

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE M'HAMED BOUGARA-BOUMERDES



**Faculté de Technologie**  
**Département Ingénierie des Systèmes Electriques**  
**Mémoire de Master**

Présenté par

**AMITOUCHE Hayet    GHARLEFA Manel**

**Filière : Electronique**

**Spécialité : Electronique des Systèmes Embarqués**

---

**Détection d'objets et estimation de la distance avec  
une Pi Camera en utilisant une Raspberry Pi4**

---

**Soutenu le 01 /07 /2024 devant le jury composé de :**

DAFEUR	Abdelhakim	MA	UMBB	Président
TRIKI	Yacine	MA	UMBB	Examineur
GANNA	Massine	MA	UMBB	Rapporteur

Année Universitaire : 2023/2024

# Remerciements

Au début de ce mémoire, nous tenons à exprimer notre gratitude envers Dieu le Tout-Puissant pour nous avoir accordé la force, la patience, la sagesse et le courage nécessaires afin de mener à bien ce modeste travail.

Nous manifestons nos remerciements à notre encadrant, Monsieur *GANNA Massine*, pour nous avoir donné l'occasion de réaliser ce travail et pour tous les efforts fournis.

Nous tenons à exprimer notre profonde gratitude à notre chef de département d'Ingénierie des Systèmes Électriques, Monsieur *MESSAOUDI Noureddine*, ainsi qu'à nos enseignants qui ont contribué à notre formation.

Nos remerciements vont également aux membres du jury pour l'honneur qu'ils nous font en acceptant de juger notre travail

Enfin, nos remerciements vont à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail

# Dédicace

Grâce à Allah tout puissant, je dédie ce modeste travail

À mes chers parents  
Qui ont toujours été à mes côtés

À mon oncle Mourad  
Qui a été mon plus grand soutien tout au long de mon parcours académique

À mes grands-parents

À mes sœurs Rania et Farah

À mes frères Imad et Abderraouf

À mes tantes et oncles

À mon meilleur binôme d'études Rayane  
Qui a toujours été à mes côtés et m'a aidée à accomplir ce travail

À tous mes cousines et cousins

À toutes mes amies spécialement Sarah, Lilia, Khouloud, Manal et Anissa

Je tiens également à remercier mes collègues de la section

À tous ceux que j'aime et ce qui m'aiment

Encore une fois, un grand merci à toute ma famille et mes amis à qui je dédie ce présent travail.

**Amitouche Hayet**

# Dédicace

Je dédie fièrement mon diplôme à

**Mon père...** Au mur sur lequel je m'appuie dans mes expressions et mes peines. À l'épaule sur laquelle je pose mes fardeaux et à la main qui me caresse à tout moment..... À mon cher et tendre amant que j'aime et apprécie ce monde et plus encore, à la patrie à laquelle j'appartiens et à la terre qui m'embrasse .... À ceux qui me sont les plus chers et les plus proches.

**Ma mère...** mon inspiration, mon soutien, ma confidente, ma force, mon espoir, ma positivité. Pour toutes ses prières estes sacrifices consentis tout au long de mes études, pour apaiser mes craintes dans les moments de doute. Reçois à travers ce modeste travail l'expression de mes sentiments et de mon éternelle gratitude pour me donner le sourire dans les moments de peine.

Que DIEU leurs procure bonne santé et longue vie.

**Mon fiancé.** A vous qui avez cru en moi et m'avez soutenu à chaque étape du chemin... Je dédie cette humble réussite.

**Ma sœur « Loubna ».** Merci beaucoup pour votre soutien et votre soutien à mes cotes pendant toutes les étapes de mes études et autres étapes.

À **mon frère et mes chères amies** ou plutôt mes sœurs **Ahlem, Imane, Meriem, Randa, Aya, Oumaima**, et ses familles trouvent ici l'expression d'un grand « merci » Pour leur amour, leur soutien et leurs encouragements en toutes circonstances.

