



Mémoire de Master

Présenté au

Département : Génie Électrique

Domaine : Sciences et Technologies

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Réalisé par :

SAADI ILYES

Thème

Détection de chute humaine en temps réel via Deep Learning

Soutenu le : 03/07/2023

Devant le Jury composé de :

Mr :	SAOUD BILAL	M.C.A	Univ. Bouira	Président
Mr :	CHELBI SALIM	M.C.A	Univ. Bouira	Rapporteur
Mr :	BENSAFIA YASSINE	M.C.A	Univ. Bouira	Examineur

Dédicaces

J'ai l'honneur de dédier ce modeste travail à ceux qui m'ont encouragé et soutenu.

Mon très cher père qui était toujours à mes cotés à tous moment.

Ma très chère mère, qui me donne toujours l'espoir de vivre.

Mes frères pour leur soutien.

Toute ma famille et ainsi mes amis.

Que dieu les garde pour moi.

Saadi Ilyes

Remerciements

Ce travail a été effectué au sein de la faculté des Sciences et des sciences appliquées de l'Université de Bouira.

Louange à **ALLAH** par la grâce duquel les bonnes choses se réalisent, je remercie **DIEU** le tout puissant et miséricordieux, de m'avoir donné la force, la santé, la volonté et le courage d'accomplir ce modeste travail.

Je tiens à remercier toutes les personnes qui m'ont soutenu de près ou de loin, en particulier :

Mon encadreur de mémoire M. CHELBI SALIM pour sa disponibilité, et son précieux suivi tout au long de la réalisation de ce travail.

M. SAADI MOHAMED mon chère frère pour son soutien inestimable, son expertise et ses conseils précieux tout au long de ce projet.

Les membres du jury d'avoir bien voulu accepter de juger ce travail, et pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs proposition.

Tous mes enseignants du département de génie électrique de l'Université de Bouira Akli Mohand Oulhadj.

Et pour finir je présente mes remerciements les plus sincères à ma chère famille, particulièrement à mes parents, pour leur soutien inconditionnel et leur encouragement durant toutes mes années d'études.

À tous, un grand merci...

Résumé

Ce projet met en évidence notre succès dans l'utilisation du Deep Learning pour la détection en temps réel des chutes humaines. En utilisant l'algorithme VGG-16 et en entraînant le modèle avec le jeu de données FPDS, nous avons développé un système hautement performant. Ce système utilise également des techniques avancées telles que la soustraction de l'arrière-plan et la détection de pose humaine avec l'algorithme YOLOv5 et OpenPifPaf pour améliorer la précision de la détection. Nous avons implémenté ce système sur le Raspberry Pi 4, offrant ainsi une solution compacte et portable pour la surveillance des chutes dans divers environnements. Les résultats obtenus ont démontré une précision élevée dans la détection des chutes, ouvrant ainsi la voie à des applications potentielles dans les domaines de la sécurité, de la santé et du bien-être des individus.

Mots clés : Détection de chute, L'apprentissage en profondeur, Raspberry Pi, Traitement d'image, OpenCV, ...

Table des matières

Remerciements	I
Résumé	II
Table des matières	III
Table des figures	VI
Liste des tableaux	VIII
Liste des Acronymes	IX
Introduction générale	1
1 Traitement d'images	3
1 Introduction	3
2 Définition d'image	3
3 Image numérique	4
4 Caractéristiques d'une image numérique	4
4.1 Dimension	4
4.2 Résolution	5
4.3 Le pixel	5
4.4 Histogramme	6
5 Les types des images	6
5.1 image binaire	6
5.2 Image à niveaux de gris	6
5.3 Image en couleurs	7

6	Les principales techniques de traitement des d'images	7
6.1	Acquisition d'une image	8
6.2	Prétraitement d'images	8
6.2.1	Modification d'histogramme	8
6.2.2	Réduction du bruit par filtrage	9
6.3	Le filtrage	10
6.4	Segmentation	10
6.4.1	Segmentation par Classification	11
6.4.2	Segmentation basée contour	13
6.4.3	Segmentation basée région	13
6.4.4	Segmentation par seuillage	15
7	Conclusion	16
2	Deep Learning	17
1	Introduction	17
2	Deep Learning	17
2.1	Pourquoi le Deep Learning est important?	18
2.2	Fonctionnement du Deep Learning	19
2.3	Domaine d'application de Deep Learning	19
3	Réseaux de neurones	20
3.1	Neurone biologique	20
3.2	Neurone artificiel	21
3.3	Correspondance entre neurone biologique et neurone artificiel	21
3.4	Comportement de neurone artificiel	22
3.5	Domaines d'application du réseau de neurones	23
3.6	Avantages et Inconvénients du réseau de neurone	23
3.6.1	Avantage	23
3.6.2	Inconvénients	23
4	Algorithmes de Deep Learning	24
4.1	Réseaux de neurones pré-entraînés non supervisés	24
4.2	Les réseaux de neurones récurrents (RNN)	24
4.3	Les réseaux de neurones convolutifs (CNN)	25
4.3.1	Architecture d'un Réseaux neuronaux convolutifs (CNN)	26
4.3.2	Les modèles réseaux de neurones convolutifs	28

4.3.3	L'application des réseaux neuronaux convolutifs	29
5	Conclusions	30
3	Réalisation	31
1	Introduction	31
2	Les outils de développement	31
2.1	Base de données	31
2.2	Méthode utilisée	32
2.2.1	L'architecture de YOLOv5	32
2.2.2	L'architecture de VGG-16	33
2.3	Langage de programmation	35
2.3.1	Python	35
2.4	Bibliothèques utilisées	35
2.4.1	OpenCV	35
2.4.2	OpenPifPaf	36
2.4.3	PyTorch	36
3	Implémentation	37
3.1	Matériel utilisé	37
3.1.1	Raspberry pi 4 :	37
3.1.2	Caméras Raspberry Pi	39
3.2	Installation et configuration	40
3.2.1	Préparation de Raspberry Pi	40
3.2.2	Configuration de la Caméra Pi	42
3.3	Test et Résultat	44
3.3.1	Apprentissage des algorithmes	44
3.3.2	Mesures d'évaluation	45
3.3.3	Le protocole SMTP	46
3.3.4	Fonctionnement de système	46
3.4	Résultats d'implémentation	49
4	Conclusion	50
	Conclusion générale	51

Table des figures

1.1	Représentation d'une image numérique [4].	4
1.2	Le groupe de pixels forme la lettre A [2].	5
1.3	histogramme associé à des images	6
1.4	Types d'images [4].	7
1.5	Les principales techniques de traitement d'images	7
1.6	Les différentes techniques de prétraitement	8
1.7	Filtrage d'une image par un noyau de convolution [2].	10
1.8	Algorithme de K-Means	12
1.9	Le résultat de l'algorithme de segmentation par classification	12
1.10	Différents types de contours	13
1.11	Le résultat de l'algorithme de segmentation en régions [13].	14
1.12	Exemple de Segmentation par croissance de régions (region-growing)	14
1.13	Le résultat de l'algorithme de segmentation en régions [13].	15
2.1	Les sous-branches de l'intelligence artificiel[20].	18
2.2	Organisation en couches interconnectées des réseaux de neurones [17].	19
2.3	Structure d'un neurone biologique [18].	20
2.4	Structure d'un neurone artificiel [18].	21
2.5	Correspondance entre neurone artificiel et neurone biologique	22
2.6	Architecteur de RNN standare	24
2.7	Architecteur de CNN standare [19].	26
2.8	résultats de max et average pooling	27
2.9	Reconnaissance d'une image avec CNN [25].	29
3.1	Exemples d'images de FPDS.	32
3.2	Structure du réseau de l'algorithme YOLO.	33

3.3	Représentation 3D de l'architecture du VGG-16.	34
3.4	Schéma fonctionnel du matériel	37
3.5	Carte Raspberry pi 4 model B	37
3.6	Camera Raspberry Pi	39
3.7	Raspberry Pi Imager	40
3.8	Le client « PuTTY »	41
3.9	Connexion Raspberry par VNC Viewer	42
3.10	le connecteur CSI de la carte Raspberry Pi.	42
3.11	Activation de la caméra Pi	43
3.12	Image du FDPS annotée avec des bounding boxes (encadrés)	44
3.13	Le résultat de l'algorithme de segmentation par classification.	46
3.14	VGG-16 et YOLOv5 basé sur le Deep Learning	47
3.15	L'organigramme de système détection de chute	48
3.16	Démarrage de système	49
3.17	Test du système de détection de chute.	49

Liste des tableaux

2.1	Transition entre le neurone biologique et le neurone artificiel	21
-----	---	----

Liste des Acronymes

CNN	Les réseaux de neurones convolutifs
DL	Deep Learning
DPI	Dots par inch
FPDS	Fall Detection Performance Standard
FP	Faux positif
FN	Faux négatif
IA	Intelligence Artificielle
OMS	Organisation mondiale de la santé
PPP	Points par pouce
RNN	Les réseaux de neurones récurrents
RPi	Raspberry Pi
SMTP	Simple Mail Transfer Protocol
TP	Vrai positif
TN	Vrai négatif
VGG	Visual Geometry Group
YOLO	You Only Look Once

Introduction générale

La chute est un problème de santé majeur, surtout chez les personnes âgées pour qui c'est une occurrence courante dans leur vie quotidienne. Parfois, cette chute devient plus grave et, si la victime n'est pas aidée à temps, elles peuvent entraîner la mort. Selon une enquête menée par l'Organisation mondiale de la santé (OMS) estime que 28 % à 35 % des personnes âgées de plus de 65 ans font des chutes au moins une fois par an [1], un chiffre qui s'élève jusqu'à 44% pour les personnes de plus de 70 ans. Les chutes sont également un facteur important de l'hospitalisation des personnes âgées, touchant plus de 50% d'entre elles selon les enquêtes de l'OMS, et sont à l'origine de près de 40% de tous les décès par blessure [1].

Les incidents de chute sont particulièrement compliqués pour les personnes âgées qui vivent seules, car il leur faut beaucoup de temps avant d'obtenir de l'aide après la chute. Selon l'Organisation mondiale de la santé (OMS), la population des personnes âgées augmentera de manière significative au cours des 20 prochaines années.

Il est possible de détecter les chutes humaines en utilisant principalement trois méthodes : les dispositifs portables, les dispositifs non portables (ambiants) basés sur des capteurs et les dispositifs basés sur la vision. Les dispositifs portables et non portables sont composés de capteurs électroniques minuscules. Quant aux dispositifs basés sur des capteurs ambiants, ils sont constitués de capteurs qui ne nécessitent pas d'être physiquement portés. Ces dispositifs portables et non portables tels que les capteurs de pression et les capteurs de vibrations du sol et les ceintures se caractérisent par leur facilité d'utilisation et leur coût abordable. Cependant, ces dispositifs peuvent parfois manquer de fiabilité et générer des fausses alarmes, ce qui peut limiter leur efficacité. Les personnes âgées en particulier celles qui souffrent de démence, oublient souvent de porter et de recharger ces dispositifs. En raison de ces fausses alertes, du manque de fiabilité, de l'état des piles et des problèmes de port, les utilisateurs et les centres de soins de santé ne recommandent pas toujours l'utilisation des détecteurs de chute portables et non portables.

Le système de détection des chutes humaines que nous avons proposées dans cette thèse est un dispositif à faible coût basé sur la vision, il est composé d'un ordinateur embarqué (Raspberry Pi) et d'une caméra. Ce système peut être installé sur les murs ou les plafonds, il commencera la détection sans aucune interférence humaine. Dans ce système, les personnes âgées ne sont pas obligées de porter un dispositif de détection. Ce système est particulièrement conçu pour personnes âgées qui sont seules, car si d'autres personnes se trouvent à proximité et qu'une chute se produit, elles s'aideront mutuellement.

Le mémoire est organisé en 3 chapitres qui nous permettront de présenter les différents aspects de notre travail.

— **Chapitre 01** : Traitement d'image

Dans ce chapitre, nous avons abordé les principes fondamentaux et les concepts clés nécessaires à la compréhension des techniques de traitement d'images, ainsi que certaines méthodes de segmentation.

— **Chapitre 02** : Deep Learning

Dans ce chapitre, nous parlerons de certains approches De Deep Learning.

— **Chapitre 03** : Réalisation

Dans ce chapitre, nous aborderons les différentes bibliothèques et matériels choisis pour mettre en œuvre notre système.

Enfin, en terminons ce travail par une conclusion générale.

Traitement d'images

1 Introduction

Le traitement d'images consiste en un ensemble de méthodes et de techniques qui visent à rendre l'analyse et la manipulation des images plus faciles, plus efficaces et plus agréables. Son objectif est d'améliorer l'apparence visuelle des images, d'extraire des informations pertinentes et de faciliter leur exploitation.

Dans ce chapitre, nous abordons des notions de base sur l'image et quelques techniques pour son traitement ensuite, nous présentons les différentes techniques connues dans ce domaine.

2 Définition d'image

Une image est une représentation visuelle d'un objet ou d'une personne, obtenue à travers divers moyens tels que la photographie, la vidéo, etc. Elle constitue également une compilation ordonnée d'informations visuelles, qui une fois affichées sur un écran, possèdent une signification perceptible par l'œil humain.

Une image peut être représentée par une fonction $I(x, y)$ de brillance analogique continue, Les coordonnées spatiales (x et y) représentent un point de l'image et I est une fonction d'intensité lumineuse et de couleurs [5]. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation [4].

3 Image numérique

Une image numérique est une représentation d'une image où sa surface est divisée en cellules ou pixels de taille fixe. Chaque pixel possède une caractéristique qui correspond à un niveau de gris ou de couleur [4]. La disposition des pixels est généralement organisée en une structure linéaire, avec des pixels disposés en lignes et en colonnes [6]. La numérisation d'une image implique la conversion de son format analogique en une représentation numérique. Cette représentation numérique est généralement exprimée sous la forme d'une matrice bidimensionnelle de valeurs numériques $F(x, y)$, comme illustré dans la figure 1.1, où :

- Les coordonnées (x et y) représentent les positions des points de l'image.
- $f(x, y)$ correspond au niveau d'intensité de chaque point.

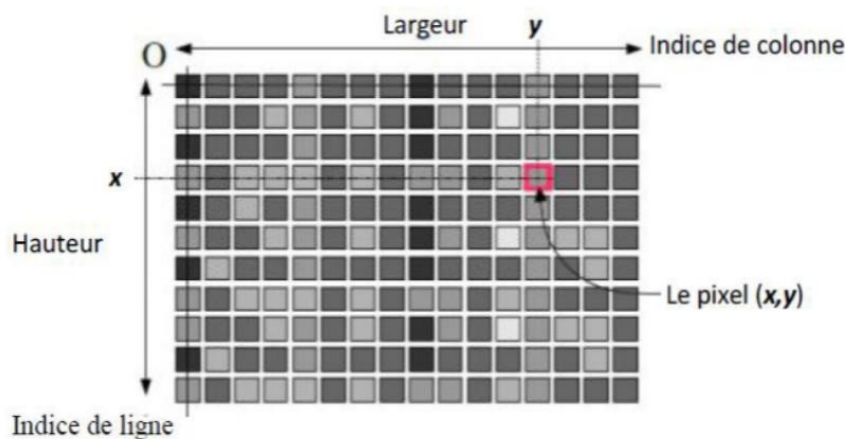


FIGURE 1.1 – Représentation d'une image numérique [4].

