient semi-assis, tête visible), la reconnaissance d'expression peut servir d'indice de détresse ou d'inconfort, mais elle reste indirecte et dépend fortement des conditions (angle, occlusion, lumière). Le dataset FER-2013 est une base classique pour l'apprentissage

d'expressions faciales (images 48×48 en niveaux de gris, 7 classes émotionnelles, découpages train/test) et peut être une brique de départ pour prototyper **[**Dataset FER-2013**].**

1.3. Méthodes et modèles utilisés en vision embarquée :

1.3.1. Du "vision classique" au deep learning :

Les systèmes historiques reposaient sur des heuristiques (détection de mouvement, segmentation simple), mais les approches récentes privilégient des modèles profonds pour la robustesse aux changements d'éclairage et de posture. L'estimation de pose (*skeleton keypoints*) est devenue centrale car elle compacte l'information (positions articulaires) tout en réduisant le besoin de détails visuels fins, utile pour *le bed-exit/fall* detection.

1.3.2. Architectures typiques (exemples concrets) :

- **Faster R-CNN / multi-tâches : utilisé pour détecter tête/haut du corps puis classifier la pose, avec une approche "tout-en-un" pour réduire latence et simplifier le pipeline [NIH].**
- **Keypoints + Action recognition** : BED Net détecte les keypoints (via un modèle pré-entraîné) puis reconnaît l'action sur une séquence temporelle via un réseau 3D/ResNet-like [Imaging].
- **YOLOv8-Pose + règle de franchissement** : pipeline léger où la décision est prise via géométrie (ligne "*bed edge*"), pratique pour une implémentation rapide [GitHub].
- **Keypoints + géométrie** : chute détectée par critères d'angles/positions à partir des points de squelette, ce qui est adapté à un système embarqué [Hackster].

1.4. Jeux de données (datasets) :

1.4.1. Détection de chute (vidéo) :

Le dataset Zenodo GMDCSA-24 contient des clips vidéo d'activités de la vie quotidienne (ADL) et de chutes, réalisés par plusieurs acteurs et environnements domestiques, ce qui peut servir à l'entraînement/évaluation en "*fall detection*" **[**Dataset GMDCSA-24**].**

1.4.2. Expression faciale

FER-2013 fournit des visages centrés en 48×48 avec étiquettes émotionnelles (7 classes), très utilisé pour des CNN compacts et des essais rapides de classification **[**Dataset FER-2013**].**

1.4.3. Postures au lit / pose estimation (panorama) :

Une revue récente synthétise plusieurs datasets "*in-bed*" avec diverses modalités (RGB, LWIR, profondeur, pression) et discute leurs tailles, labels et limites, ce qui est utile pour justifier les choix de données et les métriques **[**arXiv**]**.

1.5. Déploiement sur Raspberry Pi : edge-AI et contraintes :

Le Raspberry Pi est une plateforme fréquente pour prototyper des solutions de santé à coût réduit, mais la contrainte principale reste dans le calcul temps réel (FPS) et la latence sans GPU dédié. L'arrivée de caméras "AI" (ex. IMX500) permet de déporter l'inférence sur le capteur, libérant le CPU du Raspberry Pi pour la logique applicative (alertes, interface, réseau) **[**Raspberry Pi.**]**.

Les modèles disponibles dans des "*model zoos*" (classification, détection d'objets, segmentation, pose estimation) facilitent le prototypage rapide d'applications de surveillance sur l'edge **[**Raspberry Pi.**]**. Des démonstrations orientées "*fall detection*" sur Raspberry Pi montrent aussi l'intérêt d'un traitement local (*edge computing*) pour une solution "*privacy-conscious*", car l'analyse peut se faire sans envoyer la vidéo brute au cloud **[**Hackster**]**.

1.6. Enjeux de recherche et limites actuelles :

- **Occlusion et couverture** : les draps/couvertures masquent les membres, rendant l'estimation de pose difficile ; la multi modalité (LWIR, pression, profondeur) est souvent proposée pour compenser [arXiv].
- **Variabilité d'éclairage et angles caméra** : les performances chutent si la distribution (hôpital vs domicile, hauteur de caméra) diffère de l'entraînement.
- **Données annotées limitées** : les datasets réels sont rares pour des raisons de confidentialité et de coût, ce qui complique la comparaison et la généralisation [arXiv].
- **Fausses alarmes** : un défi majeur des alarmes "*bed-exit*" est la réduction des faux positifs, car ils fatiguent le personnel et diminuent la confiance dans le système [ACM].
- **Confidentialité** : la vidéo de chambre est sensible ; des stratégies incluent floutage, *pose-only* (squelettes), modalités non RGB, et traitement local [ACM].

Chapitre II : Concepts de base et fondements théoriques

Introduction

Un patient semi-alité est une personne dont l'état de santé limite partiellement sa mobilité, l'obligeant à passer une grande partie du temps au lit tout en conservant une capacité réduite à s'asseoir ou à effectuer certains mouvements avec assistance. Cette situation nécessite une surveillance régulière afin d'assurer son confort et de prévenir les complications liées à la position prolongée.

Dans ce cadre, le chapitre présente les notions générales indispensables à la compréhension du projet. Il introduit les concepts fondamentaux des systèmes embarqués et du traitement d'images, ainsi que leur rôle dans le développement de solutions intelligentes dédiées à la surveillance médicale. Ce chapitre aborde également la notion de patient semi-alité et met en évidence l'importance d'un suivi postural continu.

2.1. Les systèmes embarqués :

Un système embarqué est un système informatique dédié, intégré au sein d'un dispositif matériel, conçu pour exécuter une ou plusieurs fonctions spécifiques. Contrairement aux ordinateurs classiques, il est généralement destiné à une application précise et fonctionne souvent en temps réel avec des ressources matérielles limitées, telles que le processeur, la mémoire et l'énergie [Tammy Noergaard, 2013].

Les systèmes embarqués se caractérisent par leur autonomie, leur fiabilité, leur faible consommation énergétique et leur capacité à fonctionner de manière continue sans intervention humaine. Ils sont largement utilisés dans plusieurs domaines tels que l'automobile, l'industrie et les télécommunications, et particulièrement dans le domaine médical, où ils permettent le développement de solutions de surveillance et d'assistance intelligentes [Lee & Seshia, 2017].

Dans de nombreux domaines, les systèmes embarqués sont largement utilisés, notamment dans le transport (aéronautique, automobile, ferroviaire), les appareils électroniques grand public (appareils photo, téléviseurs, électroménager, smartphones), ainsi que dans les secteurs de l'énergie et de l'automatisation. La Figure 2.1 illustre quelques exemples d'applications des systèmes embarqués.

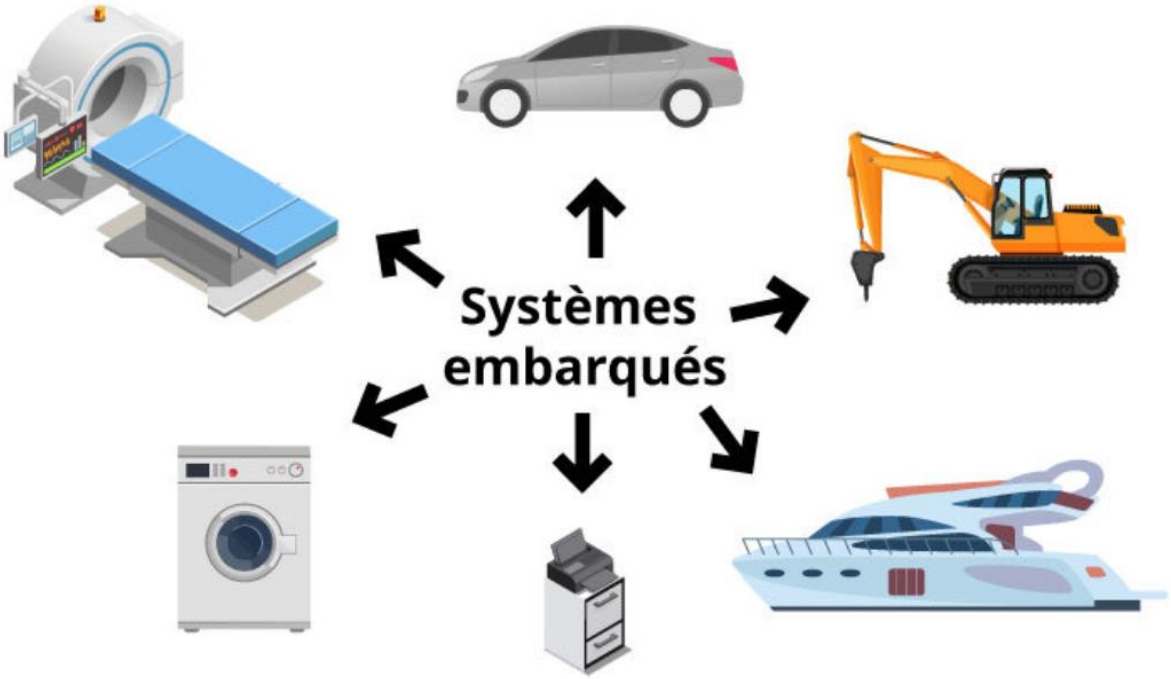


Figure 2.1 : Applications des systèmes embarqués

Dans le domaine médical, ces systèmes jouent un rôle particulièrement important. Ils permettent la surveillance continue des patients, la collecte de données physiologiques et l'analyse en temps réel des informations. Grâce à leur intégration avec des capteurs et des caméras, ils contribuent au développement de dispositifs médicaux intelligents visant à améliorer la qualité des soins et la prise en charge des patients [Biomedical Engineering – Medical Embedded Systems, 2020].

Dans le cadre de ce projet, le Raspberry Pi a été choisi comme plateforme embarquée. Ce choix s'explique par sa compacité, son faible coût et sa compatibilité avec le langage Python ainsi que les bibliothèques de traitement d'images. Il constitue ainsi une solution adaptée pour le traitement des données visuelles et la surveillance en temps réel des patients semi-alités [Raspberry Pi Foundation, 2024].

Un système embarqué peut être décomposé en deux entités principales :

Le système contrôlé (environnement) : il représente le procédé équipé de capteurs et d'actionneurs, interagissant avec le monde extérieur.

Le système de contrôle : il regroupe les éléments matériels (microprocesseurs, microcontrôleurs) et logiciels responsables du traitement des données et de la prise de décision.

2.1.1 Les caractéristiques principales d'un système embarqué :

Les systèmes embarqués sont des systèmes informatiques dédiés, intégrés dans des dispositifs ou des équipements afin d'assurer des fonctions spécifiques. Ils sont conçus pour fonctionner de manière autonome dans des environnements contraints, tout en interagissant étroitement avec des composants matériels et logiciels.

- Autonomie :

Les systèmes embarqués sont capables de fonctionner sans intervention humaine continue. Ils peuvent prendre des décisions automatiquement en s'appuyant sur des règles prédéfinies ou des algorithmes de contrôle, tout en réagissant aux informations provenant de leur environnement.

- Spécialisation :

Un système embarqué est généralement optimisé pour accomplir une tâche bien définie. Contrairement aux systèmes informatiques à usage général, il ne mobilise que les ressources strictement nécessaires à l'exécution de sa fonction, ce qui améliore son efficacité globale [Marwedel, 2018].

- Intégration :

Les systèmes embarqués sont étroitement intégrés aux dispositifs physiques dans lesquels ils sont implantés. Ils fonctionnent en coordination avec d'autres éléments matériels et logiciels, tels que les capteurs, les actionneurs et les modules de communication.

- Fonctionnement dans un environnement contraint :

Ces systèmes sont conçus pour opérer avec des ressources limitées, notamment en termes de mémoire, de puissance de calcul et de consommation énergétique. Cette contrainte impose une optimisation rigoureuse du matériel et des algorithmes utilisés.

- Interactivité :

Les systèmes embarqués peuvent interagir avec leur environnement en collectant des données à partir de capteurs, en les analysant et en déclenchant des actions via des actionneurs ou d'autres dispositifs connectés.

- Fiabilité et robustesse :

Dans des applications critiques, en particulier dans le domaine médical, les systèmes embarqués doivent présenter un haut niveau de fiabilité et de robustesse. Ils sont conçus pour fonctionner de manière continue, détecter les défaillances et garantir la sécurité des utilisateurs [Jane W. S. Liu, 2000].

- Performances adaptées :

Les performances d'un système embarqué sont étroitement liées à son domaine d'application. Il doit exécuter ses tâches de manière rapide et efficace en utilisant des techniques optimisées, tout en respectant les contraintes de ressources imposées

2.1.2 Importance des systèmes embarqués dans le domaine médical :

Les systèmes embarqués jouent un rôle crucial dans le domaine médical, en raison de leur capacité à assurer une surveillance continue et fiable des patients. Ils sont utilisés dans plusieurs applications telles que le suivi des paramètres physiologiques, l'assistance aux patients à mobilité réduite et la détection

d’événements critiques en temps réel. Grâce à leur faible consommation énergétique et à leur intégration avec différents capteurs et dispositifs, les systèmes embarqués contribuent à améliorer la qualité des soins et la prise en charge médicale, tout en réduisant la charge de travail du personnel soignant [Biomedical Engineering – Medical Embedded Systems, 2020].

2.1.3 les systèmes embarqués en temps réel :

Un système temps réel est un système dont le fonctionnement est soumis à des contraintes temporelles strictes. Dans ce type de système, la validité des résultats dépend non seulement de leur exactitude, mais également du moment où ils sont produits.

Ainsi, une réponse correcte mais fournie en retard peut être considérée comme une erreur. Cette caractéristique distingue les systèmes temps réel des systèmes informatiques classiques, qui ne tiennent pas compte du facteur temporel de la même manière.

Lorsqu’il est intégré dans un système embarqué, on parle alors de système embarqué en temps réel, dans lequel les données doivent être acquises, traitées et exploitées dans un délai déterminé, en adéquation avec la dynamique du procédé à contrôler.

Les systèmes embarqués en temps réel constituent une catégorie essentielle des systèmes, caractérisée par la nécessité de respecter des contraintes temporelles strictes. Leur utilisation est particulièrement critique dans les domaines où la sécurité et la réactivité sont primordiales, tels que le domaine médical [Liu, 2000].

2.1.3.1 Types de systèmes temps réel :

On distingue deux catégories majeures selon la sévérité des conséquences en cas de dépassement des délais :

- **Le temps réel dur (*Hard Real-Time*) :** Dans ces systèmes, le respect des échéances est impératif. Un seul dépassement peut entraîner une panne systémique, des pertes économiques massives ou, dans le cadre médical, mettre en danger la vie du patient.
  Exemple : Le contrôleur de freinage d'un véhicule ou un système d'assistance respiratoire.
- **Le temps réel mou (*Soft Real-Time*)** : Dans ce cas, le non-respect des échéances n'entraîne pas de défaillance critique, mais réduit la qualité de service en dégradant les performances.
  Exemple : L'affichage des données sur un écran de contrôle ou l'enregistrement d'un journal d'activité.