**Gharlefa Manel**

## Résumé

Chaque jour, nous utilisons nos yeux pour voir le monde qui nous entoure. Le regard est parmi les cinq sens essentiels chez l'être humain, l'organe qui prend en charge cette activité est l'œil. De nombreux troubles peuvent affecter notre vision : myopie, hypermétropie, astigmatisme, presbytie etc. Qui entraînent une baisse de la vue et à longue durée peut aboutir à la cécité. Ce projet a pour objet de développer un système embarqué permet de faciliter la vie aux personnes ayant une déficience visuelle par l'utilisation d'une caméra. L'acquisition des signaux et des images sont faites par la carte Raspberry pi4.

De telles lunettes peuvent être utilisées par les personnes aveugles ou fortement malvoyantes qui ont une grande difficulté à reconnaître les objets et à se déplacer. Cette technologie existe bien qu'elle soit coûteuse et généralement pas disponible pour les gens dans les pays en développement.

**MOTS CLES :** Cécité, Lunettes intelligentes, Déficience visuelle, Caméra.

## Abstract

Every day, we use our eyes to see the world around us. Sight is among the five essential senses in humans, and the organ responsible for this activity is the eye. Many disorders can affect our vision: myopia, hyperopia, astigmatism, presbyopia, etc. which lead to a decrease in vision and, in the long term, can result in blindness. This project aims to develop smart glasses to facilitate the lives of people with visual impairments through the use of a camera. The acquisition of signals and images is done by the Raspberry Pi 4 board.

Such glasses can be used by blind or severely visually impaired people who have great difficulty recognizing objects and moving around. This technology exists, although it is expensive and generally not available to people in developing countries.

**KEYWORDS:** Blindness, Smart glasses, Visual impairment, Camera.

# Table des matières

<b>Remerciements</b> .....	<b>ii</b>
<b>Dédicace</b> .....	<b>iii</b>
<b>Dédicace</b> .....	<b>iv</b>
<b>Résumé</b> .....	<b>v</b>
<b>Abstract</b> .....	<b>v</b>
<b>Table des matières</b> .....	<b>vi</b>
<b>Tables des figures</b> .....	<b>ix</b>
<b>Liste des tableaux</b> .....	<b>xi</b>
<b>Liste des Abréviations</b> .....	<b>xii</b>
<b>Introduction générale</b> .....	<b>1</b>
Chapitre I : Introduction à l'Anatomie de l'Œil et aux Bases de l'Image Numérique .....	<b>3</b>
Introduction.....	<b>3</b>
I.1 L'œil .....	<b>3</b>
I.1.1 Anatomie de l'œil .....	<b>3</b>
I.1.2 La structure de l'œil.....	<b>4</b>
I.1.3 Fonctionnement de l'œil.....	<b>5</b>
I.1.4 Déficience et handicap visuelle .....	<b>6</b>
I.1.5 Types de déficience .....	<b>6</b>
I.1.6 Conséquences de déficient dans la navigation .....	<b>7</b>
I.1.7 Solutions pour Améliorer l'Accessibilité et l'Autonomie .....	<b>8</b>
I.2 Le traitement des images .....	<b>10</b>
I.2.1 L'image numérique.....	<b>10</b>
I.2.2 Caractéristique d'une image numérique .....	<b>11</b>
I.2.3 Le traitement d'images .....	<b>13</b>
Conclusion .....	<b>16</b>

Chapitre II. Outils d'aide à la Vision Artificielle .....	17
Introduction.....	17
II.1 Apprentissage automatique .....	17
II.1.1 En ce qui concerne l'intelligence artificielle .....	18
II.1.2 Méthode d'apprentissage.....	18
II.1.3 Apprentissage supervisé.....	19
II.1.4 Apprentissage non-supervisé .....	19
II.1.5 Apprentissage par renforcement.....	21
II.1.6 Processus d'Apprentissage Automatique .....	21
II.2 L'apprentissage en profondeur (Deep Learning) .....	21
II.2.1 Applications du Deep Learning.....	22
II.2.2 Des differences: Machine learning contre Deep Learning.....	22
II.3 Vision par Ordinateur.....	23
II.3.1 Utilisation de la vision par ordinateur .....	24
II.3.2 La détection d'objets .....	24
II.3.3 La détection et le suivi d'objets.....	25
II.3.4 Outils utilisés en computer vision .....	25
II.3.5 Algorithmes de Vision par Ordinateur .....	26
II.3.6 Algorithmes de détection d'objet .....	26
II.3.7 Algorithmes de segmentation d'images .....	27
II.3.8 Algorithmes de classification d'images .....	28
II.3.9 Algorithmes de reconnaissance .....	29
II.3.10 Algorithmes de Détection de Mouvement .....	29
II.3.11 Algorithmes de Reconnaissance de Mouvements Humains .....	29
II.4 Réseaux de neurones profonds.....	30
II.4.1 Importance des réseaux de neurones profonds.....	30
II.4.2 Réseaux de neurones artificiels .....	31