4 Caractéristiques d'une image numérique

L'image est composée d'un ensemble d'informations organisées, qui se distinguent par les paramètres suivants :

4.1 Dimension

La dimension d'une image correspond à sa taille, et elle est représentée sous la forme d'une matrice, où chaque élément de la matrice est une valeur numérique représentant l'intensité lumineuse (pixels). Le produit du nombre de lignes et du nombre de colonnes de la matrice représente le nombre total de pixels dans une image [4].

4.2 Résolution

La résolution d'une image est déterminée par le nombre de pixels qu'elle contient par pouce (1 pouce = 2.54 centimètres). Elle est généralement exprimée en "PPP" (points par pouce) ou en DPI (dots per inch) par fois en point par cm. Lorsque le nombre de pixels par unité de longueur d'une image à numériser est élevé, cela entraîne une plus grande quantité d'informations qui décrivent l'image, ce qui se traduit par une meilleure résolution (augmentation du poids de l'image) [4].

Par exemple, considérons une image avec une résolution de 1200 dpi. Cela signifie qu'elle comporte 1200 pixels dans sa largeur et 1200 pixels dans sa hauteur, ce qui donne un total de 1 440 000 pixels (1200 x 1200 ppp).

4.3 Le pixel

Une image numérique est composée d'une multitude de points appelés (pixels) qui s'assemblent pour former l'image [5]. Le pixel représente la petite unité de base de l'image numérique et chaque pixel contient une information de couleur. Le terme "pixel" est dérivé de l'abréviation de l'expression britannique (Picture Element). Tous les pixels sont organisés dans une matrice à deux dimensions, ce qui constitue la structure de l'image [6].

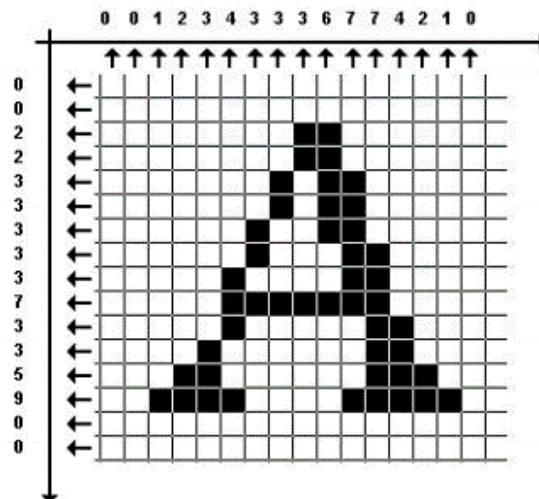


FIGURE 1.2 – Le groupe de pixels forme la lettre A [2].

4.4 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui permet de représenter visuellement les zones lumineuses dans une image. Afin de réduire l'erreur de quantification, de comparer des images prises sous différents éclairages ou encore de mesurer certaines caractéristiques d'une image, il est courant de modifier l'histogramme correspondant [4].

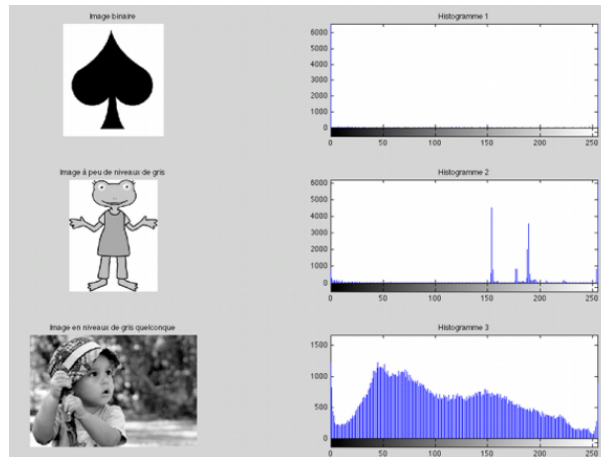


FIGURE 1.3 – histogramme associé à des images

5 Les types des images

5.1 image binaire

Une image binaire (image noir et blanc) est une image de dimension $M \times N$ dans laquelle chaque pixel ne peut prendre que la valeur 0 ou 1. Les pixels sont soit noirs (0), soit blancs (1). Le niveau de gris est codé sur un seul bit appelé Binary digIT [2].

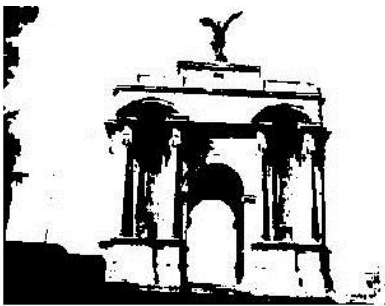
5.2 Image à niveaux de gris

Une image en niveaux de gris (Monochromes) autorise un dégradé de gris entre le noir et le blanc. En général, le niveau de gris est codé sur un octet (8 bits), ce qui permet d'avoir 256 nuances de dégradé [3]. L'expression de la valeur du niveau de gris pour un pixel donné, noté $p(i,j)$, est donc dans l'intervalle $[0, 255]$ [2].

5.3 Image en couleurs

Elle est obtenue par la combinaison de trois couleurs dites primaires (Polychromes) : rouge, vert et bleu également appelées (RVB). Chacune de ces couleurs est représentée par une image en niveaux de gris, avec des valeurs de luminosité allant de 0 à 255. Pour $R=V=B=0$ nous obtenons un noir pur, et pour $R=V=B=255$ nous obtenons un blanc pur [3]. La représentation des images couleurs se fait soit une seule image avec des valeurs de pixels résultant de combinaisons linéaires des composantes de couleur, soit trois images distinctes représentant chacune une composante de couleur, on distingue généralement deux types d'images [10] :

- Images 24 bits.
- Images à palettes.



(a) Image binaire



(b) Image Monochrome



(c) Image polychrome

FIGURE 1.4 – Types d'images [4].

6 Les principales techniques de traitement des d'images

Le traitement d'une image vise à ajouter du sens à celle-ci de manière automatisée, on va citer les différentes techniques de traitement d'image les plus connus tels que : Acquisition, prétraitement d'une image, le filtrage et la segmentation d'image.

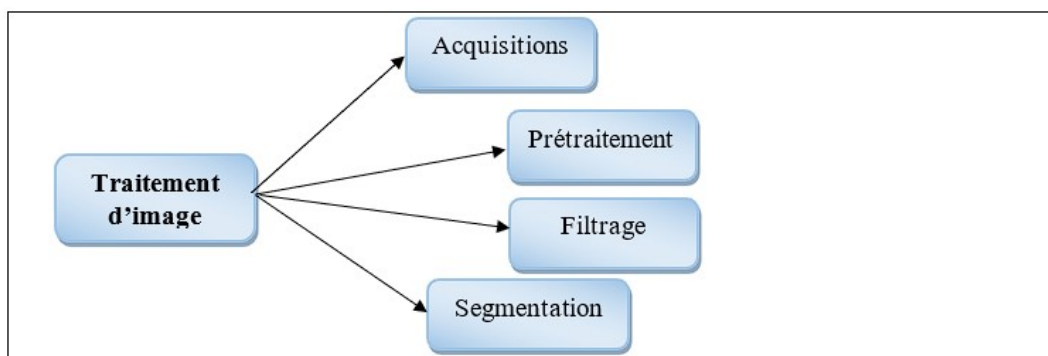


FIGURE 1.5 – Les principales techniques de traitement d'images

6.1 Acquisition d'une image

L'acquisition d'images joue un rôle fondamental dans tout processus de conception et de production d'images. Afin de pouvoir traiter une image sur un système informatique, il est essentiel de la transformer pour la rendre lisible et manipulable par ce système. La transformation de l'objet externe (l'image d'origine) en sa représentation interne se réalise à travers un processus appelé numérisation [11].

6.2 Prétraitement d'images

Cette étape, qui intervient immédiatement après l'acquisition des images, vise à améliorer leur qualité.

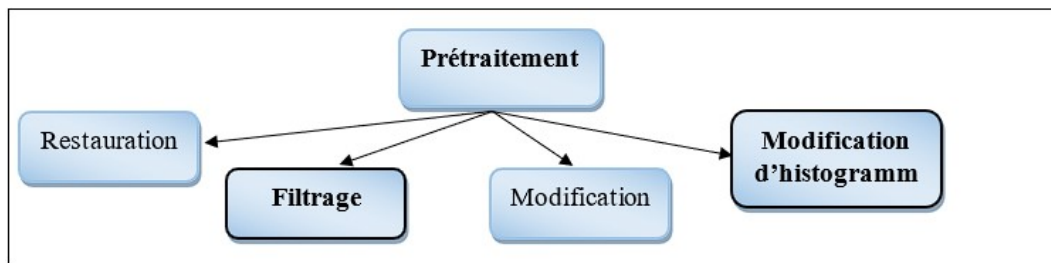


FIGURE 1.6 – Les différentes techniques de prétraitement

Parmi les techniques de prétraitement d'images les plus importantes, il y a :

- La modification d'histogramme.
- La réduction du bruit par filtrage.

6.2.1 Modification d'histogramme

On va présenter deux types de modification d'histogramme : [7]

- Étirement d'histogrammes.
- Égalisation d'histogramme.

- **Étirement d'histogrammes :**

Cette transformation consiste à ajuster la luminosité de l'image en utilisant la règle de trois.

La valeur de chaque pixel est remplacée par le résultat de la formule suivante :

$$I'(x, y) = 255 \times \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \quad (1.1)$$

Les variables $I(x, y)$ $I'(x, y)$ désignent respectivement les intensités des pixels aux coordonnées (x, y) dans l'image mal exposée et dans la nouvelle image.

I_{min} et I_{max} les niveaux de gris min et max de l'image.

- **Égalisation d'histogramme :**

Cette transformation permet d'améliorer le constat, que consiste à appliquer une transformation sur chaque pixel de l'image à partir des opérations sur chaque pixel [7] :

- Étape 1 : calcule l'histogramme $h(i) i \in [0, 255]$.
- Étape 2 Normalisation de l'histogramme $h(i)$.
- Étape 3 densité de probabilité cumulative.
- Étape 4 Transformation des niveaux de gris de l'image :

$$I'(x, y) = C(I(x, y)) * 255 \quad (1.2)$$

6.2.2 Réduction du bruit par filtrage

Généralement, le bruit dans les images est souvent caractérisé par une densité de probabilité f et une fonction de répartition F . La nature du bruit, qu'il soit plus ou moins impulsif, peut être décrite à l'aide de probabilités sous une forme spécifique. [7] :

$$f(a) = C.exp(-K|a|^\alpha) \quad (1.3)$$

Plusieurs méthodes de filtrage ont été développées en fonction du type de bruit, de son intensité et des applications spécifiques auxquelles l'image est destinée.

- Filtres linéaires stationnaires.
- Filtres non linéaires stationnaires.

Les premières méthodes de filtrage, les plus simples, reposent sur des techniques de filtrage linéaire stationnaire (invariant par translations). Cependant, ces techniques présentent des limitations, notamment en ce qui concerne la conservation des transitions. Cela a conduit au développement des filtres "non-linéaires" [3].

6.3 Le filtrage

Le filtrage est une opération qui vise à réduire le bruit en utilisant des algorithmes mathématiques tels que l'interpolation ou la morphologie mathématique [15]. Son objectif est de modifier le contenu d'un pixel en utilisant les informations locales provenant du voisinage du pixel. En général, le filtrage est effectué en convoluant l'image avec un noyau spécifique. Ce noyau peut être considéré comme une petite image qui contient un motif de transformation (linéaire ou non linéaire) qui est appliqué à chaque pixel de l'image à filtrer pour créer une nouvelle image filtrée [2].

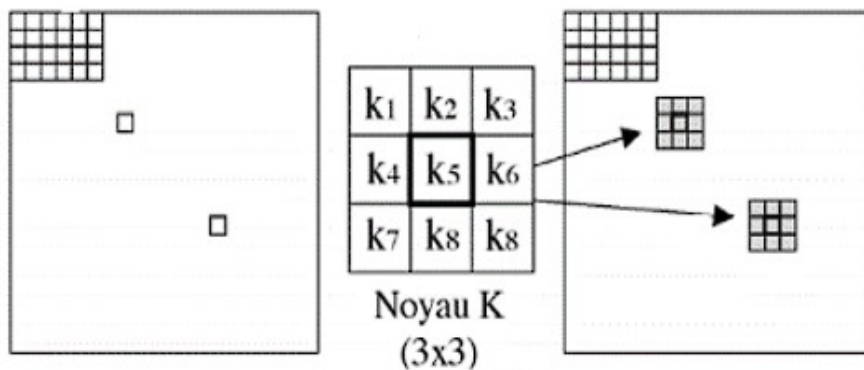


FIGURE 1.7 – Filtrage d'une image par un noyau de convolution [2].

6.4 Segmentation

La segmentation d'images est l'opération la plus importante dans un système de traitement des images, car elle est située à l'articulation entre le traitement et l'analyse des images [8].

Le processus de segmentation d'une image consiste à la découper en différentes régions dans un but de simplifier la représentation de l'image pour la rendre plus facile à analyser et à interpréter.

La segmentation d'images est typiquement utilisée pour localiser et reconnaître les contours des objets dans une image. Techniquement, la segmentation d'images est un processus d'étiquetage des pixels pour créer soit un ensemble de pixels couvrant la totalité de l'image, soit des contours qui délimitent des régions représentant une certaine homogénéité de couleur, intensité ou de texture.

Les méthodes de segmentation peuvent être divisées en deux approches. La première approche repose sur la recherche de discontinuité locale (détection des contours). La deuxième vise à détecter des zones de l'image présentant des caractéristiques d'homogénéité (extraction de région). Ces deux approches sont complémentaires car une région délimite une ligne sur son contour, et une ligne fermée délimite une région à l'intérieur [9].

Les techniques de segmentation : Il existe plusieurs approches de segmentation. Voici quelques-unes des approches couramment utilisées :

- Segmentation par classification.
- Segmentation basée contour.
- Segmentation basée région.
- le seuillage.

6.4.1 Segmentation par Classification

La segmentation par classification consiste à partitionner une image en un ensemble de classes disjointes, pour chaque pixel de l'image on attribue une étiquette parmi l'ensemble des étiquettes qui correspondent chacune à une classe. On dit qu'une classification est supervisée si le nombre de classes est connu a priori sinon il s'agit d'une classification non supervisée [8].

Parmi les méthodes les plus utilisées en segmentation d'images par classification l'algorithme de K-means (algorithme des centres mobiles) qui utilise la distance quadratique moyenne comme critère d'évaluation d'une partition [12].