2.1.3.2 Fonctionnement :

Le fonctionnement d’un système embarqué en temps réel (figure 2.2) repose généralement sur une boucle de traitement continue :

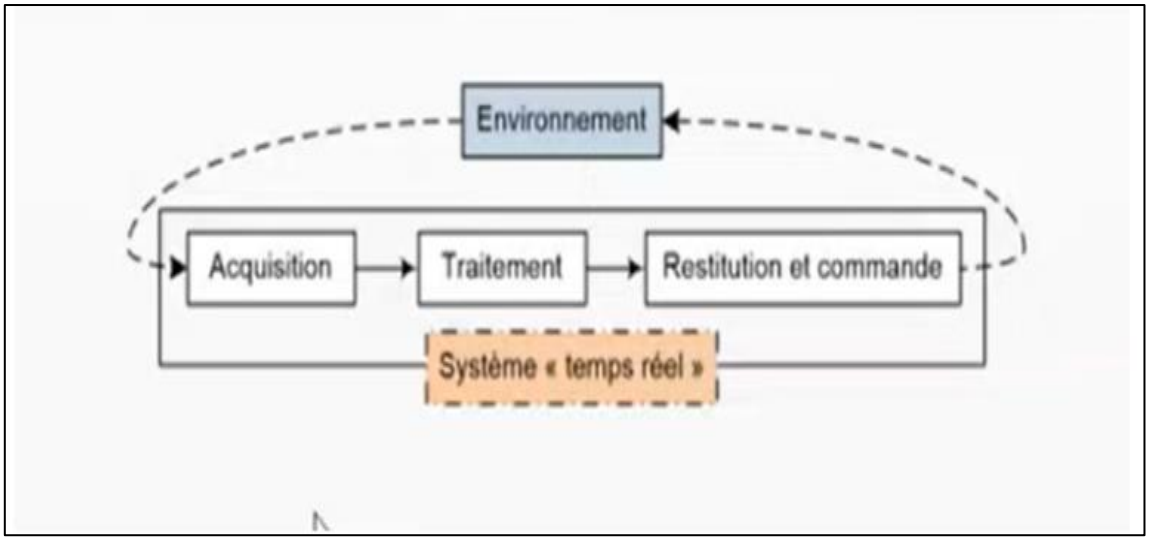


Figure 2.2 le cycle de fonctionnement d'un système temps réel

- Acquisition des données via les capteurs
- Traitement des informations
- Prise de décision
- Envoi des commandes aux actionneurs

Toutes ces étapes doivent être exécutées dans un temps déterminé afin de garantir la validité des résultats. Pour garantir cette réactivité, le système repose sur des modes de fonctionnement spécifiques :

- **Le fonctionnement cyclique (*Time-driven*) :** Le système traite les données à des intervalles réguliers (échantillonnage) basés sur l'horloge du système.
- **Le fonctionnement événementiel (*Event-driven*) :** Le système est activé par des événements extérieurs via des mécanismes d'interruptions matérielles.
- **Le multitâche et l'ordonnancement :** Étant souvent complexe, le système doit gérer plusieurs tâches simultanément (ex : capturer une image, analyser la pose, envoyer une alerte). Cela nécessite un gestionnaire de tâches appelé ordonnanceur, qui définit les priorités et gère la préemption.

Enfin, l'architecture d'un système embarqué repose sur l'interaction entre les capteurs, l'unité de traitement et les actionneurs. Cette organisation permet d'assurer une acquisition, un traitement et une réaction en temps réel aux événements provenant de l'environnement.

2.2 Le traitement d'images :

Le traitement d'images est une discipline de l'informatique et du génie biomédicale, qui consiste à analyser et à transformer des images numériques dans le but d'en améliorer la qualité ou d'en extraire des informations pertinentes. Il joue un rôle essentiel dans de nombreuses applications médicales, notamment l'imagerie médicale, la surveillance des patients et l'analyse automatique de scènes visuelles [Gonzalez & Woods, 2018].

Dans le domaine de la surveillance médicale, le traitement d'images permet d'exploiter des images ou des séquences vidéo acquises à partir des caméras, afin d'observer le comportement, les mouvements et les postures des patients de manière automatique et non intrusive. Il constitue ainsi une base fondamentale pour les systèmes de vision artificielle utilisés dans la surveillance des patients alités ou semi-alités [Richard Szeliski, 2022].

2.2.1 Image numérique :

Les images numériques peuvent être binaires, en niveaux de gris ou en couleurs. Dans le cas des images couleur, chaque pixel est défini par plusieurs composantes chromatiques (RGB). La qualité et la précision de l'image dépendent de la profondeur de quantification utilisée. [Geoff Dougherty, 2009].

Les principales caractéristiques d'une image numérique incluent la résolution spatiale, qui correspond au nombre de pixels, la profondeur de quantification, qui définit le nombre de niveaux d'intensité possibles, ainsi que la résolution temporelle dans le cas des séquences vidéo. Ces paramètres influencent directement la qualité de l'image et la précision de l'analyse réalisée par les algorithmes de traitement d'images [Gonzalez & Woods, 2018].

Dans les applications médicales, la qualité des images numériques est un facteur déterminant pour assurer une détection fiable des structures, des mouvements ou des postures du patient.

2.2.2 Types d'images numériques :

Les images numériques peuvent être classées selon leur mode de représentation et de codage de l'information visuelle. Chaque type d'image numérique possède des caractéristiques spécifiques, qui influencent les techniques de traitement et d'analyse employées en vision artificielle et en imagerie médicale.

- Images binaires :

Les images binaires sont des images numériques composées uniquement de deux niveaux d'intensité, généralement représentés par les valeurs 0 et 1, correspondant respectivement au noir et blanc. Elles sont souvent obtenues après une opération de seuillage et utilisées pour simplifier la représentation des objets ou des régions d'intérêt.

- Images en niveaux de gris :

Les images en niveaux de gris sont des images numériques dans lesquelles chaque pixel est représenté par une seule valeur d'intensité lumineuse. Cette valeur est généralement codée sur 8 bits, offrant 256 niveaux possibles allant du noir au blanc. Ce type d'image est largement utilisé en traitement d'images médicales, car il facilite l'analyse des formes, des contours et des textures.

- Images couleur :

Les images couleur sont des images numériques décrites par plusieurs composantes chromatiques. Le modèle de couleur le plus utilisé est le modèle RGB, dans lequel chaque pixel est représenté par trois composantes : rouge, vert et bleu. D'autres espaces colorimétriques tels que HSV, YCbCr ou Lab sont également employés afin d'améliorer certaines opérations de traitement comme la segmentation ou la reconnaissance d'objets.

- Images multispectrales et hyper-spectrales :

Les images multispectrales sont constituées de plusieurs bandes spectrales acquises à différentes longueurs d'onde, tandis que les images hyper-spectrales contiennent un nombre beaucoup plus élevé de bandes spectrales fines. Bien que principalement utilisées dans la télédétection, ces images peuvent également apparaître dans certaines applications biomédicales avancées.

- Images fixes et images vidéo :

Les images fixes représentent une scène capturée à un instant donné, tandis que les images vidéo correspondent à une succession d'images fixes acquises dans le temps. Les images vidéo sont particulièrement importantes dans les systèmes de surveillance médicale, car elles permettent l'analyse dynamique des mouvements et des postures des patients [Dougherty, 2009].

2.2.3 Numérisation d'une image :

La numérisation d'une image est le processus qui consiste à convertir une image du monde réel (image analogique) en une image numérique, pouvant être traitée par un système informatique. Cette conversion repose principalement sur deux opérations fondamentales : l'échantillonnage spatial et la quantification des niveaux d'intensité [Gonzalez & Woods, 2018].

- Échantillonnage spatial :

L'échantillonnage spatial consiste à discrétiser l'image analogique selon les dimensions spatiales (axes horizontal et vertical). L'image est ainsi découpée en un ensemble fini de points appelés pixels, disposés sous forme d'une matrice. La densité de cet échantillonnage définit la résolution spatiale de l'image : plus le nombre de pixels est élevé, plus les détails de l'image sont fidèlement représentés [Gonzalez & Woods, 2018].

- Quantification :

La quantification correspond à la discrétisation des valeurs d'intensité lumineuse associées à chaque pixel. Chaque valeur continue est approximée par un niveau parmi un nombre fini de niveaux possibles. Ce nombre dépend de la profondeur de quantification (par exemple : 8 bits, soit 256 niveaux de gris). Une quantification plus fine permet de mieux représenter les variations d'intensité, mais augmente la taille des données numériques [Dougherty, 2009].

- Codage numérique :

Après l'échantillonnage et la quantification, les valeurs obtenues sont codées sous forme binaire afin d'être stockées, transmises et traitées par un ordinateur. Le codage binaire permet l'exploitation de l'image par des algorithmes de traitement d'images et de vision artificielle.

- Impact de la numérisation sur la qualité de l'image :

La qualité d'une image numérisée dépend directement des paramètres de numérisation, notamment la résolution spatiale et la profondeur de quantification. Un échantillonnage ou une quantification insuffisante peuvent entraîner des pertes d'informations, telles que l'aliasing ou la dégradation du contraste, ce qui peut affecter la fiabilité des analyses ultérieures.

Dans le cadre des applications médicales et des systèmes de surveillance basés sur la vision artificielle, une numérisation appropriée des images est essentielle afin de garantir une analyse précise des mouvements, des postures et des comportements des patients.

2.2.4 Étapes du traitement d'images :

Le traitement d'image numérique est un processus structuré en plusieurs étapes clés, depuis l'acquisition de l'image jusqu'à son interprétation [Open CV, 2024]. Ce schéma (figure 2.3) illustre les principales phases de ce processus, permettant de comprendre la logique et l'enchaînement du traitement appliqué aux images numériques.

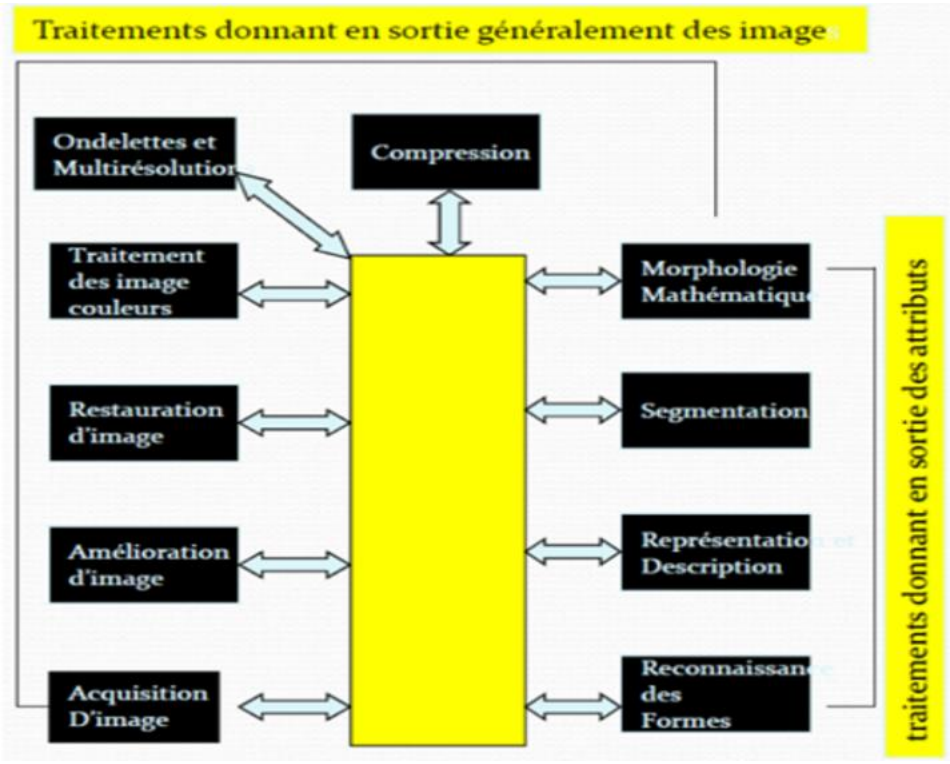


Figure 2.3 Les différentes étapes de traitement d'image

Voici les étapes fondamentales en traitement d'images numériques :

- Amélioration de l'image : L'amélioration vise à mettre en évidence des détails d'intérêt en agissant sur la luminosité, le contraste ou la netteté. Elle comprend des techniques de filtrage et d'ajustement visuel destinées à faciliter l'analyse humaine ou automatique de l'image [Gonzalez & Woods, 2018].
- Restauration de l'image : La restauration cherche à corriger des dégradations spécifiques (bruit, flou, distorsions) à l'aide de modèles mathématiques ou probabilistes. Contrairement à l'amélioration, elle est dite objective car elle s'appuie sur des modèles de dégradation connus afin de reconstruire l'image originale avec une précision accrue
- Traitement des images couleur : Le traitement des couleurs consiste à modéliser et analyser les composantes chromatiques des images numériques à l'aide d'espaces colorimétriques (RGB, HSV, YCbCr). Cette étape est devenue essentielle avec l'augmentation de l'utilisation des images couleur et permet d'améliorer la segmentation et la reconnaissance d'objets.
- Le Traitement des ondulations et des résolutions : La représentation multi-résolution, notamment basée sur les ondelettes, permet d'analyser l'image à différents niveaux de résolution. Les images sont décomposées en sous-bandes afin de faciliter la compression, la détection de détails et l'analyse pyramidale des structures visuelles.
- Compression de l'image : La compression vise à réduire l'espace de stockage requis et la bande passante nécessaire à la transmission des images, tout en préservant une qualité visuelle acceptable. Elle peut être avec perte ou sans perte, selon les exigences de l'application.
- Traitement morphologique : Le traitement morphologique repose sur des opérations structurantes appliquées aux formes afin d'extraire des composants utiles à la description géométrique des objets présents dans l'image.
- Segmentation : La segmentation consiste à partitionner l'image en régions ou objets homogènes. Elle constitue une étape clé pour la reconnaissance des formes, car elle permet d'isoler les objets d'intérêt du fond de l'image [Davies, 2018].
- Description et représentation : À l'issue de la segmentation, les régions obtenues sont décrites à l'aide de caractéristiques (forme, texture, taille). Ces descripteurs fournissent une représentation compacte et discriminante des objets.
- Identification et interprétation : Cette dernière étape consiste à attribuer une étiquette ou une classe aux objets décrits, en se basant sur leur représentation. Elle peut être réalisée à l'aide de méthodes de classification ou d'intelligence artificielle afin d'interpréter le contenu de l'image et de prendre une décision automatique.