II.4.3 Fonctionnement de réseaux de neurones.....	31
II.4.4 Types de réseaux neurone .....	32
II.5 Les modèles de détection par CNN.....	37
II.6 YOLO.....	38
II.6.1 Avantages .....	40
Conclusion .....	40
Chapitre III. Détection d’objet en utilisant un pi camera .....	41
Introduction.....	41
III.1 Matériels et ressources utilisés .....	41
III.1.1 PC Portable .....	41
III.1.2 Raspberry Pi 4.....	44
III.1.3 Camera Raspberry 5mp RV1.3 .....	49
III.1.4 Carte mémoire.....	51
III.2 Organigramme du code.....	53
III.3 Explication détaillée du code pour la détection d'objets.....	54
III.4 Tests et validation .....	60
Conclusion .....	64
<b>Conclusion Général.....</b>	<b>65</b>

# Tables des figures

Figure I- 1 : L'anatomie de l'œil.....	3
Figure I- 2: Fonctionnement de l'œil.....	5
Figure I- 3: Les aides traditionnelles à la mobilité .....	8
Figure I- 4: Les aides Électroniques [6].....	10
Figure I- 5: Echantillonnage d'une image .....	12
Figure I- 6 : Quantification d'une image .....	13
Figure I- 7: Processus de traitement d'image .....	13
Figure I- 8 : Quelques modèles de contours .....	15
Figure II- 1: Quelques algorithmes des 3 types d'apprentissage du Machine Learning :supervisé ou non supervisé et renforcement [18] .....	19
Figure II- 2: Supervised learning .....	19
Figure II- 3: Unsupervised learning .....	20
Figure II- 4 : Le cycle de vie d'un projet de Machine Learning en 8 étapes.....	21
Figure II- 5: La relation entre l'intelligence artificielle, le ML et le Deep Learning .....	22
Figure II- 6: La différence entre machine Learning et Deep Learning [23] .....	22
Figure II- 7 : Réseau neuronal simple et réseau de neurones profonds [28].....	30
Figure II- 8 : Définition d'un réseau de neurones [32] .....	31
Figure II- 9 : RNN.....	33
Figure II- 10 : Architecture de Réseaux Adverbiaux Génératifs .....	34
Figure II- 11 : Architecture d'un réseau de neurone convolutif [36].....	35
Figure II- 12 : La couche pooling .....	36
Figure II- 13 : Couche entièrement connecté (Fully Connected) [39].....	36
Figure II- 14 : Schéma des modèles de détection et classification par CNN.....	37
Figure II- 15 : Étapes importantes de la détection et de la reconnaissance des objets .....	38
Figure II- 16 : Diviser l'image en (G×G) grille .....	39
Figure II- 17 : Le résultat de Suppression non maximale [43] .....	40
Figure III - 2 : Schéma du montage de circuit de commande.....	46
Figure III - 3 : Branchement des câbles avec la Raspberry .....	47

Figure III - 4 : Alimentation d'une carte raspberry avec le PC .....	49
Figure III - 5 : Caméra Raspberry [45] .....	50
Figure III - 6 : Connecter la caméra avec la carte raspberry.....	50
Figure III - 7 : Carte mémoire SD.....	52

# Liste des tableaux

Tableau I- 1: Classifications des déficiences visuelles selon l'OMS [6] .....	7
Tableau II- 1 : La différence entre machine Learning et intelligence artificielle.....	18
Tableau II - 2 : La différence entre apprentissage supervisé et non supervisé .....	20
Tableau II - 3 : La différence entre Machine Learning et Deep Learning .....	23