Algorithme de K-means :

1. Choisir un nombre de classe K .
2. Définir une classification C aléatoire (définir les K centroides Y_i de façon aléatoire dans l'espace des niveaux de gris $[0, L - 1]$).
3. **Tant que** l'inertie intra classe n'est pas stable **faire** :
Affecter chaque niveau de gris dont le centre est le plus proche ;
Calculer les centres de gravité des classes de la nouvelle classification C' des classes ;
 $C \leftarrow C'$;
Fin tant que
4. Afficher la classification obtenue

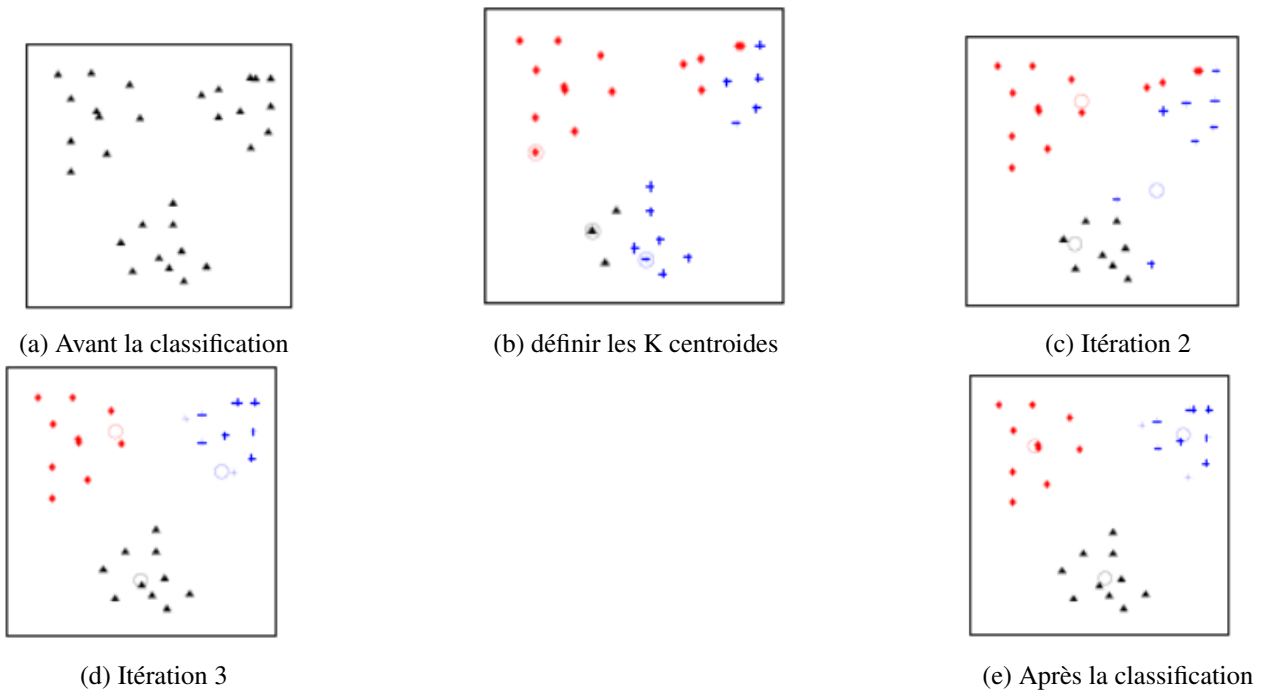


FIGURE 1.8 – Algorithme de K-Means

Sur l'exemple ci-dessous, on peut observer une image avant et après la Segmentation par classification :



(a) Image original



(b) Image après segmentation

FIGURE 1.9 – Le résultat de l'algorithme de segmentation par classification

6.4.2 Segmentation basée contour

L'approche contour vise à détecter les variations entre les régions d'une image. En général, un contour est défini comme un point situé à la frontière de deux objets ou plus ayant des niveaux de gris différents [8]. Les changements d'intensité lumineuse et de couleur sont facilement perceptibles par le système visuel humain. La figure suivante illustre différents modèles de contours [10].

De nombreuses techniques d'extraction de contours existent. Elles sont classées comme suit :

- Les algorithmes basés sur le gradient (opérateurs du premier ordre).
- Les algorithmes basés sur le Laplacien (opérateurs du second ordre).

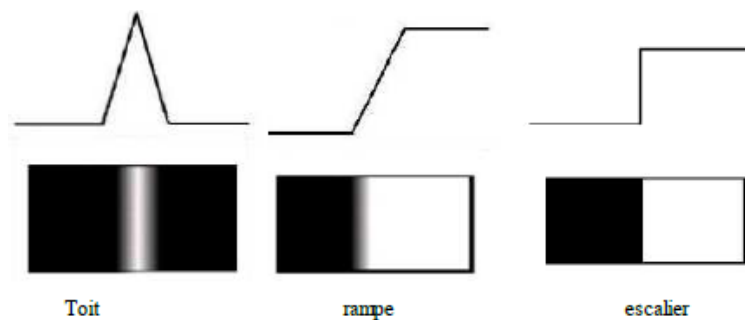


FIGURE 1.10 – Différents types de contours

- Escalier : le contour est net et précis, représentant un contour idéal.
- Rampe : le contour peut être plus flou.
- Toit : le contour peut être simplement représenté par une ligne sur un fond uniforme.

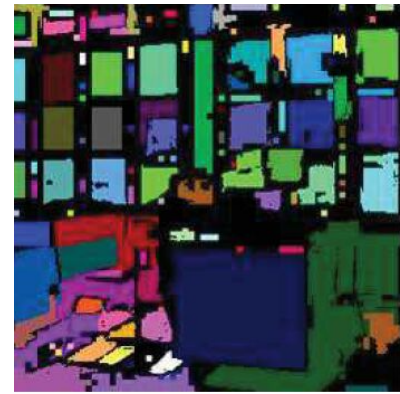
6.4.3 Segmentation basée région

La segmentation en régions est une étape fondamentale dans le domaine du traitement d'images et de la reconnaissance de formes [3]. Elle est généralement considérée comme la première étape pour comprendre et interpréter une image dans de nombreuses applications. L'objectif de la segmentation en régions est de regrouper les pixels d'une image en différentes classes en fonction de certaines propriétés spécifiques, ce qui permet de délimiter et d'identifier les différentes régions présentes dans l'image [8].

Dans cette approche, un processus de regroupement est itérativement appliqué jusqu'à ce que tous les pixels de l'image soient inclus dans des régions spécifiques. L'objectif est de segmenter l'image en se basant sur les propriétés intrinsèques des régions, ce qui signifie que des pixels similaires en termes de caractéristiques telles que la couleur, la texture ou l'intensité sont regroupés ensemble pour former une région distincte [10].



(a) Image originale



(b) Image après segmentation

FIGURE 1.11 – Le résultat de l'algorithme de segmentation en régions [13].

Il existe plusieurs techniques qui méritent d'être mentionnées :

Segmentation par croissance de régions (region-growing) : C'est une méthode locale récursive utilisée pour segmenter une image en régions cohérentes. qui a pour principe de faire croître une région avant de passer à la suivante, Ces régions croient au fur et à mesure par l'incorporation des pixels les plus similaires jusqu'à ce que l'image soit couverte [3].

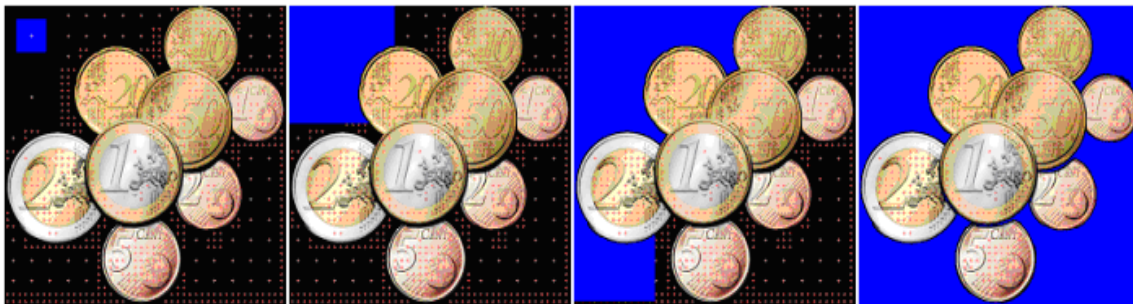


FIGURE 1.12 – Exemple de Segmentation par croissance de régions (region-growing)

Segmentation par fusion de régions (Bottom-up) : Les techniques de réunion sont des méthodes ascendantes dans lesquelles tous les pixels sont parcourus. Pour chaque voisinage de pixels, un prédicat P est testé. Si le prédicat est vérifié, les pixels correspondants sont regroupés dans une région [10].

Segmentation par division de régions (Top-down) : La division d'une image consiste à la partitionner en régions homogènes en fonction d'un critère spécifié. Le principe de cette technique est de prendre l'image elle-même comme région initiale, puis de la diviser progressivement en régions distinctes [10].

Segmentation par division et fusion (Split and Merge) : Ces méthodes combinent les deux approches précédentes, à savoir la division de l'image en petites régions homogènes, puis la fusion des régions voisines et similaires selon un critère de regroupement. L'idée de base est de considérer

chaque pixel comme une région distincte au départ [3]. Ensuite, les régions sont fusionnées si elles satisfont à la fois un critère de similarité des niveaux de gris et un critère d'adjacence entre les régions. Ce processus de fusion continue tant que le critère de fusion est respecté [10].

La Division : consiste à subdiviser l'image jusqu'à ce que la propriété soit vraie dans chaque sous-image.

La Fusion : regroupe les régions adjacentes dont l'union satisfait à la propriété de division et de fusion [8].

6.4.4 Segmentation par seuillage

Le seuillage est une technique simple qui consiste à effectuer une mesure quantitative sur une grandeur donnée. Cette technique permet de diviser les pixels en deux catégories distinctes en fonction de leur mesure : ceux dont la valeur est inférieure au seuil et ceux dont la valeur est supérieure ou égale au seuil [9].

L'opération consiste à assigner la valeur zéro à tous les pixels dont le niveau de gris est inférieur à une valeur spécifique (appelée seuil), et à attribuer la valeur max aux pixels ayant une valeur supérieure. On peut distinguer le seuillage de base, qui est une méthode simple (2 classes), où le résultat du seuillage est une image binaire (Figure 1.13). En revanche, le multi-seuillage est utilisé lorsque l'image contient plusieurs objets avec des niveaux de luminosité différents [4].



(a) Image original



(b) Image après segmentation

FIGURE 1.13 – Le résultat de l'algorithme de segmentation en régions [13].

7 Conclusion

Dans ce chapitre, nous avons abordé les fondamentaux de diverses techniques de traitement d'images. Nous avons présenté quelques méthodes qui sont couramment utilisées dans le domaine du traitement d'images.

Nous avons également discuté des prétraitements d'images qui visent à améliorer la qualité de l'image. Nous avons présenté un état de l'art des méthodes directes de segmentation d'images les plus connues, ainsi que les mesures d'évaluation des résultats, notamment celles de la segmentation par région. L'ensemble des techniques décrites dans les sections précédentes démontre que le problème de la segmentation est complexe.

Dans le prochain chapitre, nous allons présenter le Deep Learning plus en détail, et ses différentes méthodes d'apprentissage.

Deep Learning

1 Introduction

Dans le contexte de la détection en temps réel des chutes humaines, il est crucial d'introduire un élément essentiel : le Deep Learning. Cet outil joue un rôle central dans le développement d'algorithmes capables de détecter les chutes humaines en temps réel à partir de différentes sources d'imagerie ou de capteurs.

Dans ce chapitre, nous parlerons d'abord de Deep Learning avec quelques définitions, ensuite nous présenterons les réseaux de neurones, la correspondance entre neurone biologique et neurone artificiel, et nous détaillerons à la fin les différents types de Deep Learning.

2 Deep Learning

Le Deep Learning ou « apprentissage profond » en français, est un sous-domaine de l'intelligence artificielle (IA) dérivé du Machine Learning (apprentissage automatique) (Voir figure 2.1).

Les systèmes basés sur l'apprentissage profond utilisent un grand nombre de paramètres et de couches, capables d'interagir entre elles pour analyser des données non structurées et prédire des résultats. Ces systèmes sont capables d'apprendre et de s'améliorer de manière autonome. En cas d'erreur détectée par une couche, les données sont alors systématiquement éliminées et renvoyées vers les couches précédentes afin de corriger le modèle mathématique. Par conséquent, lorsque ce modèle sera exposé à de nouvelles données, il sera capable de les assimiler et de les distinguer sans nécessiter d'indication externe [16].

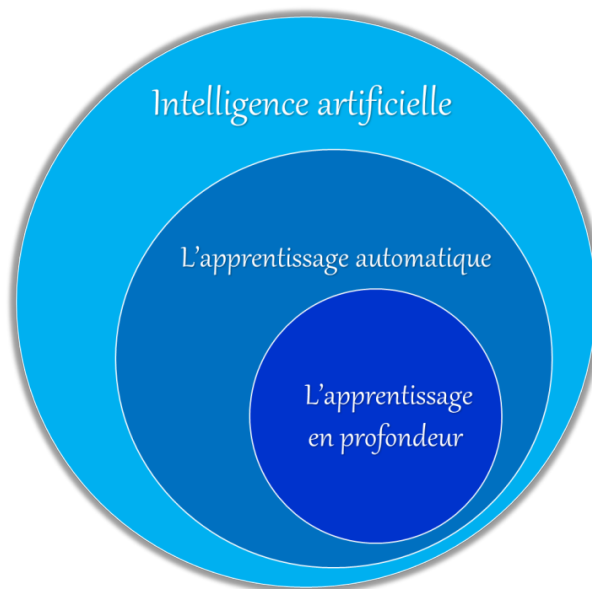


FIGURE 2.1 – Les sous-branches de l'intelligence artificiel[20].

2.1 Pourquoi le Deep Learning est important ?

Le Deep Learning est d'une grande importance en raison de sa capacité à atteindre des niveaux de précision exceptionnels. Les avancées dans la reconnaissance grâce au Deep Learning ont permis aux produits électroniques grand public de répondre aux attentes des clients, ce qui est particulièrement crucial dans les applications prioritaires en matière de sécurité, comme les véhicules autonomes. Le Deep Learning a récemment connu des progrès considérables, le propulsant au-delà des capacités humaines dans l'accomplissement de certaines tâches, comme la classification d'objets dans les images. Ces avancées ont permis d'obtenir des performances exceptionnelles et ont ouvert de nouvelles perspectives dans de nombreux domaines d'application [17].

Le Deep Learning a connu des premières théories dès les années 1980, mais son exploitation pratique n'a émergé que récemment. Voici les raisons de ce développement tardif :

1. Le Deep Learning nécessite un vaste jeu de données annotées. Par exemple, pour le développement des voitures autonomes il est nécessaire d'utiliser des millions d'images et des vidéos pour entraîner les modèles.
2. Le DL demande une capacité de calcul importante. Les GPU (unités de traitement graphique) hautes performances possèdent une architecture parallèle bien adaptée au DL. Lorsqu'ils sont utilisés avec des clusters ou des services de cloud computing, ils permettent aux équipes de réduire considérablement la durée d'entraînement des réseaux de DL [17].

2.2 Fonctionnement du Deep Learning

Les modèles de DL sont souvent appelés "réseaux de neurones profonds" en raison de leur structure basée sur des réseaux de neurones, qui sont des éléments clés de leur fonctionnement.