2.3. La Vision par Ordinateur et l'IA appliquées à l'image :

2.3.1 Définition de la vision par ordinateur :

La vision par ordinateur est un domaine de l'informatique et de l'intelligence artificielle qui vise à permettre aux machines d'acquérir, de traiter et d'interpréter des informations visuelles issues d'images ou de vidéos afin de comprendre une scène ou d'agir sur un environnement donné [Gonzalez & Woods, 2018]. Elle s'appuie sur le traitement d'images, l'analyse de caractéristiques visuelles et des algorithmes d'apprentissage pour extraire des informations pertinentes à partir des données visuelles [Davies, 2018].

2.3.2 Intelligence artificielle appliquée à l'image :

L'intelligence artificielle (IA) appliquée à l'image regroupe des méthodes permettant d'automatiser l'analyse visuelle, notamment l'apprentissage automatique (Machine Learning) et l'apprentissage profond (*Deep Learning*). Contrairement aux approches classiques basées sur des règles explicites, les méthodes d'IA apprennent directement à partir des données et améliorent leurs performances avec l'expérience [Goodfellow, Bengio & Courville, 2016].

Les réseaux de neurones convolutionnels (CNN) constituent la famille de modèles la plus utilisée pour l'analyse d'images, car ils exploitent la structure spatiale des pixels pour extraire automatiquement des caractéristiques discriminantes [Goodfellow, Bengio & Courville, 2016].

2.3.3 Principales tâches de la vision par ordinateur :

Les algorithmes de vision par ordinateur sont conçus pour accomplir un ensemble varié de tâches visant à analyser, interpréter et comprendre le contenu des images et des séquences vidéo. Ces tâches constituent le cœur des systèmes de vision artificielle et sont largement utilisées dans de nombreuses applications, notamment dans le domaine médical et la surveillance intelligente [Davies, 2018].

- Reconnaissance d'images.

La reconnaissance d'images correspond à l'identification globale du contenu d'une image. Elle permet de reconnaître des personnes, des objets, des lieux ou des scènes et constitue une base fondamentale pour d'autres tâches telles que la classification, la détection et la segmentation d'images [Szeliski, 2022].

- Classification d'images.

La classification d'images consiste à attribuer une étiquette ou une classe prédéfinie à une image entière ou à des régions spécifiques de l'image. Cette tâche est largement exploitée dans les applications médicales, par exemple pour le diagnostic assisté à partir d'images radiologiques ou l'identification d'états anormaux [Davies, 2018].

- Détection d’objets.

La détection d’objets vise à localiser et identifier simultanément des objets d’intérêt dans une image ou une vidéo. Elle combine la localisation spatiale des objets et leur classification. Des architectures basées sur les réseaux de neurones convolutionnels, telles que R-CNN ou YOLO ((You Only Look Once) est un algorithme de détection d’objets basé sur les réseaux de neurones convolutionnels.), sont couramment utilisées pour cette tâche. YOLO permet notamment une détection rapide en temps réel, ce qui le rend adapté aux systèmes embarqués [Ultralytics, 2024].

- Segmentation d’images.

La segmentation d’images est une tâche plus fine que la détection d’objets, car elle opère au niveau du pixel. Elle consiste à partitionner l’image en régions homogènes afin de délimiter précisément les objets ou les zones d’intérêt. Cette approche est particulièrement utile lorsque les objets sont proches ou se chevauchent [Szeliski, 2022].

- Suivi d’objets.

Le suivi d’objets permet de suivre la trajectoire d’un objet ou d’une personne à travers une séquence vidéo. Il assure la continuité de l’identification d’un même objet dans le temps et est essentiel pour l’analyse des mouvements et des comportements [Davies, 2018].

- Compréhension des scènes.

La compréhension des scènes vise à interpréter une image à un niveau sémantique élevé en analysant les relations entre les objets, les actions et les événements. Des modèles avancés, tels que les réseaux neuronaux graphiques (GNN) et les modèles vision-langage (VLM), sont utilisés pour représenter les relations spatiales et contextuelles entre les éléments d’une scène [IEEE TPAMI, 2020].

- Reconnaissance faciale.

La reconnaissance faciale est une application spécifique de la reconnaissance d’images qui se concentre sur l’analyse des traits du visage. Elle permet l’identification ou la vérification d’une personne à partir d’images ou de vidéos. Elle est utilisée dans des systèmes d’authentification biométrique [Szeliski, 2022].

- Estimation de pose :

L’estimation de pose consiste à déterminer la position spatiale des différentes parties du corps humain afin de reconnaître les gestes et les postures. Cette tâche est particulièrement importante pour l’analyse des mouvements et la surveillance des patients alités ou semi-alités [MediaPipe Team, 2024].

- Reconnaissance optique de caractères (OCR).

La reconnaissance optique de caractères permet d'extraire automatiquement du texte à partir d'images de documents ou de scènes. Les approches modernes basées sur les CNN et les modèles de type transformer améliorent significativement la précision et la robustesse de cette tâche [Springer OCR Deep Learning, 2021].

- Génération d'images.

La génération d'images repose sur des modèles d'IA générative capables de produire de nouvelles images à partir de données d'apprentissage ou de descriptions textuelles. Parmi les modèles les plus utilisés figurent les réseaux antagonistes génératifs (GAN), les auto-encodeurs variationnels (VAE) et les modèles de diffusion [Goodfellow et al., 2014].

2.3.4 Apport de l'IA et de la vision par ordinateur en milieu médical :

Dans le domaine médical, la vision par ordinateur et l'IA permettent d'automatiser des tâches complexes telles que la surveillance des patients, l'analyse des comportements et la détection précoce de situations à risque. L'utilisation de modèles d'apprentissage profond améliore la précision et la robustesse des systèmes de surveillance, notamment pour l'analyse vidéo en temps réel des patients alités ou semi-alités.

Ces technologies constituent ainsi un élément clé des systèmes intelligents d'assistance médicale, en contribuant à améliorer la sécurité des patients et à réduire la charge du personnel soignant [Szeliski, 2022 ; Goodfellow et al., 2016].

Conclusion :

Ce chapitre a présenté les bases essentielles des systèmes embarqués, du traitement d'images numériques, de la vision par ordinateur et de l'intelligence artificielle appliquée à l'analyse visuelle. Les notions liées aux systèmes embarqués, à leur fonctionnement ainsi qu'à leur importance dans le domaine médical ont été introduites afin de mieux comprendre leur rôle dans la conception de systèmes intelligents de surveillance.

Par ailleurs, les concepts relatifs à la représentation et à la numérisation des images ainsi que les principales étapes du traitement d'images ont été abordés, permettant de comprendre le passage d'une image brute à une information exploitable. Les tâches fondamentales de la vision par ordinateur, telles que la classification, la détection, la segmentation et l'estimation de pose, ont également été présentées en mettant en évidence l'apport des techniques d'apprentissage automatique et d'apprentissage profond.

L'ensemble de ces concepts constitue un socle théorique indispensable pour la conception et le développement du système de surveillance proposé, qui repose sur l'intégration des systèmes embarqués et des techniques avancées d'analyse d'images.

Chapitre III : Conception et réalisation

Introduction :

Un séjour prolongé au lit met les patients face à des risques significatifs pour leur santé et leur sécurité, allant de la formation d'escarres dues à une pression continue sur les tissus, aux complications cardio-respiratoires engendrées par l'immobilité, ainsi qu'à un risque accru de chutes graves lors des tentatives de lever sans assistance. Face à ces risques, la surveillance humaine montre des limites logistiques notables, car il est irréaliste pour le personnel soignant de réaliser une observation visuelle ininterrompue 24h/24, alors que les systèmes d'appel traditionnels nécessitent l'intervention active d'un patient parfois confus ou affaibli.

C'est dans ce cadre délicat que s'inscrit notre initiative, dont l'objectif central est de développer un système de surveillance à distance automatisé, intelligent et non intrusif, capable d'évaluer en temps réel l'historique des postures afin de prévenir les complications physiques, tout en générant des alertes immédiates lors de situations d'urgence, comme un risque de chute du lit.

3.1 Patients semi-alités et méthodes de surveillance :

La surveillance des patients semi-alités repose traditionnellement sur l'observation humaine effectuée par le personnel soignant ou les proches. Bien que cette approche permette une évaluation directe de l'état du patient, elle présente des limites importantes, notamment une disponibilité restreinte, un risque d'erreur humaine et une charge de travail élevée. L'absence de surveillance continue peut également retarder la détection de situations à risque.

Afin de dépasser ces limites, des méthodes de surveillance avancées basées sur l'apprentissage automatique et l'apprentissage profond ont été développées. Ces méthodes exploitent la vision par ordinateur et l'intelligence artificielle (IA) pour analyser automatiquement des images ou des séquences vidéo et assurer une surveillance non intrusive des patients semi-alités.

Parmi les principales méthodes utilisées dans les systèmes de détection basés sur l'apprentissage, on distingue :

- **Détection d'objets par réseaux de neurones convolutionnels (CNN)** : des modèles tels que YOLO, SSD ou Faster R-CNN permettent de détecter et de localiser rapidement la présence du patient dans la scène vidéo. Ces approches sont particulièrement adaptées aux applications temps réel en raison de leur rapidité et de leur précision.
- **Estimation de pose humaine** : cette méthode repose sur l'extraction automatique des points clés du corps humain (articulations, membres, tronc) à partir des images. Elle permet de représenter la posture du patient sous forme squelettique, facilitant ainsi l'analyse posturale indépendamment de l'arrière-plan ou de l'éclairage.

- **Classification des postures par apprentissage supervisé** : les caractéristiques extraites (points squelettiques, distances angulaires, coordonnées) sont exploitées par des algorithmes d'apprentissage automatique tels que les CNN, les forêts aléatoires (*Random Forest*) ou les machines à vecteurs de support (SVM) afin de classer automatiquement les différentes postures du patient (couché, semi-assis, assis, etc.).
- **Analyse temporelle des mouvements** : pour la surveillance continue, des modèles intégrant la dimension temporelle, tels que les réseaux de neurones récurrents (RNN) ou les architectures basées sur les LSTM, peuvent être utilisés afin d'analyser l'évolution des postures dans le temps et de détecter des comportements anormaux ou prolongés.
- **Systèmes d'alerte intelligents** : les résultats issus des modèles de détection et de classification sont exploités pour générer automatiquement des alertes en cas de posture à risque ou d'absence de mouvement prolongée, contribuant ainsi à améliorer la réactivité du personnel médical.

L'intégration de ces méthodes avancées basées sur l'apprentissage permet d'assurer une surveillance automatique, continue et fiable des patients semi-alités, tout en réduisant la dépendance à l'intervention humaine et en améliorant la qualité de la prise en charge médicale.

3.2. Architecture Globale du Système :

L'architecture proposée pour notre système de télésurveillance a été conçue selon une approche modulaire permettant d'assurer un traitement rapide et continu des données en temps réel. Le fonctionnement général repose sur plusieurs couches technologiques interconnectées, qui assurent le transfert des informations depuis la phase d'acquisition jusqu'à leur affichage sur l'interface destinée au personnel médical. Le pipeline global du système se compose des étapes suivantes :

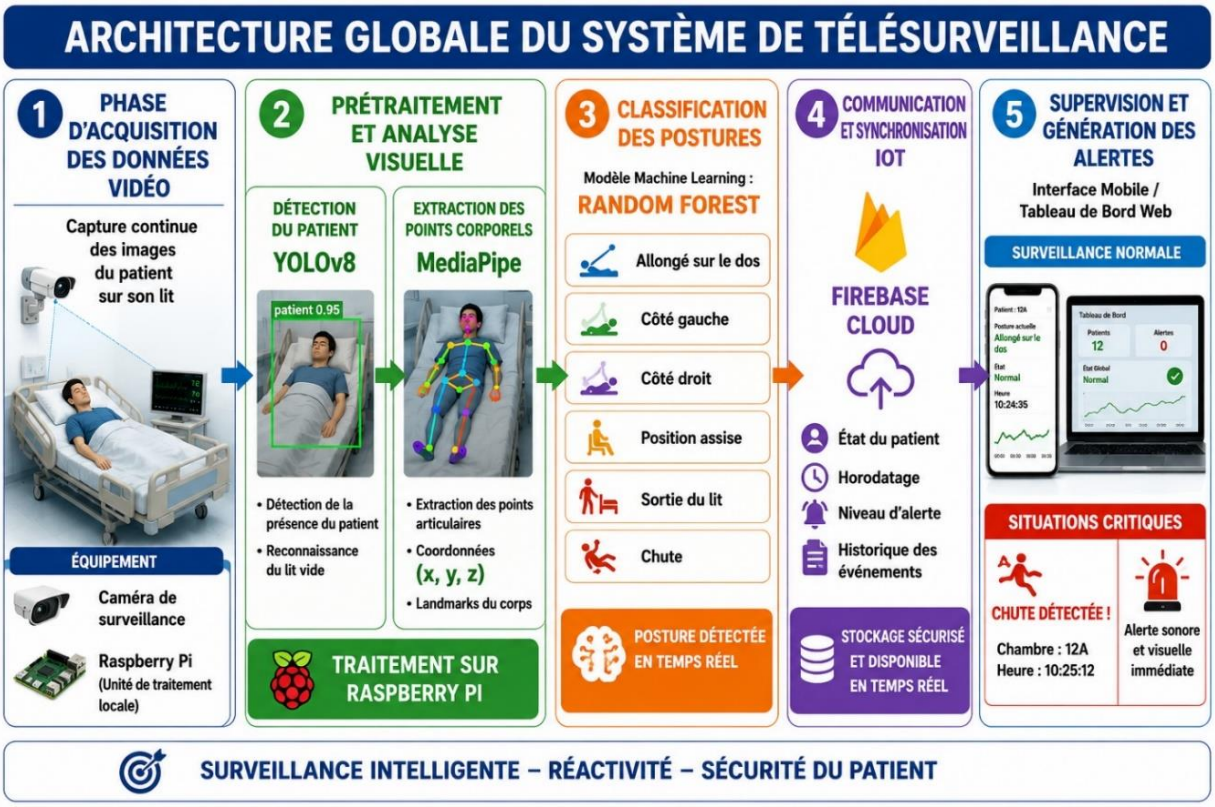


Figure 3.1 : Architecture Globale du Système

3.3 Création de la base de données :

L'efficacité d'un modèle d'apprentissage automatique repose majoritairement sur la qualité et la pertinence des données utilisées pour son entraînement. Dans le cadre de notre projet, les bases de données publiques existantes ne répondaient pas exactement à nos besoins spécifiques (surveillance de postures d'un patient semi-alités depuis un angle précis). Nous avons donc pris la décision de construire notre propre base de données (Dataset customisé). Ce processus rigoureux s'est déroulé en trois étapes fondamentales :

3.3.1. Acquisition des données vidéo et protocole expérimental :

La qualité de notre système repose sur un protocole d'acquisition rigoureux visant à simuler des conditions réelles en milieu hospitalier. Pour ce faire, nous avons sollicité la participation de neuf (09) acteurs de différentes tranches d'âge, garantissant ainsi une diversité morphologique nécessaire à la robustesse de notre modèle. L'enregistrement a été effectué à l'aide d'un smartphone haute résolution, positionné selon divers angles de vue pour capturer la scène de manière exhaustive. Afin d'augmenter la complexité du dataset et de tester la capacité du système à détecter le corps humain, malgré des obstacles visuels (problématique d'occlusion), nous avons varié les conditions expérimentales. Certaines séquences ont été filmées alors que les acteurs étaient partiellement recouverts d'une couverture légère, simulant l'état réel d'un patient alité. D'autres séquences ont été enregistrées sans couverture pour obtenir des données de référence claires. Pendant ces sessions, il a été demandé aux participants d'exécuter et de maintenir cinq postures spécifiques : décubitus

dorsal (sur le dos), décubitus latéral (gauche et droit), position assise, ainsi que des simulations de chutes. Ces enregistrements, sauvegardés au format MP4, ont constitué la source primaire à partir de laquelle, nous avons extrait les caractéristiques biomécaniques pour générer notre fichier final au format CSV.