# Liste des Abréviations

**CNN** : Les réseaux de neurones convolutifs.

**CT** : Computed Tomography.

**DMLA** : Caractérisée par la prolifération de nouveaux vaisseaux dysfonctionnels sous la rétine.

**DNN** : Réseaux de neurones profonds.

**HOG**: Histogram of Oriented Gradients.

**IA** : L'Intelligence Artificielle.

**IRM** : Résonance magnétique nucléaire.

**GAN** : Réseaux Adversarial Génératifs.

**GPS** : Global positioning system.

**ML** : Machine Learning.

**N<sub>T</sub>e** : Période d'échantillonnage.

**OMS** : L'Organisation mondiale de la santé.

**RNN** : Réseaux de neurones récurrents.

**RNP** : Réseau Neuronal Profond.

**SIFT**: La scale-invariant feature transform.

**SVM**: Support Vector Machine.

**VPU** : Traitement de la vision.

# ***INTRODUCTION GENERALE***

# Introduction générale

Selon l'Organisation Mondiale de la Santé (OMS), il existe près de 285 millions de personnes qui présentent une déficience visuelle, dont 45 millions sont malvoyants et 180 millions souffrent de handicap visuel. 90 % des personnes qui présentent une déficience visuelle vivent dans un pays à faible revenu, 82 % des malvoyants sont âgés de 50 ans et plus. Les études démontrent que l'immobilité des malvoyants est une conséquence directe de leur handicap visuel. Pour remédier à ce problème, l'appréhension de l'espace par un champ perceptif autre que la vision s'avère nécessaire [1].

Les avancées technologiques ont permis le développement de solutions innovantes pour améliorer la qualité de vie des personnes en situation de handicap visuel. La capacité de se déplacer de manière autonome et en toute sécurité dans des environnements divers constitue un défi majeur pour les malvoyants.

En effet, éviter les obstacles et s'orienter dans l'espace sont des compétences cruciales mais souvent difficiles à maîtriser sans assistance. L'un des moyens prometteurs pour répondre à ces besoins est l'utilisation de dispositifs technologiques avancés, tels que les lunettes intelligentes, qui fournissent des informations visuelles sous forme de feedback sonore ou en braille.

Cependant, pour que ces dispositifs soient réellement efficaces, ils doivent être capables de détecter précisément les objets environnants et d'estimer la distance à laquelle ils se trouvent. C'est dans ce contexte que s'inscrit notre projet de fin d'études, qui vise à concevoir et réaliser un système de détection d'objets et d'estimation de distance utilisant une Pi Camera associée à une carte Raspberry Pi4.

Ce mémoire est structuré en trois chapitres principaux :

**Chapitre I** pour traiter certaines généralités telles que de l'anatomie et la physiologie de l'œil en citant les constitutions de l'œil, quelques pathologies de l'œil et les difficultés rencontrées par les malvoyants et les personnes ayant une déficience visuelle dans la vie quotidienne.

**Chapitre II** nous allons citer une partie de l'état de l'art sur la recherche dans les Outils d'aide à la Vision Artificielle

**Chapitre III** est spécifié pour présente la partie software (la programmation) et l'interprétation des résultats.

En conclusion, le développement de ce système de détection d'objets avec détermination de distance via une Raspberry Pi4 et une Pi Camera, avec une sortie sous forme de message vocal, représente une avancée significative pour l'assistance aux personnes malvoyantes. Notre projet démontre la faisabilité et l'efficacité de cette solution, ouvrant la voie à des dispositifs plus performants et accessibles, contribuant ainsi à une meilleure autonomie et qualité de vie pour les personnes en situation de handicap visuel .

# ***CHAPITRE.I***

***Introduction à l'Anatomie de l'Œil et aux Bases de  
l'Image Numérique***

## Chapitre I : Introduction à l'Anatomie de l'Œil et aux Bases de l'Image Numérique

### Introduction

« L'œil est une merveille de la nature, un des organes les plus perfectionnés de notre corps. Miroir de nos émotions et de nos pensées secrètes ». Lorsque les causes de malvoyance ou de cécité ne sont pas traitables et que le déficit restera permanent, différentes solutions ont été proposées pour compenser la perte visuelle et restaurer, si ce n'est un réel sens visuel, au moins certaines des fonctions perdues. Dans ce chapitre, nous présentons quelques notions d'anatomie du système visuel humain, ainsi que quelques outils d'aide, et un aperçu général sur les images et leur traitement.