Le terme "profond" est couramment utilisé pour décrire le nombre de couches cachées présentes dans un réseau de neurones. Les réseaux de neurones classiques sont composés de 2-3 couches cachées (Figure 2.2), avec un ensemble de nœuds interconnectés. En revanche, les réseaux profonds peuvent comporter bien plus que 100 couches cachées.

L'entraînement des modèles de Deep Learning se fait en utilisant de vastes ensembles de données labellisées et des architectures de réseaux de neurones qui sont capables d'apprendre des caractéristiques directement à partir des données, sans nécessiter une extraction manuelle préalable [17].

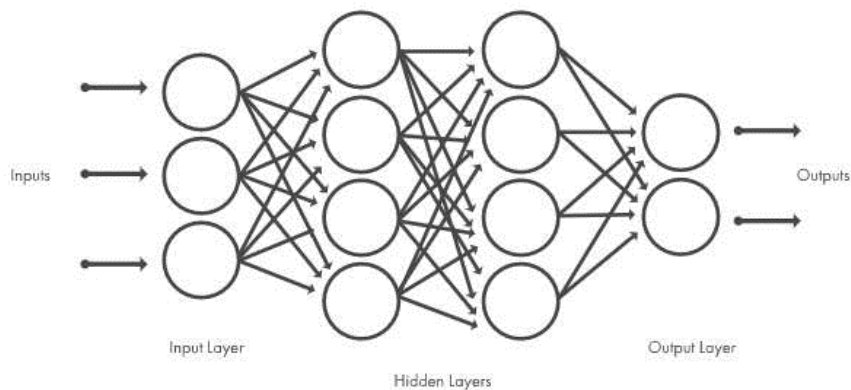


FIGURE 2.2 – Organisation en couches interconnectées des réseaux de neurones [17].

2.3 Domaine d'application de Deep Learning

Le Deep Learning est appliqué dans divers domaines (20), tel que :

- L'intelligence artificielle en général.
- Chatbots (agents conversationnels).
- La reconnaissance visuelle et la comparaison de forme.
- La robotique.
- La santé et la bio-informatique.
- La sécurité.
- Électronique.

3 Réseaux de neurones

Un réseau de neurones est un modèle mathématique de traitement distribué qui se compose de multiples éléments de calcul non linéaire (neurones), qui fonctionnent en parallèle et sont interconnectés par des poids.

Cette corrélation entre les cellules nerveuses confère aux neurones la capacité de stocker et de transmettre des informations, telles que des images, du son et des signaux qui reçoivent à travers les différents neurones. Les réseaux de neurones permettent également l'apprentissage par la répétition et l'erreur.

Les travaux réalisés par Hopfield en 1982 ont démontré que les réseaux de neurones artificiels étaient capables de résoudre des problèmes d'optimisation. De même, les travaux menés par Kohonen en 1982 ont montré que ces réseaux étaient capables de résoudre des tâches de classification et de reconnaissance [26].

3.1 Neurone biologique

Un neurone est une cellule composée d'un corps cellulaire contenant un noyau. Le corps cellulaire se ramifie pour former des structures appelées dendrites, qui peuvent être assez nombreuses pour former une chevelure dendritique ou une arborisation dendritique. Les dendrites permettent le transfert d'informations de l'extérieur vers le soma (corps du neurone). L'information traitée par le neurone est ensuite transmise le long de l'axone, qui est unique, pour être transmise à d'autres neurones. La transmission entre deux neurones n'est pas directe. En réalité, il existe un espace intercellulaire d'une taille de quelques dizaines d'angströms (10^{-9}m) entre l'axone du neurone afférent et les dendrites (on dit une dendrite) du neurone suivant [18].

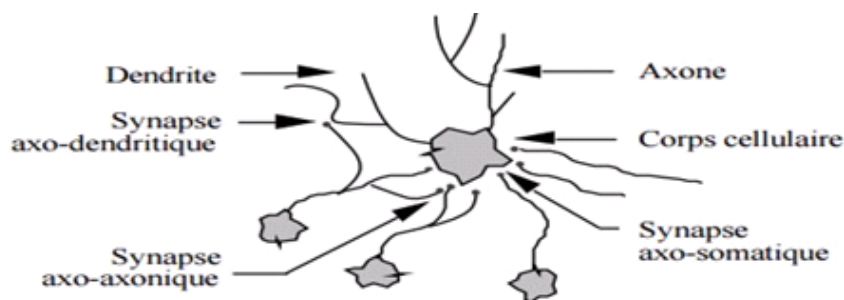


FIGURE 2.3 – Structure d'un neurone biologique [18].

3.2 Neurone artificiel

Un neurone artificiel (formel) est un élément de traitement basique qui est activé par les neurones qui le précèdent, appelés ses entrées. Chaque entrée est associée à un poids qui représente la force de la connexion, notée W . En fonction de la stimulation reçue, le neurone calcule une valeur de sortie. Cette sortie se propage ensuite pour alimenter un nombre variable de neurones en aval [18].

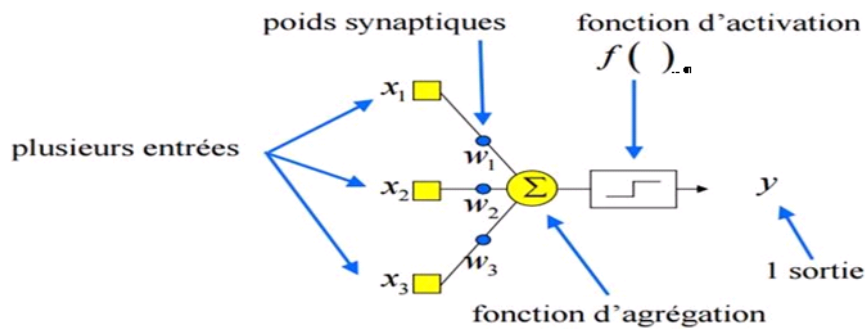


FIGURE 2.4 – Structure d'un neurone artificiel [18].

3.3 Correspondance entre neurone biologique et neurone artificiel

La structure d'un neurone artificiel est en réalité inspirée de la structure des neurones biologiques. Les éléments clés de la structure biologique des neurones ont leurs équivalents dans les neurones artificiels, dans le but de reproduire au mieux leur fonctionnement. Cette approche vise à représenter le fonctionnement des neurones de manière logique, simple et facilement modélisable sur un ordinateur [18].

Neurone biologique	Neurone artificiel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Somma	Fonction d'activation

TABLE 2.1 – Transition entre le neurone biologique et le neurone artificiel

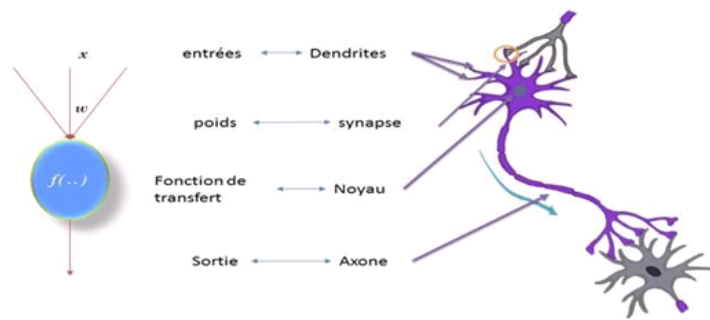


FIGURE 2.5 – Correspondance entre neurone artificiel et neurone biologique

3.4 Comportement de neurone artificiel

— **Entrées du neurone**

— **Le poids « W » :**

C'est une valeur numérique attribuée à une connexion entre deux unités (neurones), qui représente la force de la relation ou de la connexion entre ces deux unités i et j . Cette valeur est notée par W_i .

— **La fonction d'agrégation (combinaison) :**

Elle intègre les entrées et les poids en évaluant l'impact de chaque entrée en fonction de son poids. Cette évaluation est réalisée en utilisant la formule suivante :

$$\sum W_i X_i \quad (2.1)$$

Où W : le poids de la connexion à l'entrée et i . X : le signal de l'entrée i .

— **La fonction d'activation (transfert) :**

La fonction de transfert est un élément clé du comportement d'un neurone. Elle est responsable de renvoyer une valeur représentant l'activation du neurone. Elle calcule la sortie du neurone en utilisant le résultat de la fonction de combinaison : $S = F(P)$.

S : est la valeur de sortie, F : est la fonction de transfert.

La fonction de transfert utilisée dans les réseaux de neurones artificiels diffère d'un réseau à l'autre et est définie par des équations mathématiques spécifiques. Contrairement aux neurones biologiques, qui ont un état binaire, la plupart des fonctions de transfert utilisées dans les réseaux de neurones artificiels sont continues. Elles permettent ainsi une gamme infinie de valeurs, qui peuvent être comprises dans l'intervalle $[0, +1]$ ou $[-1, +1]$ [30].

3.5 Domaines d'application du réseau de neurones

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans divers domaines. Par exemple :

- Domaine du traitement d'images : englobe des tâches telles que la compression d'images, la reconnaissance de caractères et de signatures, la reconnaissance de formes et de motifs, la classification, etc.
- Domaine du traitement du signal : concerne des applications telles que le traitement de la parole, l'identification de sources, le filtrage, etc.
- Domaine du traitement automatique des langues : implique des activités telles que la segmentation en mots, l'étiquetage morphosyntaxique, la traduction automatique, etc.
- Contrôle : diagnostic de pannes, commande de processus, contrôle qualité, robotique, etc [26].

3.6 Avantages et Inconvénients du réseau de neurone

3.6.1 Avantage

- La capacité de représenter toutes sortes de fonctions, qu'elles soient linéaires ou non linéaires, simples ou complexes.
- Ils peuvent traiter des données bruitées et de classifier des modèles sur lesquels ils n'ont pas été spécifiquement entraînés.
- Ils sont particulièrement adaptés pour les entrées et les sorties de valeurs continues.
- Ils ont connu du succès sur plusieurs applications du monde réel, telles que la reconnaissance manuscrite de caractères, la pathologie médicale et les analyses en laboratoire, et bien d'autres [29].

3.6.2 Inconvénients

- Nécessitent de temps d'apprentissage prolongés, ils sont adaptés aux applications qui peuvent supporter cette durée d'apprentissage.
- La compréhension de la connaissance exprimée à travers un réseau de neurones, où les unités sont connectées par des liens pondérés, peut être difficile ou limitée.
- La détermination des paramètres, tels que la topologie du réseau, sa structure, le nombre d'unités dans chaque couche cachée ainsi que dans la couche de sortie, nécessite une approche empirique [29].

4 Algorithmes de Deep Learning

Le Deep Learning possède de nombreux algorithmes, chacun spécialisé dans un domaine d'application spécifique. Les trois grandes architectures de DL selon [25] :

- Les réseaux de neurones pré-entraînés non supervisés.
- Les réseaux de neurones récurrents (RNN).
- Les réseaux de neurones convolutifs (CNN).

en va présenter quelques-uns en se concentre sur notre objectif principal la détection de chute humaine.

4.1 Réseaux de neurones pré-entraînés non supervisés

Il existe plusieurs modèles génératifs pré-entraînés non supervisés qui peut être construit et entraîner, tous ces modèles représentent les distributions de probabilités sur plusieurs variables d'une certaine façon [20] Parmi ces modèles, nous pouvons citer :

- Réseaux de croyance profonde (DBN).
- Réseaux antagonistes génératifs (Gans).

4.2 Les réseaux de neurones récurrents (RNN)

Les réseaux neuronaux récurrents RNNs « Recurrent Neural Networks » sont une famille de réseaux de neurones, utilise des connexions entre les nœuds pour former un graphe dirigé, permettant ainsi le traitement de données séquentielles[20].

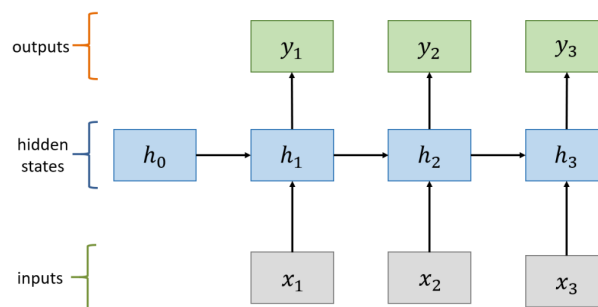


FIGURE 2.6 – Architecture de RNN standard

Ce mémoire se concentre sur l'étude des réseaux de neurones convolutifs, qui représentent la technique la plus couramment utilisée dans les domaines de la classification d'images et de la détection d'objets en DL.

4.3 Les réseaux de neurones convolutifs (CNN)

Les réseaux de neurones à convolution ont été créés dans les années 80. Désignés par l'acronyme CNN, de l'anglais « Convolutional Neural Network » [25]. Les réseaux de neurones convolutifs sont spécialement conçus pour traiter des entrées structurées en grille, où les données présentent des dépendances fortes. Les images sont un exemple typique de données structurées en grille, et elles sont particulièrement adaptées à l'utilisation de ce type de réseau de neurones.

Le concept et le fonctionnement des CNN est différent des autres réseaux de neurones, ils comportent deux parties clairement définies. En entrée, une image est fournie comme une matrice de pixels. Pour les images en niveaux de gris, cette matrice possède deux dimensions (bidimensionnelle, noir et blanc). La représentation de la couleur utilise une dimension supplémentaire de profondeur trois pour représenter les couleurs primaires (RVB) [14].

La première partie d'un réseau de neurones convolutif (CNN) est la partie convolutive elle-même. Cette partie joue le rôle d'extracteur de caractéristiques à partir des images. L'image d'entrée est soumise à une succession de filtres, également appelés noyaux de convolution, ce qui crée des images récemment acquises connues sous le nom de cartes de convolutions. Au final, les cartes de convolutions sont aplaties et concaténées en un vecteur de caractéristiques, connu sous le nom de code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite connectée en entrée d'une deuxième partie, qui est composée de couches entièrement connectées (perceptron multicouche). La sortie du CNN sera des neurones qui représentent une catégorie par neurone. Les valeurs typiquement obtenues sont souvent normalisées dans une plage allant de 0 à 1, ce qui permet de générer une répartition de probabilité sur les différentes catégories [19].

les réseaux de neurones convolutionnels tirent leur nom de l'utilisation de l'opération mathématique de convolution, qui représente un type spécifique d'opération linéaire. Cette structure ressemble à une grille [20].

4.3.1 Architecture d'un Réseaux neuronaux convolutifs (CNN)

Les réseaux de neurones convolutifs sont basés sur les perceptrons multicouches, qui s'inspirent du fonctionnement du cortex visuel des vertébrés. Bien qu'ils soient efficaces pour le traitement des images, les MLP rencontrent des difficultés lorsqu'il s'agit de gérer des images de grande taille. Cela est principalement dû à l'augmentation exponentielle du nombre de connexions requises en fonction de la taille de l'image [14].

Par exemple, si on prend une image de taille $64 \times 64 \times 3$ (64 de large, 64 de haut, 3 canaux de couleur), un seul neurone entièrement connecté dans la première couche cachée du MLP aurait 12 288 entrées ($64 \times 64 \times 3$). Ainsi, une image de taille 200×200 conduirait à traiter 384 000 entrées par neurone, ce qui, multiplié par le nombre de neurones, pourrait représenter une quantité de calcul importante [19].