Figure 3.2 : Échantillons des séquences vidéo acquises illustrant la diversité des acteurs.

3.3.2. Extraction des caractéristiques squelettiques avec MediaPipe :

Après l'enregistrement des vidéos, il n'était pas raisonnable de donner les images brutes directement au modèle de classification, car cela exigerait une énorme puissance de calcul. Nous avons donc employé la bibliothèque MediaPipe (Pose) développée par Google. Cet outil de vision par ordinateur a examiné chaque image de nos vidéos afin d'identifier le corps humain et d'extraire 33 points d'articulation (Landmarks). De ce fait, l'information visuelle dense a été convertie en un ensemble léger de coordonnées spatiales (X, Y, Z et visibilité) illustrant la posture biomécanique précise à un moment donné.



Figure 3.3 : Extraction des 33 points clés (Landmarks) sur un sujet de notre base de données.

3.3.3. Structuration et purification des données (Format CSV) :

Avant l'entraînement du modèle, les données extraites ont été structurées et nettoyées afin de garantir leur qualité. Nous avons créé un script Python qui enregistre ces vecteurs numériques dans un fichier au format tabulaire CSV. Chaque ligne de ce document représente une image de la vidéo (une "frame"), incluant les coordonnées des points articulaires ainsi qu'une étiquette (Label) signalant la posture associée. Avant l'entraînement, une étape de purification a été réalisée pour supprimer les lignes comportant des valeurs extrêmes ou absentes (par exemple, lorsque le corps était complètement hors du champ de vision), assurant ainsi un jeu de données sain et adapté à la modélisation.

Étant donné que les algorithmes d'apprentissage automatique nécessitent des données numériques pour fonctionner, nous avons procédé à une étape d'encodage des étiquettes (Data Labeling). Les postures ont été converties en classes numériques cibles comme suit : **0** (Sur le dos), **1** (Côté droit), **2** (Côté gauche), **3** (Assis), et **4** (Sortie/Chute).



Figure 3.4: Extrait du fichier CSV contenant les coordonnées articulaires et les labels

- X **(Axe horizontal) :** Cette valeur représente la position latérale du point d'articulation. Elle est normalisée entre 0.0 et 1.0 : une valeur de 0.0 correspond au bord gauche de l'image, tandis que 1.0 correspond au bord droit.
- Y **(Axe vertical) :** Cette valeur indique la position du point sur la hauteur de l'image. Elle est également normalisée : 0.0 représente le bord supérieur (le haut) de l'image et 1.0 représente le bord inférieur (le bas).
- Z **(Profondeur) :** Cette coordonnée exprime la profondeur du point par rapport à un plan de référence (généralement situé au niveau du bassin). Une valeur plus petite signifie que le membre est plus proche de la caméra, tandis qu'une valeur plus élevée indique qu'il est plus éloigné. C'est cette donnée qui permet de distinguer, par exemple, si un bras est tendu vers l'avant ou s'il reste le long du corps.
- **Visibilité (v) :** Il s'agit d'un score de confiance compris entre 0 et 1. Il indique la probabilité que le point d'articulation soit réellement visible dans l'image et non masqué par un obstacle, comme une couverture. Un score proche de 1 signifie que le point est clairement identifié, alors qu'un score proche de 0 signifie qu'il est probablement caché (occlusion).

Constituer un Dataset robuste, varié et équilibré. Notre base de données finale est composée de 10 783 images (frames) exploitables, réparties de la manière suivante :

Décubitus dorsal (Dos) :	Décubitus latéral droit :	Décubitus latéral gauche :	Position assise :	Situation critique (Chute/Sortie) :
4966 images	2434 IMAGE	1931 images	878 images	573 images

Tableau 3.1 Nombre de frames (images) extraites pour chaque position du patient.

Cette richesse de données garantit une phase d'apprentissage optimale pour notre modèle de classification.

3.4. Modélisation et Apprentissage Automatique (*Machine Learning*) :

Une fois les données spatiales extraites et structurées sous forme de vecteurs numériques, l'étape suivante consiste à développer un modèle d'intelligence artificielle capable d'interpréter ces coordonnées pour en déduire la posture du patient. Pour ce faire, nous nous sommes appuyés sur les techniques d'apprentissage supervisé (*Supervised Learning*).

3.4.1. Justification du choix de l'algorithme (*Random Forest*) :

Pour répondre à notre problématique de classification multiclasse (cinq postures différentes), nous avons opté pour l'algorithme des Forêts Aléatoires (*Random Forest*). Ce choix n'est pas fortuit, mais repose sur plusieurs justifications techniques :

- **Adaptation aux données tabulaires :** Contrairement aux modèles de *Deep Learning* (comme les CNN) qui est excellent sur des images brutes, le *Random Forest* est particulièrement performant sur des données structurées et tabulaires, comme notre fichier CSV contenant les coordonnées (X, Y, Z) de MediaPipe.
- **Robustesse face au sur-apprentissage (*Overfitting)* :** En combinant les prédictions de multiples arbres de décision indépendants (méthode d'*Ensemble Learning*), cet algorithme réduit considérablement la variance et évite de mémoriser "par cœur" les données d'entraînement.
- **Rapidité d'inférence en temps réel :** Étant donné que notre système vise à générer des alertes immédiates en cas de danger, la rapidité d'exécution est primordiale. Le *Random Forest* offre d'excellents temps de réponse, parfaitement adaptés à notre application de télésurveillance en direct.

3.4.2. Phase d'entraînement et séparation des données (*Train/Test*) :

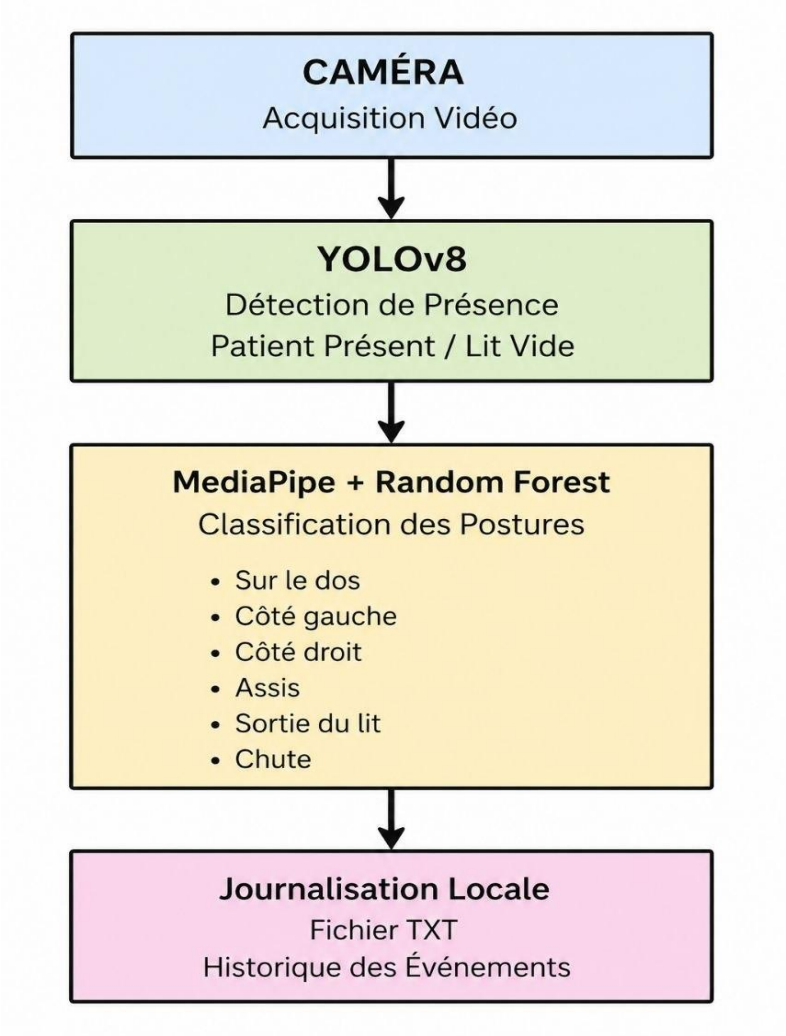
Pour construire et évaluer objectivement notre modèle, nous avons utilisé la bibliothèque Python *Scikit-Learn.* Le processus s'est déroulé selon les étapes standards du Machine Learning :

- **Séparation des données (*Data Splitting*) :** Afin de garantir que notre modèle puisse se généraliser sur de nouvelles images qu'il n'a jamais vues, nous avons divisé notre dataset en deux sous-ensembles distincts. En général, 80 % des données ont été allouées à la phase d'entraînement (*Train set*) pour permettre au modèle d'apprendre les relations complexes entre les coordonnées squelettiques et les postures. Les 20 % restants ont été utilisé pour la phase de test (*Test set*).
- **Entraînement :** Lors de cette phase, l'algorithme Random Forest a construit ses multiples arbres de décision en analysant les caractéristiques biomécaniques (*features*) associées à chaque étiquette (*label*) de posture.

- **Évaluation préliminaire :** Une fois l'entraînement terminé, le modèle a été confronté aux données du *Test set*. Ses prédictions ont été comparées aux étiquettes réelles afin de calculer les métriques de performance objective. (Remarque : Les résultats détaillés, tels que la précision globale et la matrice de confusion, feront l'objet d'une analyse approfondie dans le chapitre suivant).

3.5. Pipeline du Système de Surveillance en Temps Réel :

Le cœur de notre solution repose sur un pipeline de traitement séquentiel et hybride, exécuté en temps réel. Ce flux de travail a été spécialement conçu pour transformer un flux vidéo brut en une information médicale exploitable instantanément, en combinant intelligemment la détection d'objets, l'analyse biomécanique et la classification par intelligence artificielle.



3.5.1. Détection de la présence du patient avec YOLOv8 (Optimisation) :

La première étape du pipeline consiste à analyser chaque trame (*frame*) vidéo à l'aide du modèle YOLOv8. L'objectif principal de cette phase n'est pas de définir la posture, mais de vérifier la présence et de localiser le patient dans la chambre. Lorsqu'une personne est détectée, le système trace une boîte englobante

(*Bounding Box*) de couleur verte délimitant le corps du patient. L'intégration de YOLOv8 à ce stade agit comme un filtre d'optimisation majeur : si le modèle ne détecte personne (lit vide), le système suspend les étapes suivantes.

Cela permet d'éviter l'exécution de calculs lourds sur un arrière-plan vide, économisant ainsi considérablement les ressources matérielles (CPU/GPU) et éliminant le risque de fausses détections squelettiques (Faux Positifs).

3.5.2. Classification continue de la posture :

Une fois la présence du patient confirmée par YOLOv8, le système déclenche la seconde étape du pipeline. L'image est transmise à la bibliothèque MediaPipe qui procède à l'extraction instantanée des coordonnées des points articulaires. Ces données spatiales sont immédiatement restructurées et injectées dans notre modèle de classification *Random Forest.* En quelques millisecondes, le modèle effectue une inférence (prédiction) pour déterminer la posture actuelle du patient (Sur le dos, Côté gauche, Côté droit, Assis, ou Chute). Ce processus s'exécute en boucle continue, permettant un suivi fluide des mouvements, accompagné d'un affichage textuel directement sur le retour vidéo pour le contrôle visuel.

3.5.3. Mécanisme de journalisation et d'archivage local (Redondance) :

Dans un environnement hospitalier critique tel que le service de réanimation, une perte de connexion Internet (empêchant la transmission vers le Cloud) ne doit en aucun cas interrompre la traçabilité de l'état du patient. Pour pallier ce risque, nous avons implémenté un mécanisme de redondance locale. Le système intègre une logique de détection des transitions : à chaque fois que le patient change de posture (par exemple, passage de "Sur le dos" à "Assis" ou "Chute"), le système enregistre automatiquement cet événement. Une ligne contenant la nouvelle posture accompagnée d'un horodatage précis (*Timestamp*) est inscrite dans un fichier texte local nommé « **journal_surveillance.txt** ». Ce fichier agit comme une véritable "boîte noire" (*Data Logger),* garantissant au personnel médical un accès sécurisé et infalsifiable à l'historique de la chambre, indépendamment de l'état du réseau.

3.6. Architecture IoT et Interface de Monitoring :

Pour qu'un système de surveillance médical soit véritablement utile, la détection locale d'une anomalie doit être communiquée instantanément au personnel soignant. Cette exigence nous a conduits à développer une architecture IoT (*Internet of Things*), reliant notre système de vision par ordinateur local à un tableau de bord web interactif baptisé.

3.6.1. Synchronisation Cloud via *Firebase Realtime Database:*

Le lien de communication entre le script d'analyse vidéo et l'interface utilisateur est assuré par *Firebase Realtime Database*, une solution Cloud orientée NoSQL hébergée par Google. À chaque changement d'état du patient, le système d'intelligence artificielle génère un paquet de données (*Payload*) au format JSON contenant : la posture détectée, l'heure exacte (*Timestamp*) et le statut de l'alerte. Afin de ne pas ralentir le traitement des images (baisse des FPS *Frames Per Second*, c'est-à-dire le nombre d'images traitées par seconde), cette transmission de données a été implémentée de manière asynchrone (via la bibliothèque **threading** en Python). Ainsi, l'envoi des requêtes réseau s'effectue en arrière-plan (*Background Process*), garantissant une surveillance vidéo fluide et sans interruption.

3.6.2. Développement du tableau de bord multi-chambres :

Pour l'exploitation des données par le personnel médical, nous avons conçu une application web centralisée à l'aide du *framework Streamlit*, enrichie avec du code HTML/CSS personnalisé pour offrir une expérience utilisateur (UI/UX) de niveau professionnel. Le tableau de bord est structuré en deux vues principales :

- **Vue globale (*Ward Overview)* :** Permet à l'infirmier de surveiller simultanément l'état de plusieurs chambres (ex. Chambres 101, 102 et 103) sous forme de cartes dynamiques (*Cards*).
- **Vue détaillée (*Monitoring Log)* :** Un espace dédié à chaque chambre, affichant les informations d'identité du patient ainsi qu'un journal tabulaire détaillé des changements de postures de la session en cours.