### I.1 L'œil

#### I.1.1 Anatomie de l'œil

L'œil est l'organe principal du système visuel, comparable à un appareil photo autofocus, permettant une vision nette à toutes les distances. Il capte les images, les transforme en signal électrique vers le nerf optique, qui transmet alors cette information au cerveau. Le cerveau interprète l'image et la restitue. Le champ visuel humain couvre environ  $220^\circ$  en horizontal et  $140^\circ$  en vertical. Un œil qui voit « bien » est un œil qui restitue exactement et sans effort l'image regardée sur la rétine [1].

La vue chez l'être humain représente la principale source d'information du cerveau sur le monde extérieur : environ 70% des informations passent par la vision [2].

L'œil est composé de quatorze éléments, comme l'illustre la figure I- 1 ci-dessous.

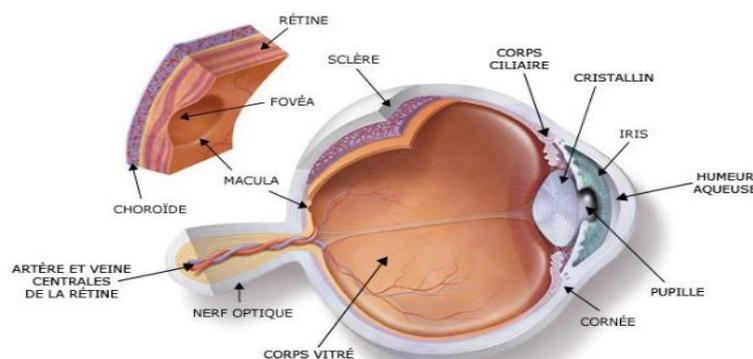


Figure I- 1 : L'anatomie de l'œil

## I.1.2 La structure de l'œil

Les structures et les fonctions de l'appareil visuel sont complexes. Constamment, chaque œil ajuste la quantité de lumière qu'il laisse passer, effectue la mise au point sur les objets proches ou éloignés, et forme des images en continu qui sont instantanément transmises au cerveau [3]. Dans cet organe ultra-perfectionné, chaque élément a son rôle et son importance on les site ci-dessus [2] :

### I.1.2.1 La paroi oculaire

C'est l'enveloppe externe de l'œil, composée de trois tuniques :

- **La sclérotique** : C'est la partie blanche de l'œil, qui offre protection et structure.
- **La choroïde** : Elle se situe sous la sclérotique et contient des vaisseaux sanguins qui nourrissent l'œil.
- **La rétine** : La rétine est une membrane épaisse constituée de centaines de millions de cellules nerveuses photoréceptrices qui transmettent l'information visuelle vers le cerveau :
  - Les cônes, qui permettent de percevoir la lumière du jour et les couleurs
  - Les bâtonnets qui nous donnent une vision des formes dans l'obscurité.

### I.1.2.2 Les milieux transparents

Ils remplissent l'intérieur de l'œil et permettent à la lumière de traverser et de se focaliser sur la rétine :

- **Le cristallin** : Il est la lentille intra-oculaire qui permet la mise au point (accommodation). Il fonctionne comme « l'autofocus » d'un appareil photo, assurant la netteté de loin comme de près. Parfaitement transparent, le cristallin est situé derrière la pupille et devant la rétine mais ne peut pas être vu à l'œil nu[4] .
- **La cornée** : Elle agit comme une fenêtre par laquelle la lumière pénètre dans l'œil. Elle a un rôle de focalisation des images sur la rétine et doit être transparente. C'est sur la cornée que le chirurgien ophtalmologue va agir grâce au laser femtoseconde.
- **L'humeur aqueuse** : C'est un liquide transparent, filtré et renouvelé en permanence, est l'humeur aqueuse, qui, avec le corps vitré, maintient la pression et la forme du globe oculaire. Principalement constituée d'eau, mais également de vitamine C, de glucose, d'acide lactique et de protéines, elle se renouvelle en environ 2 à 3 heures.