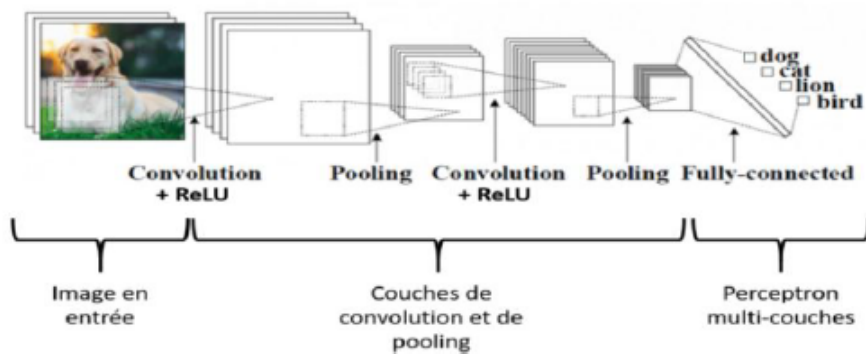


FIGURE 2.7 – Architecture de CNN standard [19].

Les blocs de construction

Une architecture (CNN) est constituée de couches de traitement empilées, chacune fonctionnant de manière indépendante [14] :

- Couche de convolution (CONV) est responsable de l'extraction des caractéristiques des données d'entrée
- Couche de pooling (POOL), réduit la taille de l'image intermédiaire en comprimant l'information.
- Couche de correction (ReLU), applique la fonction d'activation (de transfert).
- Couche "entièrement connectée" (FC) connue par le nom de couche de perceptron.
- Couche de perte (LOSS) pour calculer la perte ou l'erreur

On ne peut pas parler de couches convolutives sans parler des couches de pooling.

Couche de pooling

Le pooling est une opération essentielle dans les CNN. Il s'agit d'une forme de sous-échantillonnage de l'image intermédiaire [14]. Le pooling est un processus simple où nous réduisons la taille spatiale des images intermédiaires, ce qui permet de réduire le nombre de paramètres et de calculs dans le réseau. Afin de contrôler l'overfitting (sur-apprentissage), il est courant d'insérer périodiquement une couche de pooling entre deux couches convolutives successives dans une architecture CNN [19]. Il existe différents types de pooling :

- **Max pooling** : Cette opération renvoie la valeur maximale présente dans la région couverte par le filtre.
- **Average pooling** : Cette opération renvoie la moyenne des valeurs présentes dans la région couverte par le filtre.

Ainsi, en utilisant ces opérations de pooling, les réseaux de neurones convolutifs peuvent réduire la dimension des images intermédiaires tout en conservant les informations essentielles pour la détection d'objets [19]. Le fonctionnement de la couche de pooling est très simple, par exemple, considérons l'image d'entrée de 4×4 pixels. Une fois que l'image d'entrée passe à travers la couche de Pooling, elle se réduit en une image de 2×2 pixels (Figure 2.9).

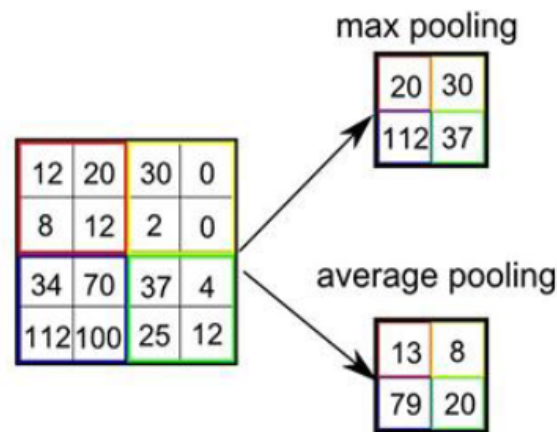


FIGURE 2.8 – résultats de max et average pooling

4.3.2 Les modèles réseaux de neurones convolutifs

Dans le domaine des réseaux convolutifs, il existe plusieurs modèles couramment utilisés, dont voici quelques exemples [32] :

LeNet : LeNet est considéré comme l'un des premiers réseaux neuronaux convolutifs (CNN) et a été développé par Yann LeCun et ses collègues dans les années 1990. Il a ensuite été amélioré en 1998. LeNet est devenu emblématique en raison de son application réussie dans la reconnaissance des codes postaux et des chiffres [32].

AlexNet : AlexNet, développé par Alex Krizhevsky et son équipe, est largement reconnu comme la première architecture CNN célèbre qui a révolutionné la vision par ordinateur. L'architecture du réseau présente certaines similitudes avec LeNet, mais elle était plus profonde et plus significative.

GoogLeNet : GoogLeNet, développée par Szegedy et son équipe de Google, a remporté le concours ILSVRC 2014, avec un faible taux d'erreur dans le top 5, se rapprochant ainsi des performances humaines.

VGGNet : Ce réseau final du concours ILSVRC 2014 a été développé par Karen Simonyan et Andrew Zisserman et désigné sous le nom de VGGNet. Ce réseau a réussi à extraire des caractéristiques complexes et discriminantes à partir des images, permettant une meilleure classification des objets dans des scénarios complexes.

ResNet : Dans les années 2015, Kaiming He et son équipe ont introduit le Residuel Network (ResNet), qui a remporté le concours ILSVRC 2015. Cette architecture de réseau neuronal convolutif (CNN) a marqué une avancée significative en utilisant des connexions résiduelles et en incorporant la normalisation par lots [32].

4.3.3 L'application des réseaux neuronaux convolutifs

Les réseaux neuronaux convolutifs (CNNs) ont de larges applications, comme :

- Vision par ordinateur.
- Médecine et imagerie médicale.
- Traitement du langage naturel.
- Véhicules autonomes
- La reconnaissance d'image et vidéo.

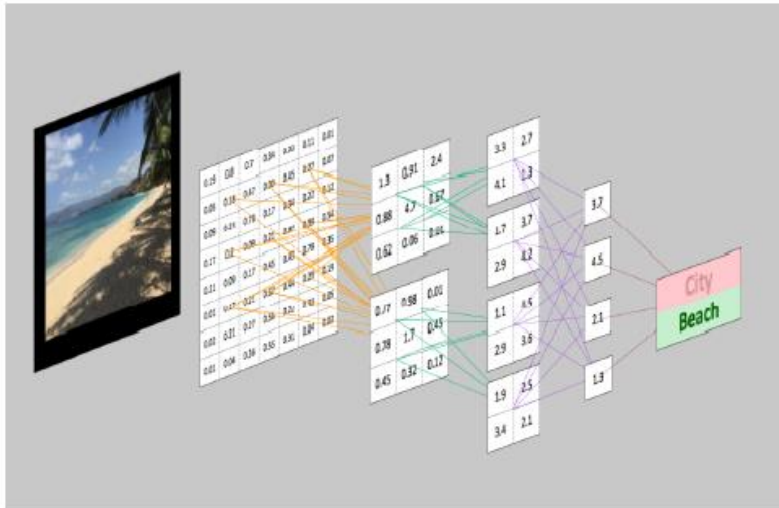


FIGURE 2.9 – Reconnaissance d'une image avec CNN [25].

Les CNN sont largement reconnus pour leur efficacité dans la reconnaissance d'images (Voir Figure 2.16) est l'une des principales raisons pour lesquelles le monde reconnaît le pouvoir de l'apprentissage profond [25].

5 Conclusions

Dans ce chapitre, nous avons exploré les concepts essentiels du Deep Learning et leur fonctionnement, ainsi que leur utilisation dans le domaine du traitement d'images. Ensuite, nous avons approfondi notre compréhension des réseaux de neurones, en mettant en évidence les similitudes entre les neurones biologiques et artificiels. Enfin, nous nous sommes concentrés sur les algorithmes de Deep Learning, en détaillant plus particulièrement les réseaux de neurones convolutionnels, qui se révèlent être des outils puissants pour extraire et classifier les caractéristiques des images.

Dans le prochain chapitre, nous allons procéder à une applications pratiques du concept des réseaux de neurones convolutionnels dans la détection de chute des personnes dans le cadre de la prévention.

Réalisation

1 Introduction

Face à l'augmentation alarmante du nombre de décès liés aux chutes humaines, il est devenu impératif de mettre en place des systèmes de détection efficaces. Dans cette étude, nous proposons l'utilisation d'un système embarqué pour la détection de chute humaine. Pour ce faire, nous avons choisi Raspberry Pi 4 ainsi que la caméra Raspberry.

Dans ce chapitre, nous aborderons les différentes bibliothèques utilisées, algorithmes et le langage de programmation choisis pour mettre en œuvre notre système. Ensuite, nous détaillerons l'implémentation de notre système et présenterons son processus d'entraînement sur la base de données.

2 Les outils de développement

Nous avons utilisé différents outils de programmation et de développement, tels que Python, OpenCV, etc., pour réaliser ce projet. Nous avons implémenté les deux algorithmes nécessaires en utilisant ces outils.

2.1 Base de données

Maldona et al. [38] ont introduit un ensemble de données de chute appelé FPDS abréviation de "Fall Detection Performance Standard", est une base de données utilisée dans le domaine de la détection de chute. C'est une ressource précieuse pour l'entraînement et l'évaluation des modèles de détection de chute. Toutes les images ont été prises à l'aide d'une seule caméra insérée dans un robot à 76 cm au-dessus du sol. FPDS comprenait un total de 2062 images étiquetées manuellement avec 1072 chutes

et 1262 (debout, en marchant, couché, assis, etc.) étiquetées manuellement. Plus d'un sous-objet peut se trouver dans une seule image. La hauteur des sous-objets est comprise entre 1,2 m et 1,8 m. Les activités ont été réalisées dans 8 environnements différents avec des conditions variées de lumière, d'ombres, de réflexions, etc.

L'ensemble de données FDPS est généralement utilisé pour entraîner des modèles de détection de chute en utilisant des techniques d'apprentissage automatique, notamment le Deep Learning.



FIGURE 3.1 – Exemples d'images de FPDS.

2.2 Méthode utilisée

Une détection humaine en temps réel est une tâche difficile, elle nécessite beaucoup de traitement. Dans notre système nous utiliserons deux algorithmes.

2.2.1 L'architecture de YOLOv5

YOLOv5 est une version améliorée de l'algorithme YOLO (You Only Look Once) qui est utilisée pour la détection d'objets dans des images, ce qui signifie que le détecteur recherche des objets spécifiques comme des personnes ou des voitures dans une image [15]. Il utilise une approche basée sur des grilles pour diviser les images en petites régions, où chaque région est responsable de la détection d'objets spécifiques [21].

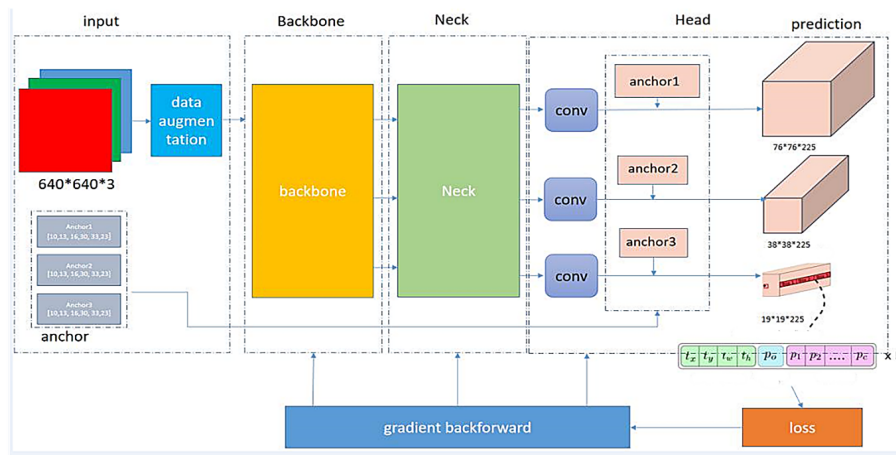


FIGURE 3.2 – Structure du réseau de l’algorithme YOLO.

L’architecture de YOLOv5 comprend deux couches entièrement connectées et 24 couches de convolution. Après les couches entièrement connectées, un tenseur de dimension $S \times S \times (B \times 5 + C)$ est produit, où B représente le nombre de cibles prédites dans chaque grille et C est le nombre de catégories d’objets à détecter.

L’algorithme YOLOv5 est capable de détecter rapidement les objets, mais il peut avoir des difficultés à détecter de petites cibles ou son efficacité peut être réduite dans de tels cas. Pour remédier à cela, YOLOv5 utilise une division plus détaillée des grilles pour mieux localiser les objets dans l’image [21].

De plus, YOLOv5 utilise un chargeur de données qui améliore les données d’entraînement en effectuant des opérations telles que la mise à l’échelle, l’ajustement de l’espace colorimétrique et l’amélioration de la mosaïque. Ces améliorations permettent d’augmenter la diversité des données d’entraînement et d’améliorer les performances de détection.

2.2.2 L’architecture de VGG-16

VGG-16 est un réseau de neurones convolutifs (CNN) développé par le ‘Visual Geometry’ Group de l’Université d’Oxford en 2014. Il est caractérisé par sa profondeur avec 16 couches, dont 13 couches de convolution et trois couches entièrement connectées. Il a été conçu pour la classification d’images en utilisant une entrée de taille fixe de 224×224 pixels et une sortie de probabilités pour 1000 classes [22].

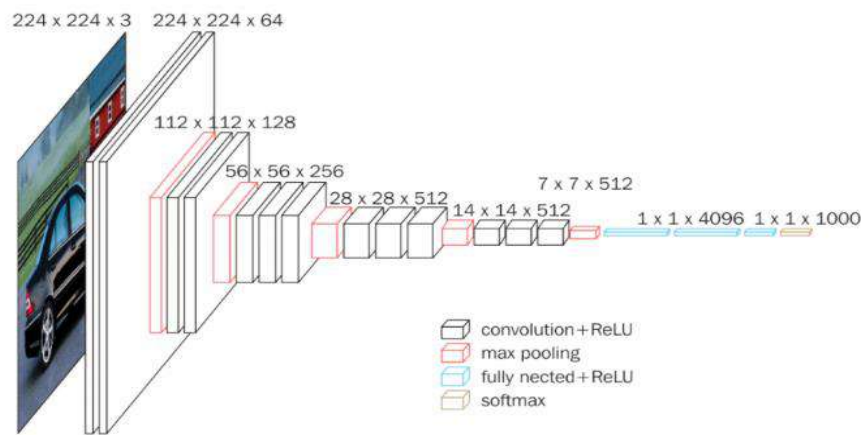


FIGURE 3.3 – Représentation 3D de l'architecture du VGG-16.

L'architecture VGG-16 est conçue pour la classification d'images et prend en entrée des images RGB de taille 224×224 pixels. Les couches de convolution utilisent des filtres de taille 3×3 pour capturer les informations spatiales, et des filtres de convolution 1×1 pour effectuer des transformations linéaires des canaux d'entrée. Des fonctions d'activation (généralement ReLU) sont appliquées après chaque couche de convolution pour introduire de la non-linéarité. Des opérations de max pooling sont utilisées pour réduire la résolution spatiale [23].

Les couches entièrement connectées ont 4096 canaux, et la dernière couche est une couche de classification avec 1000 canaux correspondant aux classes de l'ILSVRC. L'architecture VGG-16 est connue pour sa simplicité et son uniformité, bien qu'elle soit relativement profonde et nécessite des ressources de calcul importantes [22].