3.6.3. Système de gestion des alertes et notifications sonores :

Dans un service de réanimation ou de cardiologie, la réactivité est vitale en cas de chute ou de sortie de lit non assistée. Pour cela, le tableau de bord intègre un double mécanisme d'alerte critique :

- **Alerte visuelle (*Feedback UI*) :** Dès que la base de données Firebase bascule l'état d'alerte à "OUI", la carte de la chambre concernée sur le tableau de bord change de style (bordures rouges) et affiche une bannière d'avertissement exigeant une intervention immédiate.
- **Alerte sonore automatique :** Pour garantir la réactivité du personnel même s'il ne regarde pas l'écran, l'interface déclenche automatiquement une alarme sonore puissante (sirène d'ambulance) intégrée en boucle via une balise HTML **<audio>**. Cette alarme persiste tant que la situation critique n'est pas prise en charge.

3.7. Environnement de Développement et Outils Utilisés:

La réalisation de ce système de télésurveillance hybride a nécessité l'intégration de plusieurs outils logiciels, *frameworks* et plateformes modernes. L'ensemble du projet a été développé en utilisant l'écosystème technologique suivant :

- **Python :** Le langage de programmation principal choisi pour sa richesse en bibliothèques d'intelligence artificielle et sa flexibilité.
- **Visual Studio Code (VS Code) :** L'environnement de développement intégré (IDE) utilisé pour l'écriture, le débogage et l'exécution des scripts de vision par ordinateur.
- **OpenCV :** Bibliothèque incontournable, utilisée pour la capture des flux vidéo, le traitement d'images et l'affichage des annotations (*Bounding boxes* et texte) en temps réel.
- **MediaPipe (Google) :** *Framework* de *Machine Learning* utilisé pour l'extraction rapide et précise des points clés (*Landmarks*) du squelette humain.
- **Ultralytics (YOLOv8) :** Modèle de détection d'objets de pointe, implémenté pour repérer la présence du patient et optimiser les ressources matérielles.
- **Scikit-Learn :** Bibliothèque d'apprentissage automatique utilisée pour la construction, l'entraînement et l'évaluation de notre modèle de classification (*Random Forest*).
- **Firebase Realtime Database :** Plateforme Cloud utilisée comme pont IoT pour la synchronisation instantanée des états et des alertes entre la caméra, le tableau de bord web et l'application mobile.
- **Streamlit :** *Framework open-source* utilisé pour concevoir le tableau de bord interactif web pour le monitoring centralisé.
- **Flutter et Android Studio :** Utilisés pour concevoir et développer l'application mobile *cross-platform* destinée au personnel soignant. Cet environnement nous a permis de créer des interfaces utilisateur (UI) fluides et d'intégrer le système de notifications push pour une gestion mobile des alertes.
- **Le micro-ordinateur Raspberry Pi (Cible Matérielle) :** Bien que le développement et l'entraînement du modèle aient été effectués sur une station de travail, le système est conçu pour être déployé sur un micro-ordinateur de type Raspberry Pi 4. Ce choix se justifie par la nécessité d'avoir une unité de calcul compacte, basse consommation et capable d'être installée directement dans la chambre du patient. Le Raspberry Pi fait office de nœud de calcul 'Edge', récupérant le flux de la caméra USB ou PiCam pour exécuter localement notre pipeline IA (YOLO + MediaPipe + RF) avant d'envoyer uniquement les résultats vers Firebase.

Conclusion

Ce chapitre a été consacré à la conception détaillée et à la réalisation pratique de notre système de télésurveillance pour patients alités. En adoptant une approche hybride, nous avons réussi à combiner l'efficacité de détection de YOLOv8, la précision biomécanique de MediaPipe et la robustesse de classification du *Random Forest*. De plus, la création d'une base de données personnalisée et l'intégration d'une architecture IoT – couplée à un tableau de bord web et une application mobile – démontrent la viabilité et l'ergonomie de notre solution dans un environnement hospitalier critique.

Maintenant que l'architecture globale du système est en place et fonctionnelle, il est impératif d'évaluer ses performances réelles. Le chapitre suivant sera ainsi dédié à l'analyse des résultats obtenus, à l'évaluation de la précision du modèle de *Machine Learning* et à la discussion des temps de réponse du système global.

Chapitre IV : Résultats et discussion

Introduction :

Après avoir détaillé les choix architecturaux et la phase de réalisation dans le chapitre précédent, ce quatrième chapitre est dédié à l'évaluation expérimentale et à la validation de notre système de télésurveillance pour patients semi-alités. La conception d'une solution hybride, combinant la vision par ordinateur (YOLOv8 et MediaPipe) et l'apprentissage automatique (Random Forest), nécessite une analyse rigoureuse de ses performances avant tout déploiement en milieu clinique.

L'objectif de ce chapitre est double : d'un part, évaluer de manière quantitative la précision et la robustesse de notre modèle d'Intelligence Artificielle face à différentes postures et situations complexes (telles que les occlusions partielles) ; d'autre part, valider l'efficacité du système de bout en bout, depuis la captation vidéo jusqu'à la réception des alertes critiques sur l'interface de suivi et l'application mobile. À travers cette démarche, nous mettrons en évidence les points forts de notre solution, tout en discutant de ses limites actuelles et des perspectives d'amélioration.

4.1. Évaluation du Modèle d'Intelligence Artificielle :

4.1.1. Les métriques d'évaluation :

Pour quantifier de manière rigoureuse les performances de notre *classifieur Random Forest,* nous nous appuyons sur quatre métriques statistiques fondamentales issues de la matrice de confusion : l'Exactitude (*Accuracy*), la Précision, le Rappel (*Recall*) et le *F1-Score*.

- **L'Exactitude (*Accuracy)* :** Représente la proportion globale de prédictions correctes parmi l'ensemble des cas testés.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **La Précision :** Mesure la capacité du modèle à ne pas étiqueter positivement un échantillon qui est négatif. Autrement dit, lorsque le modèle prédit une posture, quelle est la probabilité qu'il s'agisse de la bonne ?

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

- **Le Rappel (*Recall)* :** Évalue la capacité du modèle à détecter tous les cas positifs réels. Il indique la proportion de postures réelles correctement identifiées par le système.

$$\text{Rappel} = \frac{TP}{TP + FN}$$

- **Le F1-Score :** Représente la moyenne harmonique de la Précision et du Rappel. C'est l'indicateur le plus robuste en cas de déséquilibre entre les classes.

$$\text{F1-Score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

4.1.2. Précision Globale du modèle (*Accuracy globale*) :

À l'issue de la phase d'évaluation sur notre ensemble de test indépendant (représentant 20% du dataset global, soit 2 157 frames), notre modèle *Random Forest* a obtenu une Précision Globale (*Accuracy*) de 87.34%.

Dans le contexte d'un déploiement sur un micro-ordinateur à ressources limitées (*Raspberry Pi 4)* et face à un flux vidéo réel non standardisé, ce score démontre la robustesse de l'approche hybride choisie. Le modèle parvient à classifier correctement près de 9 postures sur 10, ce qui est hautement satisfaisant pour une application médicale de télésurveillance.

$$\text{F1-Score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

(Où TP = Vrais Positifs, TN = Vrais Négatifs, FP = Faux Positifs, FN = Faux Négatifs).

4.1.3. Analyse de la Matrice de Confusion

La matrice de confusion générée permet d'observer la distribution des prédictions par rapport aux classes réelles et d'identifier précisément les zones de convergence et de confusion algorithmique.

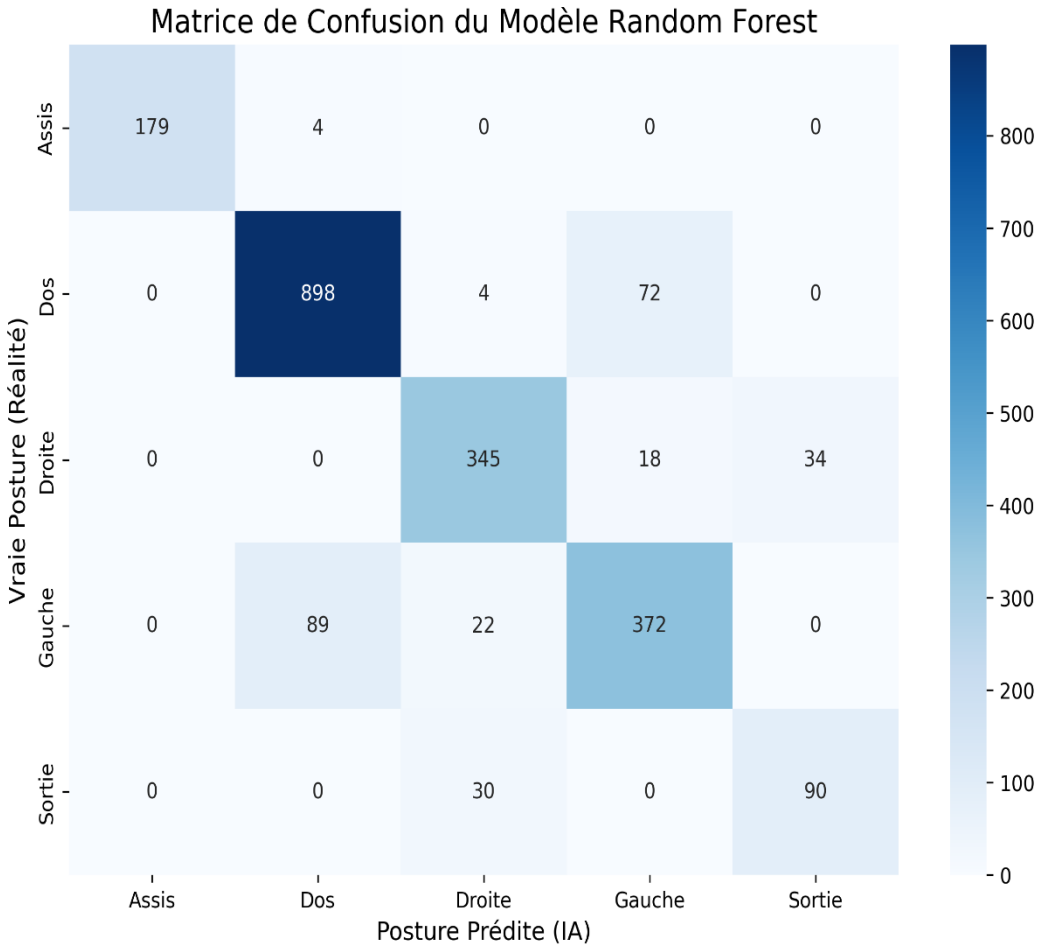


Figure 4.1 : Matrice de confusion du modèle Random Forest.

L'examen de la diagonale principale de la figure 4.1 confirme que la grande majorité des trames de test sont correctement assignées à leurs classes respectives. On note cependant des interférences mineures, notamment entre les classes latérales ("Gauche" et "Droite") et la classe "Dos", un phénomène attribuable aux similitudes morphologiques des coordonnées squelettiques lorsque le patient effectue une transition de posture.

4.1.4 Rapport de classification détaillé par posture

Le tableau suivant présente le rapport de performance détaillé extrait par classe à l'aide de la librairie *Scikit-Learn :*

Posture (Classe)	Précision	Rappel (Recall)	F1-Score
Assis	1.00	0.98	0.99
Dos	0.91	0.92	0.91
Droite	0.86	0.87	0.86
Gauche	0.81	0.77	0.79
Sortie	0.73	0.75	0.74
Moyenne pondérée	**0.87**	**0.87**	**0.87**

Tableau 4.1: Résultats de l'évaluation du classifieur par classe.

4.2. Interprétation et Discussion des Résultats

4.2.1. Les points forts et performances optimales :

Les résultats du tableau démontrent une efficacité remarquable sur deux postures clés :

- **La posture "Assis" (F1-Score de 99%) :** Cette quasi-perfection s'explique par la signature biomécanique unique de cette position. L'angle formé entre le tronc et les membres inférieurs à 90° offre des variations de coordonnées (x, y) majeures et non ambiguës pour MediaPipe, éliminant tout risque de fausse classification.
- **La posture "Dos" (F1-Score de 91%) :** Représentant la classe majoritaire avec un support conséquent de 974 trames de test, cette posture a été parfaitement assimilée par l'arbre de décision. La visibilité simultanée des deux épaules et des deux hanches fournit au classifieur des repères stables et symétriques.

4.2.2. Les limites initiales du modèle et résolution du phénomène d'occlusion :

L'évaluation initiale de notre modèle a mis en évidence des scores perfectibles pour les classes « Gauche » (79%) et « Sortie » (74%). Une analyse approfondie des erreurs a révélé que ces limitations étaient directement liées à deux facteurs scientifiques majeurs :

- **Le problème d'occlusion (Masquage par la literie) :** En situation réelle, lorsqu'un patient se retourne sur le côté ou s'allonge sous une couverture/drap, les membres inférieurs et une partie du tronc

subissent une occlusion visuelle. Face à ces données manquantes, l'algorithme MediaPipe tend à extrapoler de manière erronée les coordonnées des points cachés, générant un bruit numérique important (coordonnées aberrantes) qui induit en erreur le classifieur *Random Forest*.

- **Le déséquilibre des classes (Class Imbalance) :** La posture « Sortie » (mouvements rapides, chutes ou levers du lit) disposait initialement d'un support plus faible dans notre base de données (120 frames), limitant l'apprentissage de la variabilité morphologique de cette position.

1. **Approche corrective et ré-entraînement du modèle :**

Pour surmonter cette limite critique et valider la robustesse de notre système face aux contraintes hospitalières réelles, nous avons mis en place une procédure corrective en trois étapes :

- **Enrichissement de la Base de Données (Data Augmentation) :** Nous avons enregistré et intégré de nouvelles séquences vidéos spécifiques mettant en scène des patients en situation d'occlusion (corps partiellement masqué par des couvertures dans différentes positions).
- **Exploitation de l'indice de visibilité (*Visibility Score*) :** Lors de l'extraction des descripteurs via MediaPipe, nous avons exploité le paramètre de visibilité (v) associé à chaque *landmark* (allant de 0.0 à 1.0). Le script de capture a été configuré pour enregistrer fidèlement ces valeurs proches de 0 pour les membres masqués, permettant ainsi au Random Forest d'apprendre mathématiquement qu'une absence de visibilité sur certains points inférieurs correspond grammaticalement à un patient couvert et non à une anomalie de posture.
- ***Fine-Tuning* du *Random Forest* :** Le modèle a été ré-entraîné sur ce dataset enrichi (portant le support de test à 2 160 *frames*).