### I.1.2.3 Autres parties importantes

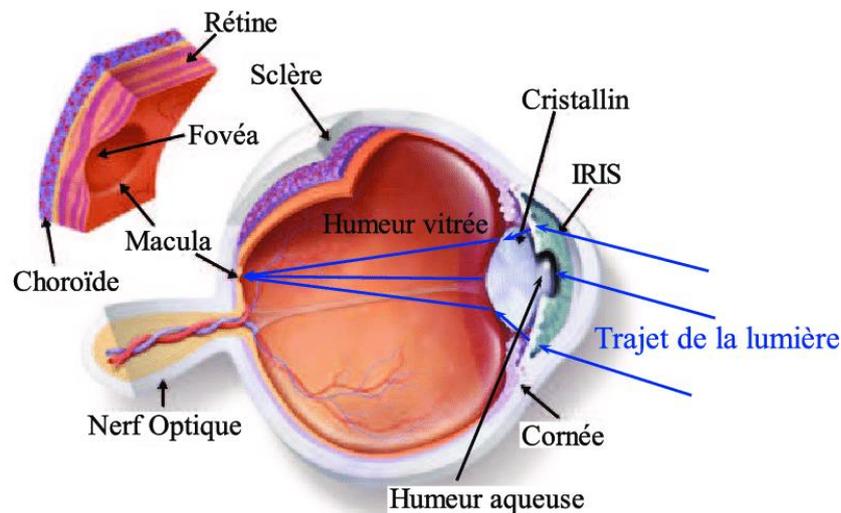
- **L'iris** : c'est une membrane pigmentée circulaire, perforée en son centre pour former la pupille. L'iris donne leur couleur aux yeux (en fonction de la pigmentation plus ou moins importante de la membrane).

- **La pupille :** c'est une structure constituée de l'espace libre au centre de l'iris. Ce dernier comprend deux groupes de muscles : L'un composé de fibres radiales (disposées comme les rayons d'une roue) qui élargit la pupille, l'autre comportant des fibres circulaires, qui la rétrécit. Leur action modifie le diamètre de la pupille et régule la quantité de lumière entrant dans l'œil, tout comme le diaphragme d'un appareil photo détermine le diamètre d'ouverture de l'objectif.

**Le nerf optique :** c'est un nerf reliant chaque globe oculaire, plus précisément au niveau de la rétine, jusqu'au cerveau, il se charge de transférer les informations reçues afin qu'elles soient traitées par le cerveau [5]. Il mesure 4 mm de diamètre pour 5 cm de long. C'est lui qui permet au cerveau d'enregistrer, d'interpréter et de traduire les images.

### I.1.3 Fonctionnement de l'œil

Lorsque la lumière pénètre dans l'œil, elle traverse successivement la cornée, l'humeur aqueuse, le cristallin et l'humeur vitrée avant d'atteindre la rétine (voir figure I.2). La quantité de lumière entrant est régulée par l'ouverture de la pupille, qui dépend de la contraction (myosis) ou du relâchement (mydriase) de l'iris. L'iris agit ainsi comme un diaphragme circulaire, contrôlant l'entrée de la lumière. La zone de l'espace réel visible par un œil fixe est appelée champ visuel. Lorsqu'un œil est en bonne santé, le champ visuel normal est de 60° vers le nez, 100° vers la tempe, 65° vers le haut et 75° vers le bas.



**Figure I- 2:** Fonctionnement de l'œil

### **I.1.4 Déficience et handicap visuelle**

La déficience visuelle se caractérise par une altération ou une absence de perception visuelle par l'œil, résultant d'une atteinte au niveau de l'œil lui-même ou des voies visuelles jusqu'au système cérébral. Ces atteintes peuvent être présentes dès la naissance ou survenir ultérieurement suite à des accidents ou des maladies telles que le diabète, la DMLA ou le glaucome. Le traitement de ces déficiences visuelles relève de la prise en charge médicale, qui peut inclure l'utilisation de médicaments, la chirurgie ou des dispositifs optiques correctifs. La déficience peut affecter à la fois l'acuité visuelle, mesurée en pourcentage par rapport à une vision normale, et/ou le champ visuel, que ce soit d'un œil ou des deux.