VGG-16 est connu pour son architecture simple et uniforme, avec des couches de convolution de taille fixe, des filtres de petite taille et des couches entièrement connectées en fin de réseau. Bien qu'il soit relativement profond, il a été largement utilisé en raison de ses performances solides dans la classification d'images.

2.3 Langage de programmation

Le langage de programmation choisi pour notre méthode s'est concentré sur le langage python

2.3.1 Python

Python est un langage de programmation puissant et facile. Il offre des fonctionnalités de haut niveau et permet une approche simple mais efficace de la programmation orientée objet et disponible gratuitement pour toutes les plateformes [34].

Il est largement utilisé dans le domaine de l'informatique et de la science des données. Python propose également de nombreuses bibliothèques (packages) pour le traitement des données, les calculs matriciels, l'analyse et la visualisation des données.

Pourquoi Python ?

Python est utilisé dans le domaine du Deep Learning grâce à des environnements de travail tels que TensorFlow, Keras et PyTorch. Ces environnements offrent des fonctionnalités essentielles pour la gestion de l'algorithme de rétropropagation, qui est crucial pour entraîner de grands réseaux de neurones, notamment les réseaux de neurones convolutionnels (CNN).

Les avantages du choix de Python sont les suivants :

- C'est totalement gratuit.
- Facilité d'apprentissage et de lecture.
- Il fonctionne sur tous les principaux systèmes d'exploitation et plates-formes informatiques.

2.4 Bibliothèques utilisées

Pour les besoins du Deep Learning, on a généralement besoin des bibliothèques suivantes :

2.4.1 OpenCV

OpenCv, qui signifie Open Source Computer Vision, est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de traitement d'images et de vision par ordinateur. Elle est accessible via des API pour les langages C, C++, et Python. OpenCV est distribuée sous une licence BSD libre et est compatible avec toutes les plateformes. Cette bibliothèque est aujourd'hui développée, maintenue, documentée et utilisée par une communauté active de plus de 40 000 membres [40].

En utilisant OpenCV avec Python, les développeurs peuvent bénéficier de fonctionnalités avancées pour le Deep Learning, telles que la détection et la reconnaissance d'objets, le suivi de mouvement.

2.4.2 OpenPifPaf

OpenPifPaf est une bibliothèque open-source puissante pour la détection de pose humaine basée sur le Deep Learning. Elle permet aux développeurs et aux chercheurs d'exploiter les réseaux de neurones convolutifs pour localiser et estimer les poses humaines dans des images ou des vidéos, ouvrant ainsi la voie à de nombreuses applications pratiques [31].

2.4.3 PyTorch

PyTorch une bibliothèque open source développée par Facebook AI Research. Est un cadre complet pour la création de modèles Deep Learning, qui est un type de Machine Learning est couramment utilisé dans une variété d'applications, notamment la reconnaissance d'images et la détection d'objets. Écrit en Python, il est relativement facile à apprendre et à utiliser pour la plupart des développeurs d'apprentissage automatique [33].

PyTorch se distingue par son excellent support pour les GPU et son utilisation de l'auto-différenciation en mode inverse, qui permet de modifier les graphes de calcul. Cette combinaison en fait une option largement préférée pour la création rapide de prototypes et l'expérimentation [33].

3 Implémentation

Cette section décrit le matériel utilisé pour effectuer les expérimentations, ainsi que le fonctionnement de notre système :

3.1 Matériel utilisé

Pour faire l'implémentation du système de détection de chute humaine, nous avons choisi d'utiliser une carte RPi 4 model B et la caméra Pi (1.3v).

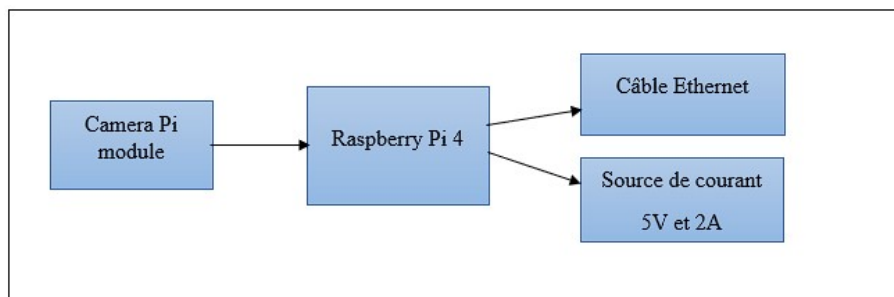


FIGURE 3.4 – Schéma fonctionnel du matériel

3.1.1 Raspberry pi 4 :

Le Raspberry Pi est un nano-ordinateur mono-carte équipé d'un processeur ARM. Il a été développé en collaboration avec des professeurs de l'université de Cambridge par la Fondation Raspberry Pi. Avec sa taille similaire à une carte de crédit, il peut être connecté à un écran, permettant de l'utiliser comme un ordinateur standard [24]. Parmi les différentes versions du Raspberry Pi, le modèle le plus récent est le Raspberry Pi 4 Model B.



FIGURE 3.5 – Carte Raspberry pi 4 model B

Voici les caractéristiques principales du Raspberry Pi 4 modèle B : [39]

- Processeur ARM Cortex-A72 quadricœur 64 bits cadencé à 1,5 GHz (3 fois plus puissant que les modèles précédents)
- Mémoire SDRAM LPDDR4 de 1 Go, 2 Go, 4 Go ou 8 Go
- Gigabit Ethernet pour des connexions réseau haut débit
- Connectivité sans fil double bande 802.11ac
- Bluetooth 5.0 pour une communication sans fil
- Deux ports USB 3.0 et deux ports USB 2.0 pour connecter des périphériques
- Prise en charge de deux moniteurs avec des résolutions jusqu'à 4K
- Graphiques VideoCore VI prenant en charge OpenGL ES 3.x
- Décodage matériel 4Kp60 de la vidéo HEVC (High Efficiency Video Coding)

Le fonctionnement :

Avant de commencer à explorer les cas d'utilisation du Raspberry Pi, il est essentiel de comprendre son fonctionnement. Contrairement à un ordinateur traditionnel, le RPi ne possède pas de stockage interne en raison de sa taille compacte. Au lieu de cela, les données sont stockées sur une carte SD.

Pour utiliser le Raspberry Pi, il est essentiel d'avoir [27] :

- Une carte micro SD (de 16 GB ou plus).
- Un câble d'alimentation.
- Un câble RJ45 est utilisé pour établir une connexion réseau.
- Un câble HDMI permet de relier un RPi à un moniteur.
- Un clavier et souris pour taper les commandes

3.2 Installation et configuration

3.2.1 Préparation de Raspberry Pi

Installation du système d'exploitation :

Il existe une grande variété de systèmes d'exploitation disponibles pour le RPi. Pratiquement tous les systèmes d'exploitation conçus pour être compatibles avec les processeurs ARM peuvent être installés sur le RPi.

Raspbian : En 2021, Raspbian a changé de nom, et devient Raspberry Pi OS, est le système d'exploitation officiel spécialement développé pour le Raspberry Pi. Il est basé sur Linux Debian (raspberrypi + debian = raspbian = Raspberry Pi OS) et il est très régulièrement mis à jour [39].

Raspberry Pi a développé un outil graphique d'écriture de carte SD qui fonctionne sur Mac OS et Windows appelé Raspberry Pi Imager ; c'est l'option la plus simple pour la plupart des utilisateurs, car elle téléchargera automatiquement le système d'exploitation et l'installera sur la carte SD.

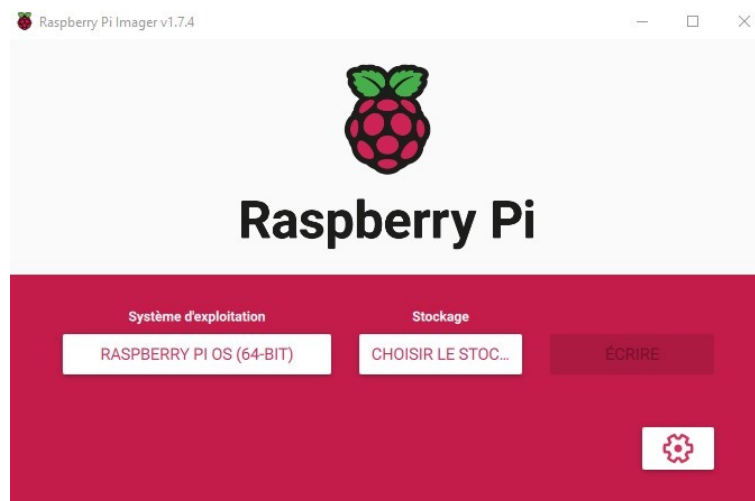


FIGURE 3.7 – Raspberry Pi Imager

Suivez les étapes du logiciel Pi OS Imager pour continuer. Une fois que votre carte microSD est imagée, insérez-la dans votre RPi et suivez les instructions de configuration.

Accès à distance au Raspberry Pi

À cette étape, nous avons attribué une adresse IP fixe à notre carte, ce qui nous permettra de la manipuler à distance en utilisant les protocoles SSH et VNC. Cette adresse sera également utilisée pour : Connexion à distance via SSH :

Dans mon cas j'ai utilisé « PuTTY » est un programme client SSH, Telnet et Rlogin gratuit et open source, largement utilisé pour se connecter à des serveurs à distance. Il fournit une interface graphique conviviale pour établir des connexions sécurisées à des machines distantes et exécuter des commandes à distance [35].

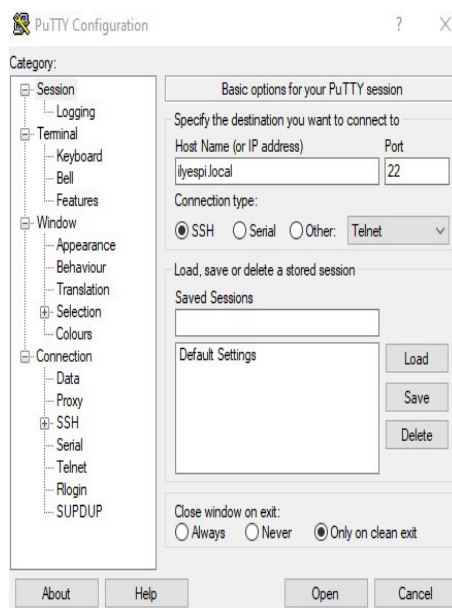


FIGURE 3.8 – Le client « PuTTY »

Connexion a VNC Viewer

Après avoir établi une connexion à distance via SSH avec PuTTY, vous pouvez procéder à l'étape de configuration de VNC pour accéder à l'interface graphique de votre Raspberry Pi à distance.

Pour commencer, assurez-vous que VNC Server est installé sur votre Raspberry Pi en exécutant la commande 'sudo apt-get install realvnc-vnc-server' via SSH. Une fois installé, lancez VNC Server en utilisant la commande 'vncserver'.

Lancez VNC Viewer sur votre ordinateur et entrez l'adresse IP ou nom d'hôte de votre Raspberry Pi. VNC Viewer est un logiciel client utilisé pour accéder à distance à un ordinateur ou à un périphérique qui exécute un serveur VNC (Virtual Network Computing). Il permet aux utilisateurs de visualiser et de contrôler l'interface graphique d'un ordinateur comme Raspberry Pi distant depuis leur propre ordinateur local [36].

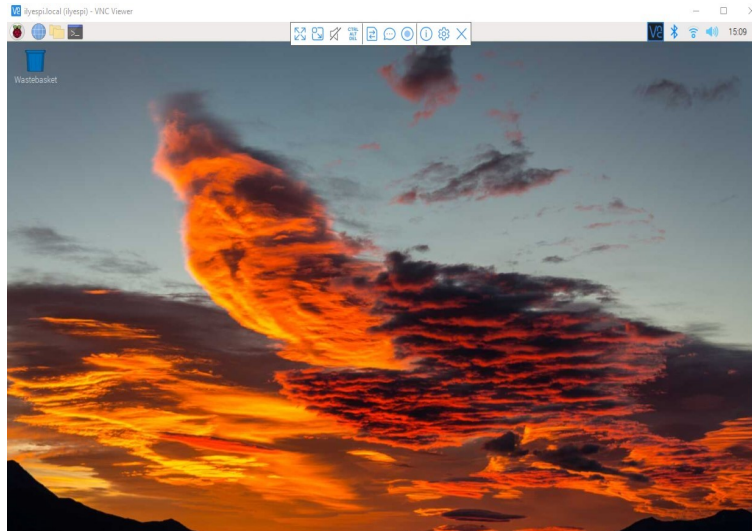


FIGURE 3.9 – Connexion Raspberry par VNC Viewer

3.2.2 Configuration de la Caméra Pi

Pour configurer la caméra Raspberry Pi 1.3V, assurez-vous tout d'abord que la caméra est correctement connectée à votre Raspberry Pi. La caméra doit être connectée à travers la nappe de connexion qui est insérée dans le connecteur CSI (Camera Serial Interface) de la carte Raspberry Pi.

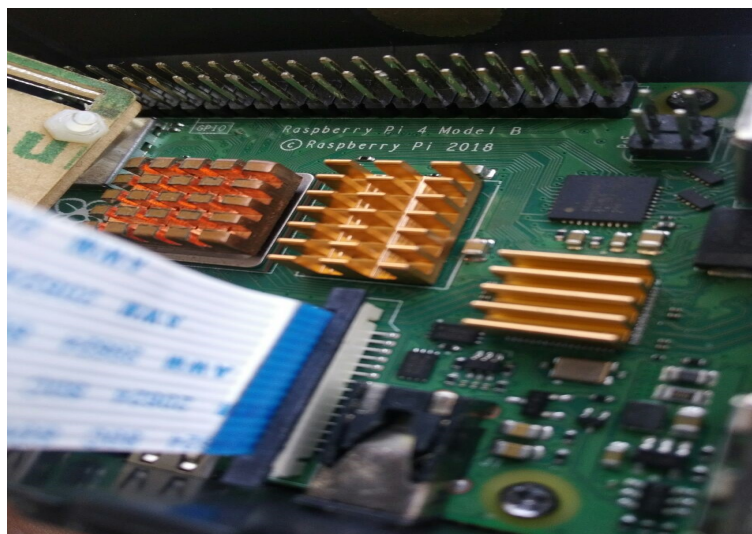


FIGURE 3.10 – le connecteur CSI de la carte Raspberry Pi.

Une fois la connexion physique vérifiée, vous devez activer la caméra dans la configuration de votre Raspberry Pi. Pour cela, vous pouvez utiliser la commande suivante dans le terminal : "sudo raspi-config". Cette commande ouvre le menu de configuration de la Raspberry Pi. À partir de là, vous pouvez sélectionner l'option "Interface Options" ou "Options d'interface", puis choisir "Legacy Camera" pour activer la caméra.