2. **Impact sur les performances globales :**

Cette approche itérative a permis de redresser significativement la précision du système. Comme le démontrent les derniers résultats d'évaluation, l'exactitude globale (*Accuracy*) est passée de 87.34% à 87.59%.

Plus important encore, le taux de reconnaissance (F1-Score) de la classe critique « Sortie » a progressé à 0.75 avec un Rappel (*Recall*) de 0.77, prouvant que le modèle est désormais capable de discriminer efficacement un mouvement de sortie de lit d'un simple ajustement de la couverture.

4.2.3. Validation visuelle en temps réel et scénarios de test :

Afin de valider de manière empirique et visuelle le comportement en temps réel de notre modèle hybride (YOLOv8 + MediaPipe + Random Forest), nous avons soumis le système à une série de tests sur des séquences vidéo enregistrées dans des conditions similaires au milieu hospitalier. Cette évaluation visuelle

permet de confirmer la pertinence des métriques quantitatives obtenues précédemment. Nous présentons ci-dessous quatre scénarios cliniques caractéristiques.

Scénario A : Détection de la posture nominale "Assis" :

Dans ce premier scénario, le patient est positionné en position assise sur le bord ou au centre du lit. Comme le montre la **Figure 4.2**, l'algorithme YOLOv8 réussit parfaitement à localiser le patient dans l'espace de la chambre. Simultanément, la bibliothèque MediaPipe extrait la topologie du squelette.

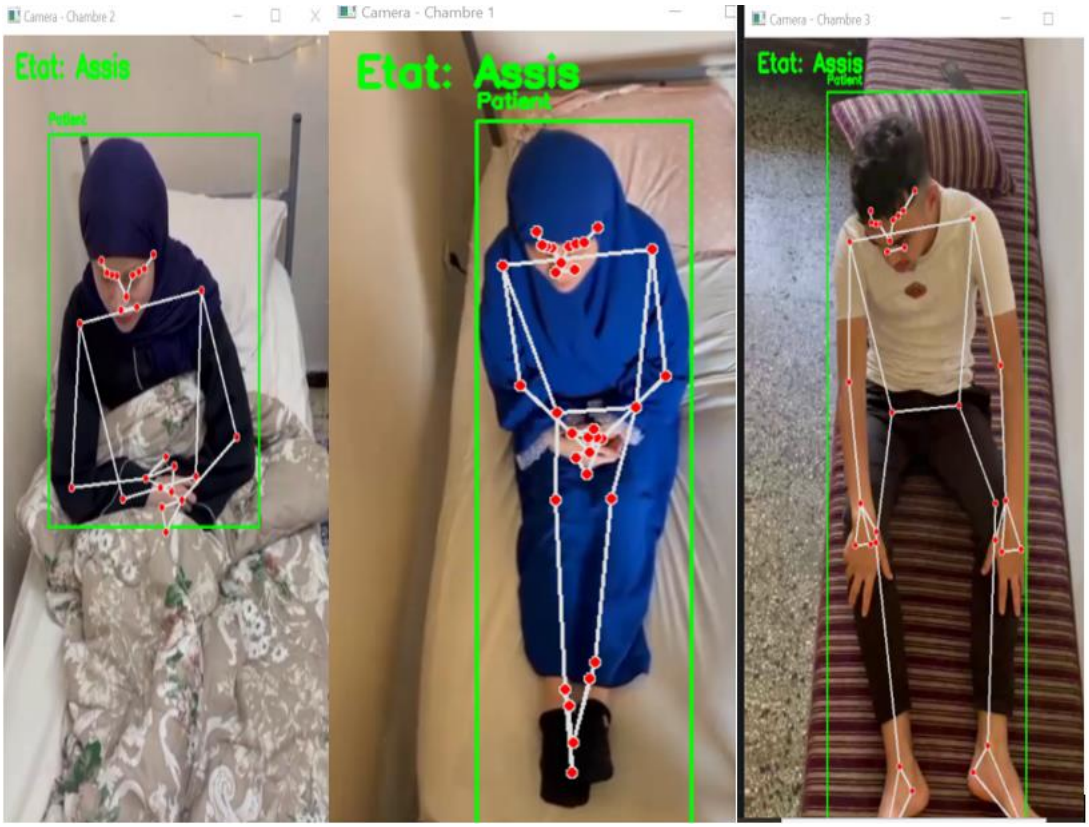


Figure 4.2 : Extraction du squelette et classification en temps réel de la posture "Assis".

Analyse de l'image : L'angulation marquée entre le tronc et les segments inférieurs permet au classifieur *Random Forest* de renvoyer une prédiction stable avec un indice de confiance maximal, confirmant le F1-score de 99% mesuré dans notre rapport de performance.

Scénario B : Détection de la posture "Dos" (Décubitus dorsal)

Le deuxième scénario illustre la situation de repos la plus fréquente où le patient est allongé sur le dos.

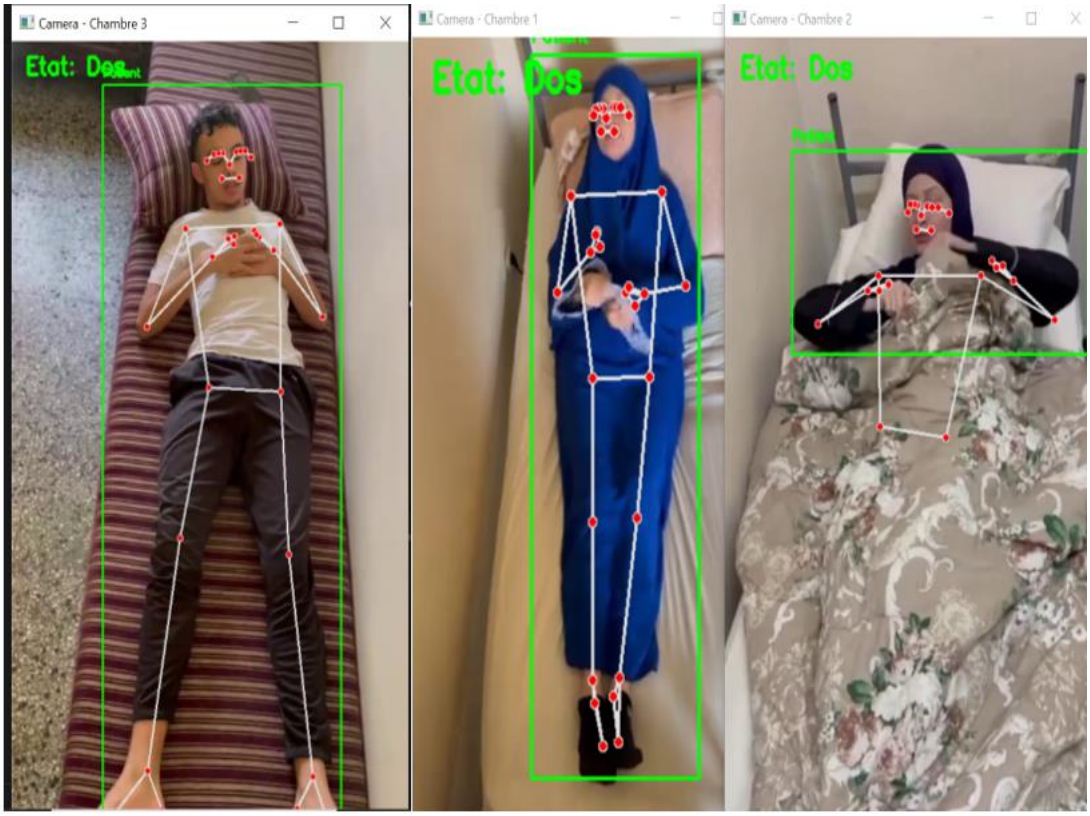


Figure 4.3 : Capture du flux vidéo montrant la détection correcte de la posture "Dos".

Analyse de l'image : Bien que la surface d'appui aplatisse la perspective visuelle, les repères anatomiques des épaules et des hanches restent symétriques et parfaitement visibles pour la caméra. Le système étiquette la trame courante comme "Dos" de manière ininterrompue.

Scénario C : Résolution d'une occlusion partielle sous la couverture ("Droite / Gauche")

Ce scénario représente l'une des contributions majeures de notre travail : la robustesse face au masquage par la literie. Le patient est allongé sur le côté (position latérale) et recouvert d'un drap ou d'une couverture opaque, dissimulant ses membres inférieurs.



Figure 4.4 : Performance du système en situation d'occlusion partielle par la couverture.

Analyse de l'image : Malgré l'occlusion visuelle de la partie inférieure du corps, l'intégration des scores de visibilité (*Visibility Score*) de MediaPipe dans le vecteur de caractéristiques permet au modèle *Random Forest* de comprendre que l'absence de certains *landmarks* est due à la couverture. Le système évite ainsi les prédictions aberrantes et maintient un étiquetage postural conforme à la réalité médicale.

Scénario D : Déclenchement de l'alerte critique "Sortie de lit"

Le dernier scénario valide la fonction de sécurité la plus critique du système : la détection d'une tentative de sortie de lit ou d'une chute potentielle.

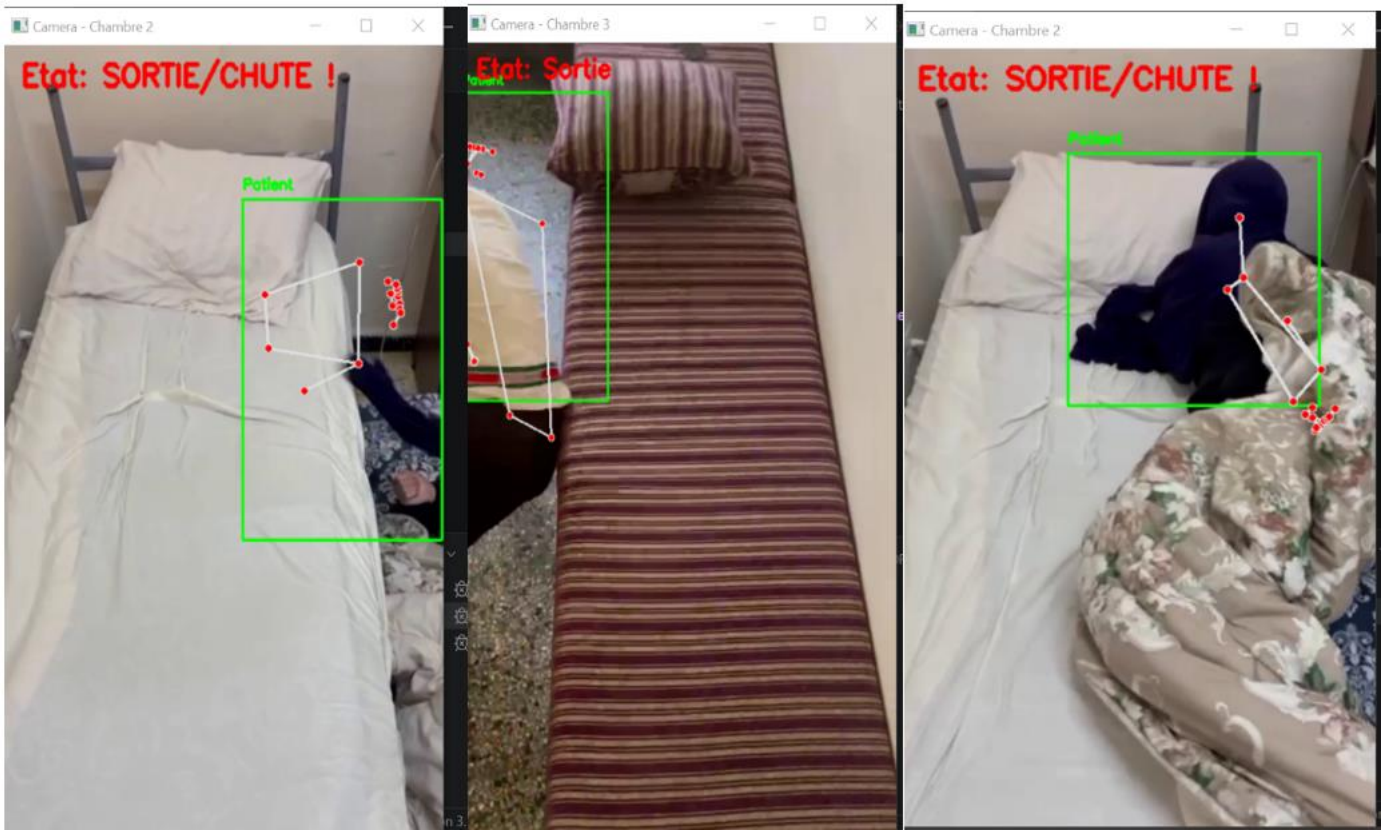


Figure 4.5 : Détection instantanée d'un événement critique de type "Sortie de lit".

Analyse de l'image : Lorsque le patient bascule hors de la géométrie standard du lit, les coordonnées spatiales globales subissent une rupture de tendance brutale. Le *Random Forest* identifie immédiatement la classe « Sortie ». Cette transition logicielle va instantanément forcer la mise à jour de la base de données Firebase pour alerter le personnel soignant en moins de deux secondes.

4.3. Présentation et Validation de l'Application Mobile (*Dashboard NurseView*) :

L'architecture de notre système de télésurveillance ne se limite pas à la détection algorithmique sur l'unité embarquée (Raspberry Pi). Elle intègre un volet applicatif interactif, baptisé *NurseView*, conçu pour assurer la liaison entre l'intelligence artificielle et le personnel soignant. Cette section est dédiée à la validation fonctionnelle de cette interface de suivi, en mettant l'accent sur l'ergonomie, la fiabilité de l'affichage en temps réel et la réactivité du système d'alerte.

4.3.1. Démonstration de l'interface de suivi en temps réel :

L'interface utilisateur principale a été développée à l'aide du framework *Streamlit* et structurée sous forme de tableau de bord *(Dashboard*) centralisé. Ce choix technologique permet un déploiement flexible, accessible via n'importe quel navigateur sur ordinateur, tablette ou smartphone.

- **Affichage dynamique du statut :** L'état de chaque chambre est synchronisé en temps réel grâce à une connexion avec *Firebase Realtime Database*. Dès que le modèle Random Forest met à jour une posture sur le nœud cloud, le tableau de bord intercepte la modification et rafraîchit l'interface de façon instantanée.
- **Indicateurs visuels et colorimétriques :** Afin de faciliter la distinction cognitive, des cartes patients (*Patient Cards*) ont été intégrées avec un code visuel strict. Un badge vert indique une situation stable, tandis qu'une bordure rouge clignotante est immédiatement déclenchée en cas d'anomalie. Les informations essentielles du patient (nom, âge, diagnostic et médecin traitant) restent visibles en permanence.