Le handicap visuel découle de cette déficience ; il traduit les perturbations que celle-ci entraîne dans les activités quotidiennes. La réduction ou la perte de la vue impacte nos capacités dans les domaines personnels, familiaux et sociaux.

### **I.1.5 Types de déficience**

Dans la classification internationale des maladies et la classification internationale du fonctionnement, du handicap et de la santé, l'Organisation mondiale de la santé (OMS) a établi cinq catégories de déficiences qui prennent en compte à la fois la diminution de l'acuité visuelle et la réduction du champ visuel. Les catégories 1 et 2 sont celles de la malvoyance (ou basse vision ou vision réduite), celles de 3 à 5 sont celles de la cécité. Les évaluations sont toujours basées sur une diminution de l'acuité visuelle ou sur une réduction du champ visuel.

Le tableau I- 1 présente les critères sélectionnés par la récente révision de la classification internationale des maladies de l'OMS. Malgré le fait que ces définitions soient maintenant considérées comme la référence mondiale dans l'épidémiologie des déficiences visuelles, la majorité des États ont néanmoins maintenu leur propre définition du handicap visuel. En Algérie, lorsque le taux d'incapacité est attribué, la cécité totale se réfère à une vision complètement abolie, sans aucune perception de la lumière. Selon les données actuelles, elle correspond à la cécité totale classée par l'OMS (catégorie 5).

Catégorie OMS	Acuité visuelles et champ de vision	Type d'atteint visuelle	Atteinte visuelle Type de déficience visuelle
<b>Catégorie 1</b>	1/10 <AVC<3/10 Champ visuel d'au moins 20 °	Basse vision	Moyenne
<b>Catégorie 2</b>	1/20 <AVC<1/10	Basse vision	Sévère
<b>Catégorie 3</b>	1/50 <AVC<1/20 5°< champ visuel<10°	Cécité	Profonde
<b>Catégorie 4</b>	AVC<1/50 mais perception lumineuse préservée champ visuel <5°	Cécité	Presque totale
<b>Catégorie 5</b>	Cécité absolue, absence de perception lumineuse	Cécité	Totale

**Tableau I- 1:** Classifications des déficiences visuelles selon l'OMS [6]

### I.1.6 Conséquences de déficient dans la navigation

La navigation est une activité quotidienne cruciale pour les individus, leur permettant de se déplacer efficacement dans leur environnement. Cependant, pour les personnes atteintes de déficience visuelle, cette tâche peut s'avérer extrêmement difficile, voire impossible, sans les adaptations appropriées. Dans ce mémoire, nous explorons les déficiences visuelles et leurs conséquences sur la navigation, en mettant en lumière les défis auxquels sont confrontées ces personnes et les solutions innovantes visant à améliorer leur accessibilité et leur autonomie.

Nous identifions les principaux défis auxquels sont confrontées les personnes malvoyantes et aveugles lorsqu'elles tentent de se déplacer, notamment :

- **Reconnaissance de l'environnement** : La capacité à identifier les obstacles, les intersections et les repères est essentielle pour une navigation sûre.
- **Orientation spatiale** : La difficulté à percevoir les directions et les distances peut entraîner une désorientation fréquente.
- **Accès à l'information** : Les informations visuelles telles que les panneaux indicateurs ou les feux de signalisation ne sont pas accessibles aux personnes aveugles ou

malvoyantes.

- **Communication avec l'environnement** : L'interaction avec les autres piétons et les conducteurs peut être entravée en raison du manque de communication non verbale.

### I.1.7 Solutions pour Améliorer l'Accessibilité et l'Autonomie

Pour répondre aux défis auxquels sont confrontées les personnes malvoyantes et aveugles, nous examinons diverses solutions technologiques et sociales visant à améliorer leur accessibilité et leur autonomie dans la navigation quotidienne. Parmi ces solutions, on trouve des approches classiques et modernes.