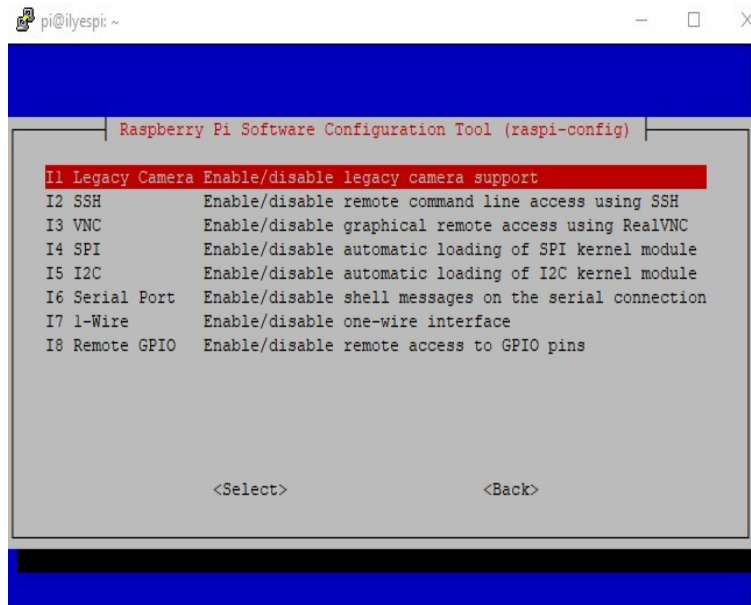


FIGURE 3.11 – Activation de la caméra Pi

3.3 Test et Résultat

3.3.1 Apprentissage des algorithmes

L'apprentissage des algorithmes utilisés dans ce système repose sur l'utilisation d'un ensemble de données spécifique appelé Fall detection Data set (FDPS). Ce jeu de données est préparé de manière à inclure des exemples de chutes et de mouvements normaux, permettant ainsi aux algorithmes d'apprendre à distinguer les deux.

Pour l'algorithme YOLOv5, nous avons entraîné le modèle à détecter les personnes dans les images en utilisant un processus d'apprentissage supervisé. Nous avons fourni au modèle des images annotées où les positions des personnes. Le modèle a ensuite appris à associer les caractéristiques visuelles spécifiques aux personnes pour les localiser et les suivre dans les vidéos.

Quant à l'algorithme VGG-16, nous avons utilisé un processus similaire d'apprentissage supervisé. Nous avons fourni au modèle des images annotées où chaque image était étiquetée comme "chute" ou "mouvement normal". En analysant les caractéristiques visuelles extraites des images, le modèle a appris à reconnaître les motifs distinctifs des chutes et des mouvements normaux.

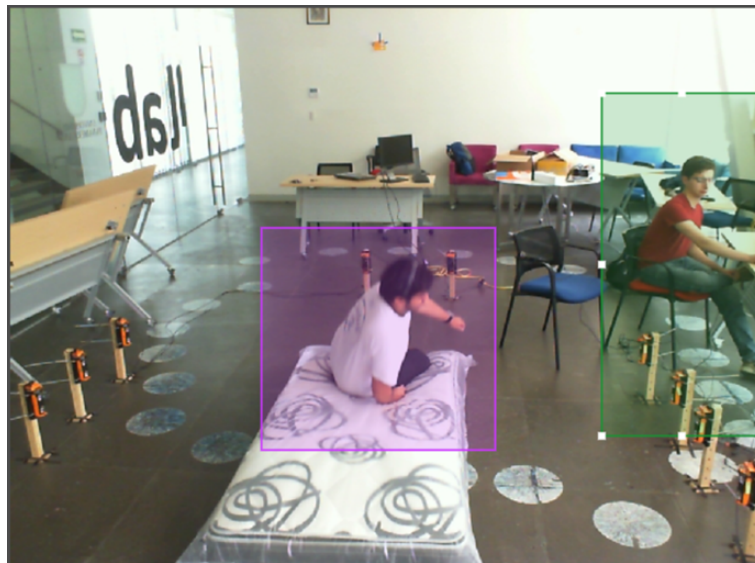


FIGURE 3.12 – Image du FDPS annotée avec des bounding boxes (encadrés)

Le processus d'apprentissage des deux algorithmes a nécessité une phase d'entraînement où les modèles ont été ajustés aux données du Fall detection Data set. Les modèles ont été optimisés à l'aide de techniques telles que la rétropropagation du gradient et l'optimisation des poids pour maximiser leurs performances de détection et de classification.

3.3.2 Mesures d'évaluation

Un problème de détection de chute peut être considéré comme un problème binaire. Dans un problème binaire, nous avons deux résultats pour chaque entrée, Vrai (V) et Faux (F). Pour un problème de détection de chute, T signifie qu'il y a eu une chute et F signifie qu'il n'y a pas eu de chute.

De nombreux travaux ont été réalisés pour détecter les chutes. Pour comparer et évaluer les performances des expériences menées par différents chercheurs, de nombreuses mesures ont été conçues. Certaines des métriques importantes utilisées par la plupart des chercheurs sont les suivantes [37] :

Sensibilité (Recall) : La sensibilité signifie que le degré d'exactitude d'une expérience peut détecter une chute réelle comme une chute.

$$\text{Sensibilité} = TP / (TP + FN) \quad (3.1)$$

F-Score : Le score F, également appelé score F1, est une mesure de la précision d'un modèle sur un ensemble de données. Il est utilisé pour évaluer les systèmes de classification binaire, qui classent les exemples en « positif » ou « négatif »

$$F1Score = 2(\text{PrécisionRecall}) / (\text{Précision} + \text{Recall}) \quad (3.2)$$

Accuracy : est la proportion de vrais résultats parmi le nombre total de cas examinés.

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN) \quad (3.3)$$

Précision : L'utilisation de la précision en tant que métrique d'évaluation est justifiée lorsque nous souhaitons avoir une grande confiance dans nos prédictions, comme le montre l'équation suivante :

$$\text{Précision} = (TP) / (TP + FP) \quad (3.4)$$

Toutes les mesures susmentionnées peuvent être définies en termes de TP (vrais positifs), TN (vrais négatifs), FP (faux positifs), and FN (faux négatifs).

- TP = la détection correcte des chutes.
- TN = la détection des activités de la vie quotidienne en tant que chute.
- FP = le nombre de chutes non détectées comme des chutes. De même.
- FN = la détection d'une activité normale comme une chute [37].

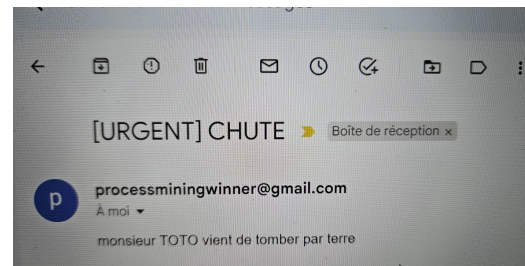
3.3.3 Le protocole SMTP

L'envoi de courrier électronique est une fonctionnalité essentielle de notre projet de détection de chute. Une fois qu'une chute est détectée par notre système, nous avons développé un programme qui envoie automatiquement un courrier électronique pour informer les destinataires appropriés de l'incident.

Pour mettre en œuvre cette fonctionnalité, nous avons utilisé le protocole SMTP (Simple Mail Transfer Protocol), qui est un protocole standard pour l'envoi de courrier électronique sur Internet. Nous avons choisi d'utiliser un compte Google pour l'envoi des courriers électroniques.

```
283
284
285
286
287
288
289
290
if 'F' in activity_dict[prediction+5]:
    # Send email alert
    subject = "[URGENT] CHUTE"
    message = "monsieur TOTO vient de tomber par terre"
    receiver = "ilyes.sd10@gmail.com"
    send_email_alert(subject, message, receiver)
```

(a) SMTP Programme



(b) recevoir de message

FIGURE 3.13 – Le résultat de l'algorithme de segmentation par classification.

Le programme que nous avons développé utilise une bibliothèque ou un module compatible avec le protocole SMTP pour établir une connexion sécurisée avec les serveurs de courrier électronique de Google. Il utilise ensuite les informations d'identification du compte configuré avec l'app password pour s'authentifier et envoyer le courrier électronique contenant les détails de la chute détectée.

Cette fonctionnalité d'envoi de courrier électronique s'est révélée essentielle pour informer rapidement les personnes concernées en cas de détection de chute, leur permettant de prendre des mesures appropriées dans les plus brefs délais.

3.3.4 Fonctionnement de système

Le système repose sur l'utilisation de la Raspberry Pi 4, qui agit comme une plateforme de calcul, et de la caméra Pi, qui capture les vidéos en temps réel. Pour détecter le mouvement des personnes dans la vidéo, on a utilisé l'algorithme YOLOv5 et la base de données FDPS. YOLOv5 est un réseau de neurones convolutionnels (CNN) conçu pour la détection d'objets, et FDPS est une base de données d'images annotées qui a été utilisée pour l'entraînement du modèle. Cette étape permet d'identifier et de suivre les personnes présentes dans la vidéo.

Ensuite, on a utilisé l'algorithme VGG-16, un autre réseau de neurones convolutionnels, en conjonction avec la base de données FDPS, pour détecter les chutes. VGG-16 est connu pour sa capacité à extraire des caractéristiques visuelles et à effectuer des classifications. Le modèle entraîné sur la base

de données FDPS a été utilisé pour analyser les mouvements détectés précédemment et déterminer si une personne a chuté ou non.

Pour obtenir des informations plus détaillées sur la position de la personne détectée, nous avons utilisé OpenPifPaf, une bibliothèque de vision par ordinateur qui permet d'extraire les coordonnées précises des parties du corps humain, telles que les épaules, les coudes, les genoux, etc. Cela te permet de localiser précisément la position de la personne dans la vidéo.

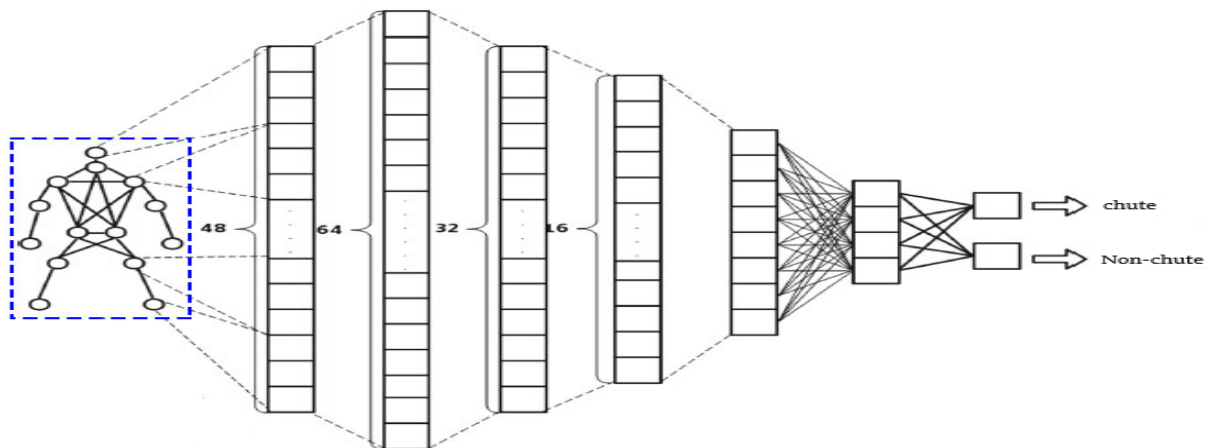


FIGURE 3.14 – VGG-16 et YOLOv5 basé sur le Deep Learning

Enfin, lorsque la détection de chute se produit, nous avons mis en place un protocole SMTP pour envoyer un e-mail d'alerte. Ce protocole permet d'envoyer rapidement une notification à une personne désignée ou à un système de sécurité, afin qu'ils puissent prendre les mesures appropriées.

Ce système combine donc la puissance du Deep Learning, en utilisant des algorithmes avancés tels que YOLOv5 et VGG-16, avec la flexibilité et la facilité d'utilisation de Raspberry Pi. Il offre une solution efficace pour détecter les chutes dans les vidéos de surveillance, ce qui peut être particulièrement utile dans les environnements où la sécurité et la prévention des accidents sont primordiales, tels que les maisons de retraite, les hôpitaux ou les lieux de travail.

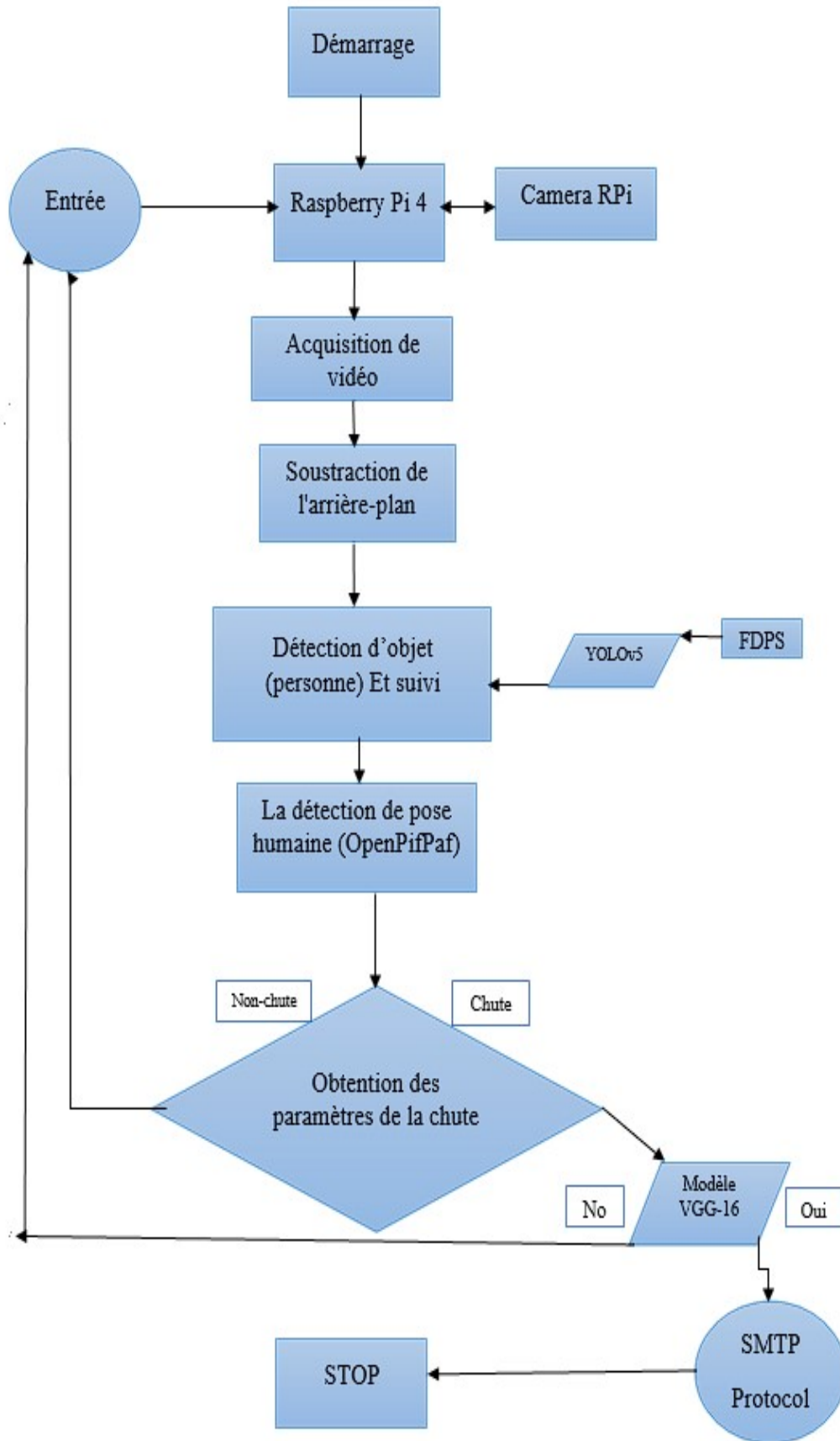
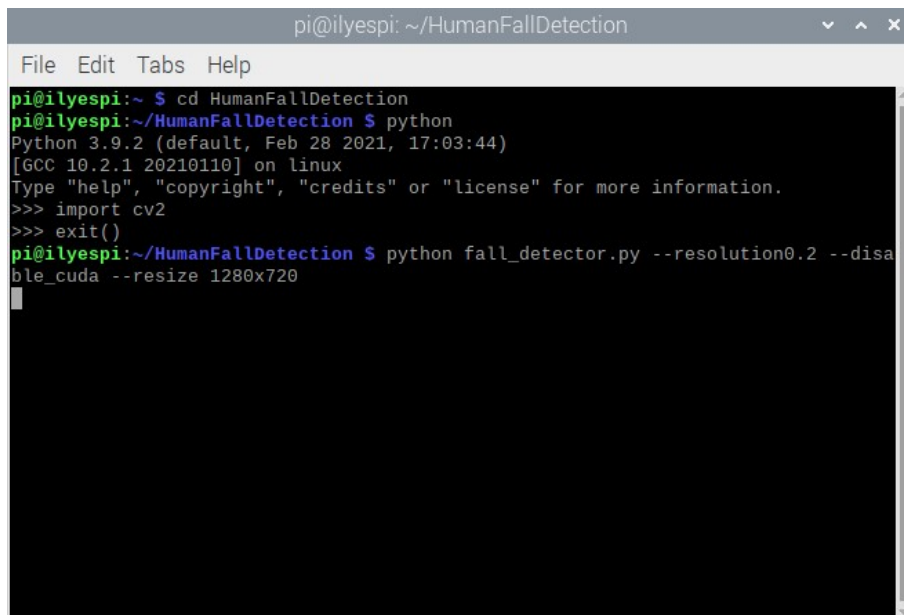