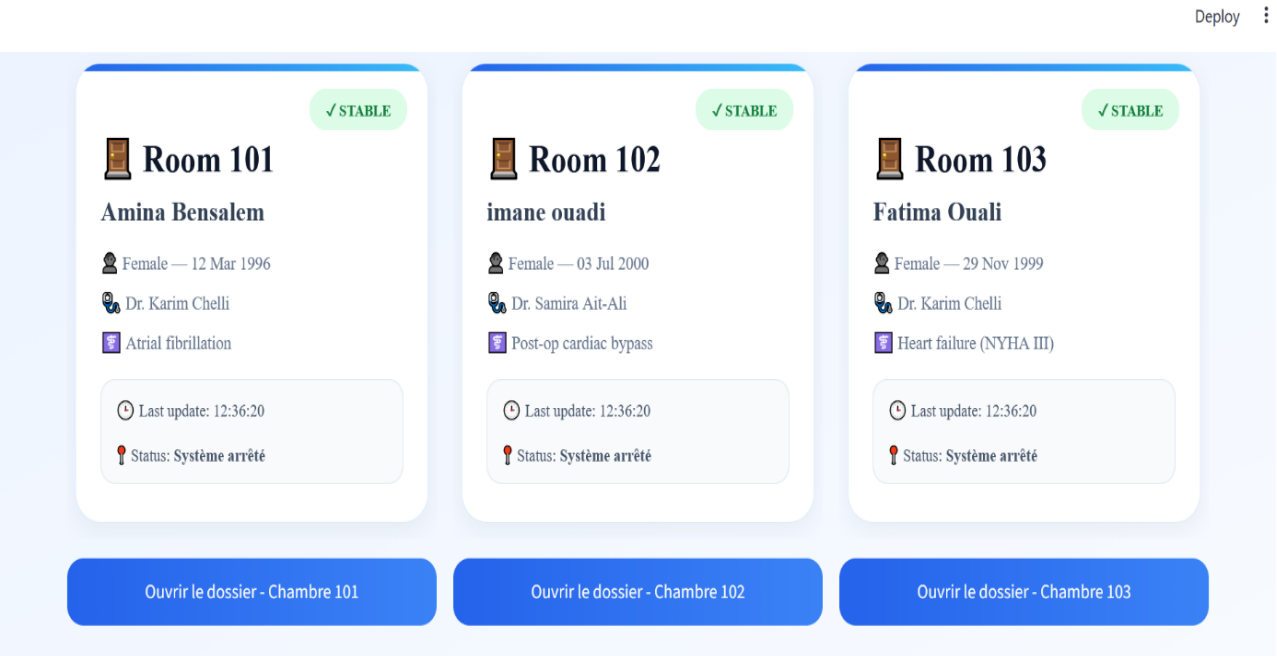


Figure 4.6 : Interface principale du Dashboard NurseView en temps réel.

4.3.2. Visualisation de l'historique et de la traçabilité du patient

La surveillance médicale exige un suivi chronologique rigoureux. Pour répondre à ce besoin, l'application intègre une vue détaillée par chambre, permettant de tracer l'évolution comportementale du patient.

- **Le journal des événements (Logs) :** Chaque changement de posture est horodaté et enregistré dans un historique. L'application génère un tableau de données (*DataFrame*) affichant la date, l'heure exacte, la position détectée, et une note d'état (ex : "Patient sous surveillance" ou "Intervention d'urgence requise").
- **Utilité clinique :** Cette traçabilité permet aux médecins de vérifier si un patient alité respecte les protocoles de retournement (pour éviter les escarres) et d'analyser son niveau d'agitation durant la nuit.

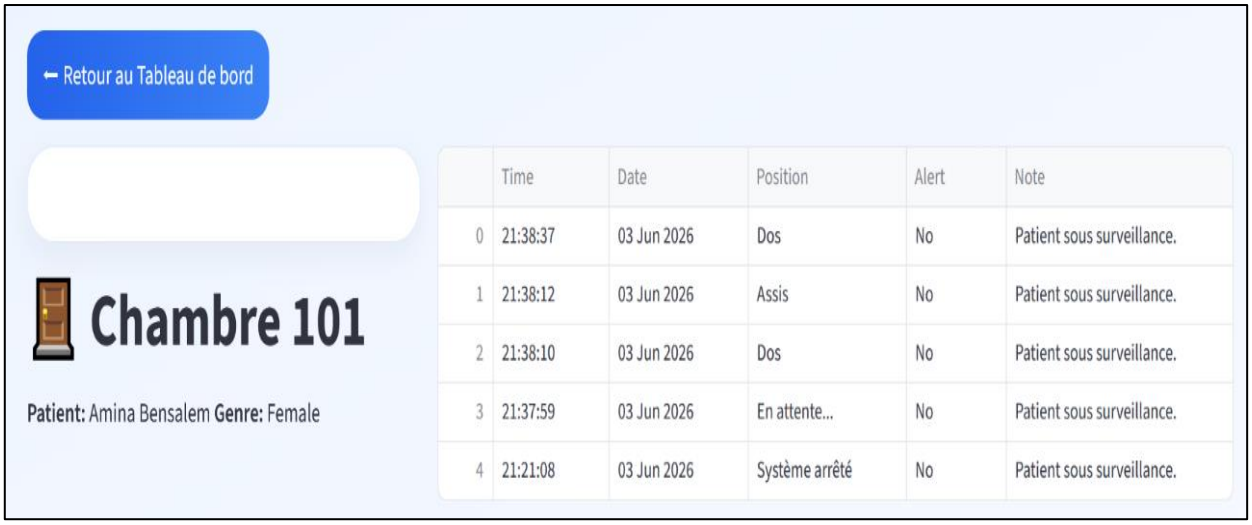


Figure 4.7 : Écran d'historique et de traçabilité des postures.

4.3.3. Test de réactivité des alertes critiques (Sortie/Chute) :

La métrique la plus cruciale pour la sécurité du patient est le temps de latence entre la détection d'un événement critique et la réception de l'alerte par le soignant.

- **Mécanisme d'alerte sonore et visuel :** Pour contourner les limitations habituelles des navigateurs web, un système de forçage audio (*autoplay*) a été implémenté. Lorsqu'une posture "Sortie" ou "Chute" est classifiée, une variable d'alerte globale bascule, déclenchant instantanément une sirène d'urgence couplée à une bannière clignotante sur l'interface.
- **Temps de réponse :** Les tests effectués montrent que le délai de transmission via Firebase jusqu'au déclenchement de l'alarme sonore sur le Dashboard est de l'ordre de **1 à 2 secondes**, garantissant une intervention quasi-immédiate du personnel soignant.

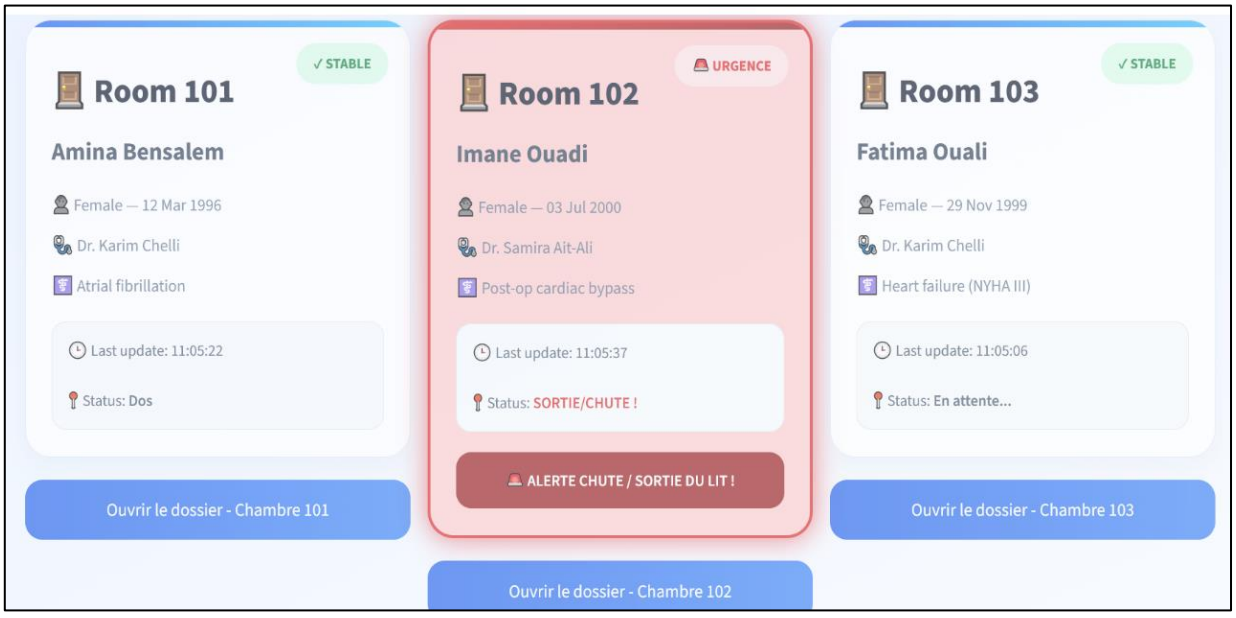


Figure 4.8 : interface quand il y'a une alerte de chute

4.4. Évaluation globale du Système de Bout en Bout (*End-to-End*) :

L'évaluation de notre solution hybride serait incomplète sans une validation de la chaîne de traitement dans son intégralité (End-to-End). Le flux de données traverse plusieurs couches technologiques :

- **Acquisition :** Flux vidéo capturé dans la chambre.
- **Edge Computing (Raspberry Pi) :** Extraction des descripteurs (YOLOv8 + MediaPipe) et classification (Random Forest).
- **Cloud :** Transmission de l'état via l'API Firebase.
- **Visualisation :** Réception et affichage sur l'application NurseView.

Les tests de robustesse globale démontrent que le système parvient à maintenir une fréquence de traitement suffisante (FPS) sur le Raspberry Pi 4, tout en assurant une consommation de bande passante réseau extrêmement faible. En effet, seules les métadonnées de classification (chaînes de caractères : "Assis", "Dos", "Sortie") et l'horodatage sont envoyées vers Firebase, et non le flux vidéo lourd. Ce choix architectural préserve la confidentialité totale du patient (*Privacy-by-Design*), puisque les images ne quittent jamais la chambre d'hôpital, tout en assurant une résilience face aux pannes de réseau à haut débit.

4.4.1. Journalisation et Traçabilité des Événements (Le Journal de Surveillance) :

En plus de l'affichage dynamique sur l'interface *NurseView* et de la mise à jour de la base de données Firebase, notre système intègre un mécanisme de journalisation locale automatique. À chaque changement de posture détecté dans l'une des chambres, le script génère instantanément une trace textuelle dans un fichier de rapport nommé « journal_surveillance.txt ».

Chaque entrée de ce journal est structurée de manière rigoureuse : elle enregistre l'horodatage précis (heure, minute, seconde), le numéro de la chambre concernée, ainsi que le statut ou l'alerte spécifique détectée.

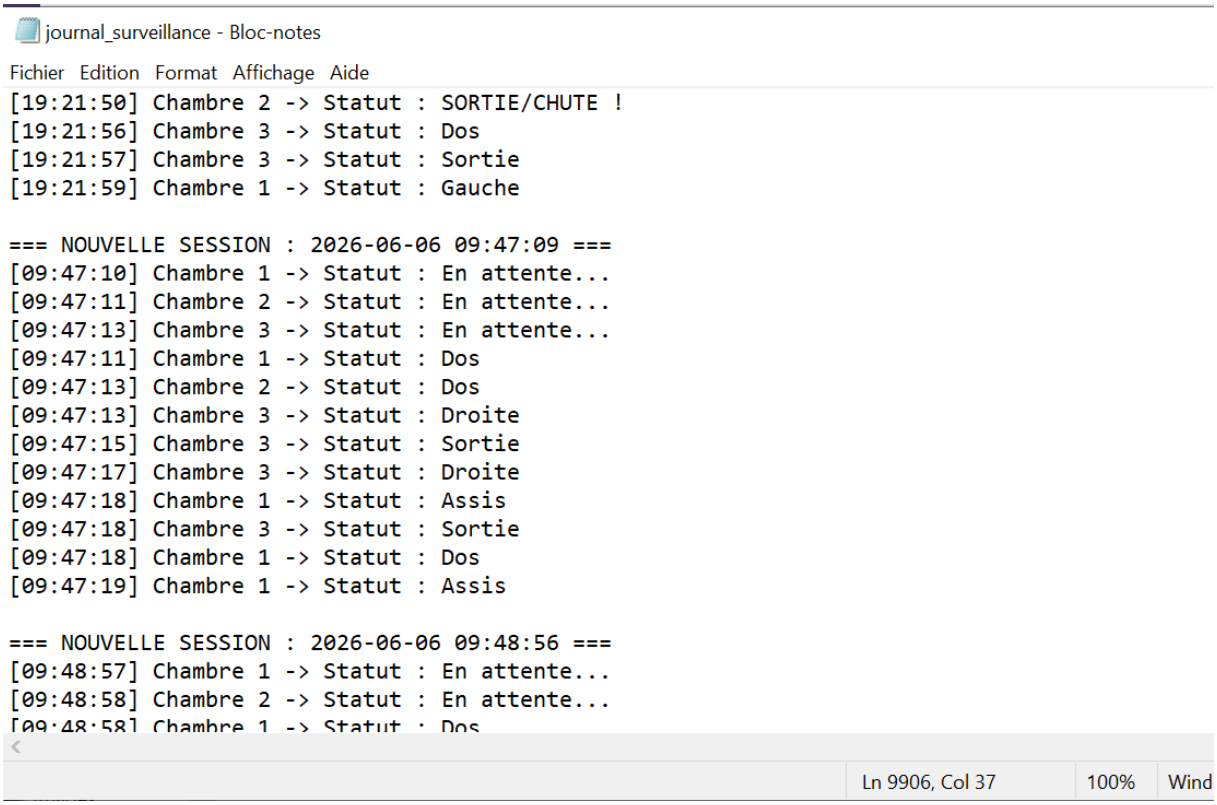


Figure 4.9 : Capture d'écran du fichier de journalisation locale (journal_surveillance.txt).

Intérêt clinique et technique : L'intégration de ce journal de surveillance offre deux avantages majeurs :

- **Audit Médical (Historique) :** Il permet au personnel soignant ou aux médecins de consulter à posteriori le fil des événements de la nuit, offrant ainsi un historique fiable pour analyser le comportement ou les incidents liés à un patient.
- **Résilience et Sécurité :** En cas de coupure temporaire de la connexion Internet (perte de liaison avec Firebase), ce fichier local garantit qu'aucune alerte critique ne soit perdue, assurant ainsi la continuité de la traçabilité des données au sein de l'établissement hospitalier.

4.4.2. Étude de l'Intégration du Système Embarqué (Raspberry Pi) :

Architecture et Contraintes :

A. Architecture de Connexion Théorique (Edge-to-Cloud) :

Dans la conception initiale de notre système, la carte Raspberry Pi 4 est positionnée comme une passerelle locale (*Edge Gateway*) placée dans chaque chambre. Le schéma de connexion théorique s'articule comme suit :

- **Captation locale :** Une caméra (USB ou module CSI) est connectée directement à la carte Raspberry Pi pour filmer le lit du patient.

- **Traitement au niveau de la bordure (Edge Computing) :** La carte embarquée doit exécuter localement les scripts d'extraction (YOLOv8 + MediaPipe) et le classifieur (*Random Forest*) pour analyser la posture sans saturer le réseau.
- **Communication Cloud :** La Raspberry Pi se connecte au réseau Wi-Fi local de l'hôpital et utilise des requêtes sécurisées (HTTPS/REST) pour envoyer uniquement les alertes textuelles (ex : "Chute", "Sortie") vers la base de données Firebase Realtime.

B. Justification Technique de la Non-Implémentation Pratique :

Bien que cette architecture soit idéale sur le plan conceptuel, sa mise en œuvre physique sur une carte Raspberry Pi 4 standard a révélé des limites matérielles critiques lors des phases de pré-test, ce qui nous a conduits à privilégier une simulation sur station de travail (PC) :

- **Puissance de calcul insuffisante :** L'exécution simultanée de YOLOv8 (même en version Nano) et de MediaPipe Pose (qui extrait 33 points clés en 3D) demande des ressources CPU/GPU intenses. Sur une architecture ARM comme celle de la Raspberry Pi, cela provoque une chute drastique du taux de rafraîchissement (généralement inférieur à 2 ou 3 FPS).
- **Exigence du Temps Réel :** Un système de détection de chutes médicales ne peut tolérer aucune latence. Un retard de traitement de quelques secondes dû à la surcharge de la carte embarquée annulerait l'intérêt du système d'alerte instantanée.
- **Gestion multi-chambres :** Notre projet simule une surveillance multi-chambres centralisée. Centraliser le traitement sur un PC performant permet de valider la logique logicielle et la synchronisation avec Firebase pour plusieurs flux vidéo en même temps.

Conclusion de l'étude matérielle :

L'utilisation du PC nous a permis de valider fonctionnellement notre concept (*Proof of Concept*). Pour un déploiement réel sur Raspberry Pi à l'avenir, il sera indispensable d'ajouter un accélérateur matériel de calcul (comme une clé *Intel Movidius Neural Compute Stick* ou un TPU *Google Coral*) ou d'optimiser les modèles via la quantification.

4.4.3. Perspectives d'Évolution et Améliorations Futures :

Bien que notre système *NurseView* ait prouvé sa viabilité fonctionnelle à travers les simulations réalisées, plusieurs axes de développement peuvent être envisagés pour transformer ce prototype en une solution médicale industrielle et totalement autonome :

- **Optimisation Algorithmique pour l'Embarqué (Edge AI) :** Afin de concrétiser le déploiement sur les cartes Raspberry Pi (comme étudié dans la section 4.5.4), la perspective immédiate est l'optimisation des modèles. L'utilisation de techniques comme la quantification (conversion des poids du modèle de Float32 à Int8) ou le pruning (élagage des neurones redondants) permettra de réduire l'empreinte mémoire de YOLOv8 et MediaPipe, augmentant ainsi considérablement le nombre de FPS sur un matériel limité.
- **Extension Fonctionnelle des Postures :** Le modèle actuel peut être enrichi pour détecter d'autres situations cliniques critiques. Il serait pertinent d'entraîner le classifieur *Random Forest* à reconnaître des signes d'agitation motrice (crises de convulsions), l'immobilité prolongée (prévention des escarres) ou encore le glissement progressif du patient vers le bas du lit.
- **Intégration de Capteurs Multi-modaux :** Pour renforcer la précision du système et pallier le problème d'occlusion (lorsque le patient est entièrement sous la couverture), une perspective majeure consiste à coupler la vision par ordinateur avec des capteurs non intrusifs, tels que des tapis de pression connectés sous le matelas ou des capteurs thermiques/infrarouges pour une efficacité optimale en vision nocturne.
- **Système d'Alerte Mobile Avancé :** Enfin, au-delà du Dashboard centralisé de l'infirmière, le système gagnerait à intégrer un module de notification push mobile (via *Firebase Cloud Messaging*) ou des alertes SMS d'urgence. Cela permettrait de notifier instantanément le personnel soignant directement sur leurs smartphones ou montres connectées lorsqu'ils sont en déplacement dans d'autres services

4.5. Travaux connexes :

Plusieurs travaux se sont intéressés à la surveillance automatique des patients alités et semi-alités, allant des capteurs de contact aux approches basées sur la vision par ordinateur et l'estimation de pose. Le tableau ci-dessous résume les principales approches existantes, en précisant pour chacune la référence (article ou projet), leurs limites, et la manière dont notre solution se positionne par rapport à elles.

Travaux connexes (référence)	Approche / Technique	Limite principale	Notre positionnement
Capteurs de contact (tapis de pression, accéléromètres) *Haque et al., 2020 — Nature*	Mesure de poids / mouvement par capteurs physiques	Intrusif, maintenance, fausses alarmes	Solution sans contact (caméra), non intrusive
Classification de posture au lit *Vision-Based In-Sleep Upper-Body & Head Pose Classification — NIH/PMC*	Classification de postures à partir d'une caméra	Cas du sommeil ; non spécifique à la sortie/chute	5 postures incluant « Sortie / Chute »
BED Net *Bed-exit detection via keypoints & action recognition — MDPI Sensors*	Keypoints + reconnaissance d'action sur séquence	Modèle lourd, complexe à embarquer	Pipeline léger (MediaPipe + Random Forest)
YOLOv8-Pose bed-exit monitor *Projet GitHub (Ultralytics)*	Détection de pose + franchissement de ligne	Décision binaire, sensible à la géométrie	Classification fine en 5 postures
Approches multimodales *In-Bed Pose Estimation: A Review — arXiv*	RGB + LWIR + profondeur + pression	Matériel coûteux et complexe	Une seule caméra RGB + score de visibilité
Détection de chute sur Raspberry Pi *Fall detection on Pi AI Camera (HigherHRNet) — Hackster.io*	Estimation de pose + critères géométriques	Modèle simple, portée limitée	Chaîne complète + Dashboard + journal local

Tableau 4.1 : Positionnement de notre approche par rapport aux travaux connexes.

Notre contribution se démarque de ces travaux sur plusieurs aspects. Nous proposons une approche hybride et légère, associant YOLOv8, MediaPipe et Random Forest, conçue pour fonctionner sur une plateforme embarquée. Le problème de l'occlusion est traité de manière explicite à l'aide d'une seule caméra RGB, en exploitant le score de visibilité fourni par MediaPipe. Par ailleurs, nous avons constitué un jeu de données spécifique au lit médicalisé, regroupant environ 10 783 images issues de 9 acteurs, et défini une classification fine en cinq postures intégrant la classe critique « Sortie / Chute ». Enfin, l'ensemble repose sur une chaîne de traitement de bout en bout (Firebase, tableau de bord NurseView et journalisation locale) conçue selon le principe du *Privacy-by-Design*.

Conclusion :

Ce chapitre a permis de valider expérimentalement l'efficacité de notre solution de télésurveillance. L'évaluation quantitative du modèle d'Intelligence Artificielle a révélé une exactitude globale de 87.59 %, prouvant la pertinence de l'approche hybride combinant vision par ordinateur et apprentissage automatique. L'analyse détaillée des matrices de confusion nous a permis de comprendre les limites physiques du système, notamment le phénomène d'occlusion, et d'y remédier scientifiquement via l'exploitation du score de visibilité.

Par ailleurs, la validation de la plateforme de suivi *NurseView* a confirmé que la transition de la prédiction algorithmique brute vers une interface médicale ergonomique et réactive s'effectue de manière fluide et sécurisée. Ce système de bout en bout constitue une réponse technologique viable, économique et respectueuse de la vie privée pour assister le personnel hospitalier dans la prévention des risques liés aux patients semi-alités.

Conclusion générale et perspectives :

Le génie biomédical a pour mission principale de concevoir des systèmes innovants capables d'améliorer la qualité des soins et la sécurité des patients. Dans ce cadre, notre travail a porté sur la conception et la réalisation d'un système intelligent de surveillance des patients semi-alités, basé sur la vision par ordinateur et l'intelligence artificielle, afin d'assurer une surveillance automatique, continue et fiable tout en détectant rapidement les situations à risque.

L'approche adoptée repose sur la combinaison de plusieurs technologies performantes. L'utilisation de YOLOv8 pour la détection, de MediaPipe pour l'extraction des caractéristiques biomécaniques, et de l'algorithme Random Forest pour la classification des postures a permis de transformer un flux vidéo en une information médicale exploitable en temps réel.

Les résultats obtenus confirment la capacité du système à reconnaître efficacement les différentes postures du patient et à détecter les situations critiques, notamment les tentatives de sortie de lit et les risques de chute. L'intégration d'un système d'alertes intelligentes et d'une interface de monitoring renforce la réactivité du personnel soignant et contribue à améliorer la qualité de la prise en charge.

Cependant, ce projet a également mis en évidence plusieurs défis importants. Parmi ceux-ci, nous citons le problème d'occlusion lié aux couvertures, ainsi que l'absence de bases de données adaptées à notre problématique, ce qui nous a conduits à concevoir et développer notre propre dataset à partir de zéro. Cette étape a nécessité un travail rigoureux de collecte, d'annotation et de structuration des données. Nous pouvons également mentionner les contraintes des systèmes embarqués, ainsi que l'optimisation de la base de données, qui reste essentielle pour garantir une gestion rapide, fiable et sécurisée des informations.

Au-delà des résultats techniques, cette expérience nous a permis de développer des compétences solides en intelligence artificielle, en traitement d'image et en intégration des systèmes, tout en mettant en évidence le rôle croissant des technologies intelligentes dans le domaine médical.

En conclusion, ce projet constitue une étape concrète vers le développement de systèmes de surveillance médicaux intelligents, capables d'améliorer la sécurité des patients et de contribuer à l'évolution vers des hôpitaux intelligents et connectés.

En perspective, plusieurs axes d'amélioration peuvent être envisagés pour étendre et renforcer ce travail. L'optimisation des modèles pour leur déploiement sur des systèmes embarqués représente une priorité afin d'assurer un fonctionnement réel en temps réel. L'intégration de capteurs complémentaires, tels que l'infrarouge ou les capteurs de pression, pourrait améliorer la robustesse du système face aux problèmes d'occlusion et aux conditions d'éclairage. Par ailleurs, l'amélioration de la base de données, notamment par

l’utilisation de solutions plus performantes et scalables, permettrait de mieux gérer de grandes quantités de données et d’assurer une meilleure traçabilité des informations médicales. D’autres perspectives incluent le développement d’applications mobiles plus avancées, l’intégration avec des systèmes hospitaliers existants, ainsi que la mise en place d’analyses prédictives basées sur l’intelligence artificielle afin d’anticiper les situations à risque

Bibliographie

- Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine, 25(1), 44–56. Disponible sur : https://www.nature.com/articles/s41591-018-0300-7
- Aceto, G., Persico, V., & Pescapé, A. (2020). Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0. Journal of Industrial Information Integration, 18, 100129. Disponible sur : https://www.sciencedirect.com/science/article/pii/S2452414X20300080
- Haque, A., Milstein, A., & Fei-Fei, L. (2020). Illuminating the dark spaces of healthcare with ambient intelligence. Nature, 585(7824), 193–202. Disponible sur : https://www.nature.com/articles/s41586-020-2669-y
- Dataset GMDCSA-24 (chute, vidéos). Zenodo. Disponible sur : https://zenodo.org/records/11216408
- Dataset FER-2013 (expressions faciales). Kaggle. Disponible sur : https://www.kaggle.com/datasets/msambare/fer2013
- In-Bed Pose Estimation: A Review (modalités, datasets, défis). ArXiv. Disponible sur: https://arxiv.org
- Vision-Based In-Sleep Upper-Body & Head Pose Classification. NIH / PMC. Disponible sur : https://pmc.ncbi.nlm.nih.gov/articles/PMC8199396/
- BED Net : Bed-exit detection via keypoints and action recognition. MDPI Sensors. Disponible sur: https://www.mdpi.com/1424-8220/23/5/2451
- Bed Exit Alarm via Webcam. ACM Digital Library. Disponible sur: https://dl.acm.org
- Raspberry Pi AI Camera / IMX500 (offload inference, model zoo). Raspberry Pi. Disponible sur : https://www.raspberrypi.com/products/ai-camera/
- Fall detection on Pi AI camera (HigherHRNet + géométrie). Hackster.io. Disponible Sur : https://www.hackster.io
- YOLOv8 pose bed-exit monitor (franchissement bord de lit). GitHub. Disponible sur : https://github.com/ultralytics/ultralytics
- Continuous Patient Monitoring with AI (analyse vidéo en milieu hospitalier). ArXiv. Disponible sur : https://arxiv.org
- Noergaard, T. (2013). *Embedded Systems Architecture*. Elsevier.

- Lee, E. A., & Seshia, S. A. (2017). *Introduction to Embedded Systems*. MIT Press.
- Marwedel, P. (2018). *Embedded Systems Design*. Springer.
- Liu, J. W. S. (2000). *Real-Time Systems*. Pearson Education.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th Edition). Pearson Education.
- Dougherty, G. (2009). *Medical Image Processing*. Springer.
- Davies, E. R. (2018). *Computer Vision: Principles, Algorithms, Applications, Learning*. Academic Press.
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer. Disponible sur: Computer Vision Book
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Disponible sur : Deep Learning Book Website
- Raspberry Pi Foundation. (2024). *Raspberry Pi Documentation*. Disponible sur : Raspberry Pi Documentation
- OpenCV. (2024). *Open Source Computer Vision Library*. Disponible sur : OpenCV Official Website
- Ultralytics. (2024). *YOLO Documentation*. Disponible sur : YOLO Documentation
- MediaPipe Team. (2024). *MediaPipe Pose Documentation*. Disponible sur : MediaPipe Documentation
- IEEE Transactions on Pattern Analysis and Machine Intelligence. (2020). *Graph Neural Networks for Scene Understanding*.
- Springer. (2021). *Optical Character Recognition using Deep Learning*.
- Goodfellow, I., et al. (2014). *Generative Adversarial Networks*. NeurIPS. Disponible sur : GAN Paper NeurIPS 2014.
- Cats Power Design. (s.d.). *Système embarqué : qu'est-ce que c'est ?*.https://www.catspowerdesign.fr/actualites/systeme-embarque?utm_source=chatgpt.com
- Liu, J. W. S. (2000). *Real-Time Systems*.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing*.