FIGURE 3.15 – L’organigramme de système détection de chute

3.4 Résultats d'implémentation

Finalement, après avoir consacré du temps et des efforts à la réalisation de notre projet, nous avons atteint l'étape cruciale de le tester sur le Raspberry Pi 4. Ce moment marque une étape importante dans notre parcours, car il nous permet de mettre en pratique les concepts théoriques et les algorithmes que nous avons étudiés. En utilisant la puissance de calcul du Raspberry Pi 4 et en exploitant les fonctionnalités de la caméra Pi, nous sommes en mesure d'effectuer des tests en temps réel sur des images réelles.

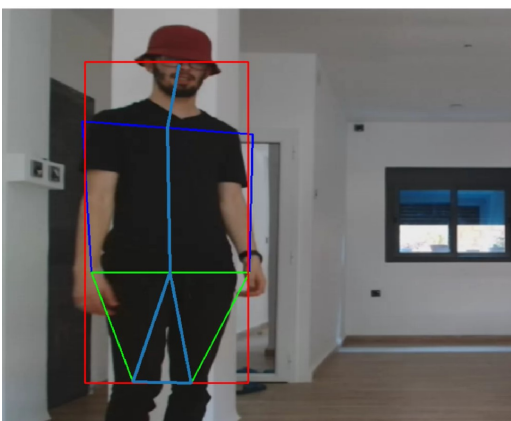


```
pi@ilyespi: ~/HumanFallDetection
File Edit Tabs Help
pi@ilyespi:~ $ cd HumanFallDetection
pi@ilyespi:~/HumanFallDetection $ python
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> exit()
pi@ilyespi:~/HumanFallDetection $ python fall_detector.py --resolution0.2 --disable_cuda --resize 1280x720
```

FIGURE 3.16 – Démarrage de système

Après l'exécution de cette instruction, la vidéo se lance, permettant ainsi d'analyser le contenu en temps réel.

Avg FPS: 3.0, Frame: 2120 Pred: Normal



(a) Détection Normal

Avg FPS: 3.0, Frame: 2191 Pred: FALL



(b) Détection de chute

FIGURE 3.17 – Test du système de détection de chute.

4 Conclusion

Dans le dernier chapitre, nous avons introduit le langage de programmation, les bibliothèques et les algorithmes utilisés pour mettre en œuvre notre projet. Ensuite, nous avons abordé les détails de l'implémentation, en décrivant le matériel utilisé et la configuration de notre système. Nous avons ensuite présenté les phases d'entraînement et de test, ainsi que les résultats obtenus lors de ces différentes étapes.

Conclusion générale

Ce travail rentre dans le cadre du projet de fin d'étude pour l'obtention du diplôme « Master en Electronique des Systèmes Embarqués ».

En guise de conclusion, nous rappelons que l'objectif de ce travail est de développer un système de détection de chute humaine en temps réel en utilisant des techniques de Deep Learning.

Dans ce rapport, nous avons abordé différents aspects liés à la détection de chute humaine en temps réel en utilisant des techniques de Deep Learning. Au cours des trois chapitres, nous avons exploré les notions de base sur l'image et les techniques de traitement d'image, les principes fondamentaux du Deep Learning avec un accent particulier sur les réseaux de neurones convolutifs (CNN), ainsi que la réalisation et l'implémentation pratique de notre projet en utilisant la Raspberry Pi 4, la caméra Pi et des algorithmes.

En conclusion, ce projet de détection de chute humaine en temps réel a été une expérience enrichissante et stimulante. Nous avons pu appliquer les connaissances acquises dans les domaines du traitement d'image et du Deep Learning pour développer un système fonctionnel capable de détecter les chutes et d'envoyer des alertes en temps réel. Bien que ce projet présente certaines limites, telles que la dépendance à l'utilisation de certaines bases de données et l'utilisation d'un matériel spécifique, il ouvre également des perspectives intéressantes pour des améliorations futures. Des efforts supplémentaires pourraient être déployés pour améliorer les performances du modèle de détection, explorer d'autres architectures de réseaux de neurones ou étendre le système à d'autres applications de surveillance et de sécurité.

Je souhaite que ce travail soit au niveau de la tâche qui m'a été confiée, et qu'il sera utile pour les utilisateurs.

Bibliographie

- [1] World Health Organization. WHO Global Report on Falls Prevention in Older Age. World Health Organization, Geneva, Switzerland, 2007.
- [2] A. MEDJAOUI, F. FARES. « Segmentation des Images par Contours Actifs : Application sur les Images Satellitaires à Haute Résolutions ». Mémoire de master. L'université Abou Bakr Belkaid–Tlemcen Faculté des sciences : Département d'Informatique, 2012.
- [3] M. BERGOUNIOUX, " Introduction au traitement mathématique des images-méthodes déterministes ", EDITION SPRINGER, 2015.
- [4] M. Sandeli, "Traitement d'images par des approches bio-inspirées application à la segmentation d'images". Thèse de doctorat, Université Constantine 2, 2014.
- [5] M. BENALI. " Reconnaissance Automatique des Chiffres Manuscrits ". Thèse de doctorat, Université Abou Bakr Belkaid– Tlemcen, 2017.
- [6] R. Gonzalez, E. Woods, "Digital Image Processing", 1992.
- [7] J-P. COCQUEREZ et S. PHILIPP. " Analyse d'images : Filtrage et segmentation". p. 84-100, 1995.
- [8] A. NAKIB, "Conception de métaheuristiques d'optimisation pour la segmentation d'images, Application aux images biomédicales". Thèse de doctorat, Université Paris 12 -Val de Marne, UFR de Sciences et Technologie, pp. 1-23, 2007.
- [9] J.J ROUSSELLE. " Les Contours Actifs : Une méthode de segmentation, Application a l'imagerie médicale". Thèse de doctorat, Université François Rabelais de Tours, p. 18-21, 2003.
- [10] C. Houassine, "Segmentation d'images par une approche biomimétique hybride". Université M'hamed Bougara– Boumerdes, 2012.
- [11] R.C. Gonzalez, R.E. Woods, "Digital Image Processing", 3rd Edition, Pearson Prentice Hall, New Jersey, 2008.
- [12] M. Bhattacharya, D. Singh Rajoot, P. K. Singh. "A New Technique For Feature Selection And Cluster Center Initialization". p. 4-5, 2010.
- [13] Dishayloo. "Seuillage d'image". Consulté le 18/05/2023 sur le site : <https://commons.wikimedia.org/wiki/File:RauschBild.png>.
- [14] M. Zakari. "Classification des images avec les réseaux de neurones". Mémoire de master., L'université de Tlemcen, 2017.
- [15] Redmon, Joseph, et al, "You Only Look Once: Unified, Real-Time Object Detection ", IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

- [16] C. Deluzarche, "Deep Learning". Futura Tech. Disponible sur : <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262> , consulté le : 15/05/2023.
- [17] <https://fr.mathworks.com/discovery/deep-learning.html> , consulté le : 20/05/2023.
- [18] KOUAME KHODISSOU BARETY J-B. " ETUDE ET MISE EN PLACE D'UNE INTELLIGENCE ARTIFICIELLE AUTONOME CAPABLE DE SAUVEGARDER ET SÉCURISER DES DONNÉES INFORMATIQUES". 2019.
- [19] R. IDOUI, A. HIMEUR, "Reconnaissance d'images par les réseaux de neurones convolutionnels (CNN)", Université Mohammed Seddik Ben Yahia- Jijel, 2020.
- [20] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning". MIT Press. 2016.
- [21] <https://doi.org/10.1371/journal.pone.0259283> , consulté le : 02/04/2023.
- [22] K. SIMONYAN, A. ZISSERMAN, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION ". Visual Geometry Group, Department of Engineering Science, University of Oxford.
- [23] <http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420> , consulté le 25 Mai 2023.
- [24] Wikipédia « Raspberry Pi », consulté le 25 Mai 2023 sur le site : https://fr.wikipedia.org/wiki/Raspberry_Pi
- [25] J. Patterson, A. Gibson, "Deep Learning: A Practitioner's Approach". Beijing: O'Reilly Media, 2017.
- [26] P. Wira. " Réseaux de neurones artificiels : Architecture et applications ". p. 10-28, 2009.
- [27] <https://www.grafikart.fr/blog/raspberry-pi-utilisation>, consulté le 25 Mai 2023.
- [28] https://fr.wikiversity.org/wiki/R%C3%A9seaux_de_neurones/Applications_des_r%C3%A9seaux_de_neurones , consulté le 25 Mai 2023
- [29] H. BHAVSAR, A. GANATRA, "A COMPARATIVE STUDY OF TRAINING ALGORITHMS FOR SUPERVISED MACHINE LEARNING", INT. J. SOFT COMPUT. ENG (IJSCE), VOL. 2 (N°4), pp. 2231-2307. 2012.
- [30] R. GHAYOULA, "Contribution à l'Optimisation de la Synthèse des Antennes Intelligentes par les Réseaux de Neurones", 2008.
- [31] <https://pypi.org/project/openpifpaf/> , consulté le : 02/05/2023.
- [32] S. Sakib et al. " An Overview of Convolutional Neural Network: Its Architecture and Applications ". Independent University of Bangladesh, Bangladesh, 2018.
- [33] <https://pytorch.org> , consulté le : 02/05/2023.
- [34] <https://www.python.org> , consulté le : 02/05/2023.
- [35] Wikipédia « PuTTY », consulté le 10/04/2023 sur le site : <https://fr.wikipedia.org/wiki/PuTTY>

[36] <https://www.realvnc.com/> , consulté le : 10/04/2023.

[37] W. Zhu, N. Zeng, N. Wang, et al., "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations", NESUG proceedings: health care and life sciences, Baltimore, Maryland, Vol. 19, pp. 67. 2010.

[38] S. Maldonado-Bascon, C. Iglesias-Iglesias, P. Martin-Martin, S. Lafuente-Arroyo, "Fallen people detection capabilities using assistive robot", Electronics, Vol. 8 (N°9), pp. 915. 2019.

[39] A. PAJANKAR. "Raspberry Pi computer vision programming". Birmingham :Packt Publishing, pp. 1-25, 2015.

[40] <https://opencv.org> , consulté le : 03/05/2023.

Résumé

Ce projet met en évidence notre succès dans l'utilisation du Deep Learning pour la détection en temps réel des chutes humaines. En utilisant l'algorithme VGG-16 et en entraînant le modèle avec le jeu de données FPDS, nous avons développé un système hautement performant. Ce système utilise également des techniques avancées telles que la soustraction de l'arrière-plan et la détection de pose humaine avec l'algorithme YOLOv5 et OpenPifPaf pour améliorer la précision de la détection. Nous avons implémenté ce système sur le Raspberry Pi 4, offrant ainsi une solution compacte et portable pour la surveillance des chutes dans divers environnements. Les résultats obtenus ont démontré une précision élevée dans la détection des chutes, ouvrant ainsi la voie à des applications potentielles dans les domaines de la sécurité, de la santé et du bien-être des individus.

Mots clés : Détection de chute, L'apprentissage en profondeur, Raspberry Pi, Traitement d'image, OpenCV, ...

ملخص

يسلط هذا المشروع الضوء على نجاحنا في استخدام التعلم العميق للكشف في الوقت الحقيقي عن سقوط الإنسان. باستخدام خوارزمية VGG-16 وتدريب النموذج باستخدام مجموعة بيانات FPDS ، قمنا بتطوير نظام عالي الأداء. يستخدم هذا النظام أيضًا تقنيات متقدمة مثل الطرح في الخلفية واكتشاف الوضع البشري باستخدام خوارزمية YOLOv5 و OpenPifPaf لتحسين دقة الكشف. لقد قمنا بتنفيذ هذا النظام على Raspberry Pi 4 ، مما يوفر حلاً مضغوطًا ومحمولاً لرصد السقوط في بيئات مختلفة. أظهرت النتائج التي تم الحصول عليها دقة عالية في الكشف عن السقوط ، مما يمهد الطريق للتطبيقات المحتملة في مجالات سلامة وصحة ورفاهية الأفراد.

كلمات مفتاحية : كشف السقوط ، تعلم عميق ، Raspberry Pi ، معالجة الصور ، OpenCV ، ...

Abstract

This project highlights our success in using Deep Learning for real-time detection of human falls. By using the VGG-16 algorithm and training the model with the FPDS dataset, we have developed a high performance system. This system also uses advanced techniques such as background subtraction and human pose detection with YOLOv5 and OpenPifPaf algorithm to improve detection accuracy. We have implemented this system on the Raspberry Pi 4, providing a compact and portable solution for monitoring drops in various environments. The results obtained demonstrated high accuracy in the detection of falls, thus paving the way for potential applications in the fields of safety, health and well-being of individuals.

Key words : Fall detection, Deep Learning, Raspberry Pi, Image processing, OpenCV, ...