#### I.1.7.1 Solutions Classiques

Les solutions classiques incluent des méthodes éprouvées et largement adoptées pour aider les personnes malvoyantes et aveugles. Parmi celles-ci :

- **Chiens guides** : Les chiens guides ont longtemps été utilisés pour aider les personnes aveugles à se déplacer en toute sécurité. Ils sont formés pour éviter les obstacles, trouver des portes et des escaliers, et fournir un accompagnement fiable.
- **Canne blanche** : La canne blanche est un outil indispensable pour les personnes malvoyantes, leur permettant de détecter les obstacles et de s'orienter dans leur environnement immédiat.
- **Braille** : Le braille est un système de lecture et d'écriture tactile utilisé par les personnes aveugles. Il permet l'accès à l'information écrite, comme les livres, les étiquettes et les panneaux de signalisation.



Figure I- 3: Les aides traditionnelles à la mobilité

#### I.1.7.2 Solutions Modernes

Les solutions modernes tirent parti des avancées technologiques pour offrir des niveaux d'assistance et d'autonomie plus élevés. Ces solutions incluent :

- **Applications mobiles** : De nombreuses applications mobiles sont conçues spécifiquement pour aider les personnes malvoyantes. Par exemple, des applications comme Seeing AI et Be My Eyes utilisent la caméra du smartphone pour décrire les objets et les environnements, lire des textes, et même identifier les couleurs et les visages.
- **Dispositifs portables** : Des dispositifs portables tels que des lunettes intelligentes et des cannes électroniques sont de plus en plus disponibles. Ces dispositifs intègrent des technologies avancées comme les capteurs à ultrasons et les caméras pour détecter les obstacles et fournir des retours haptiques ou auditifs à l'utilisateur.
- **Aides à la navigation** : Utilisant des technologies telles que le GPS et la réalité augmentée, ces aides fournissent des instructions de navigation précises et en temps réel. Par exemple, des systèmes comme Aira et BlindSquare offrent une assistance vocale pour guider les utilisateurs dans des environnements complexes et inconnus.
- **Aménagements Urbains** : Pour créer des environnements plus accessibles, les aménagements urbains doivent intégrer des principes de conception inclusive. Cela inclut :
  - **Signalisations tactiles** : L'utilisation de signalisations tactiles, telles que les bandes de guidage au sol et les plaques en braille sur les panneaux, aide les personnes malvoyantes à se repérer dans les espaces publics.
  - **Trottoirs sûrs** : Des trottoirs bien conçus, avec des bordures abaissées et des revêtements antidérapants, augmentent la sécurité et l'accessibilité pour les personnes utilisant des aides à la mobilité comme les cannes ou les chiens guides.
  - **Passages piétons accessibles** : Les passages piétons doivent être équipés de feux de signalisation sonores et de bandes podotactiles pour guider les personnes malvoyantes de manière sûre et efficace à travers les rues.



**Figure I- 4:** Les aides Électroniques [6]

Ces solutions, qu'elles soient classiques ou modernes, lorsqu'elles sont mises en œuvre de manière intégrée, peuvent considérablement améliorer l'accessibilité et l'autonomie des personnes malvoyantes et aveugles, leur permettant de mener une vie plus indépendante et sécurisée.

## I.2 Le traitement des images

### I.2.1 L'image numérique

L'image numérique peut être vue comme une matrice « Tableau rectangulaire », chaque élément de la matrice étant appelé pixel, abréviation de « picture element » [7].

Elle est une image divisée en éléments de tailles fixes appelés cellules ou pixels. Chaque pixel représente un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle ou calculé à partir d'une description interne de la scène à représenter. Pour numériser une image, celle-ci est convertie de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques  $f(x,y)$ , où  $x$  et  $y$  sont les coordonnées cartésiennes d'un point de l'image et  $f(x,y)$  est le niveau de gris en ce point. Les données d'image sont généralement rangées sous forme de tableau  $I$  de  $n$  lignes et  $p$  colonnes pour faciliter leur représentation pour l'affichage et l'adressage. Chaque élément  $I(x,y)$  du tableau représente un pixel de l'image et a une valeur de niveau de gris codée sur  $m$  bits ( $2^m$  niveaux

de gris, 0 étant noir et  $2^m-1$  étant blanc). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur [8]. Il existe quatre types d'image :

- Images binaires.
- Images d'intensités.
- Image couleur RGB.
- Image couleur indexée.

## **I.2.2 Caractéristique d'une image numérique**

Elle est caractérisée par les paramètres suivants :

### **I.2.2.1 Pixel**

Le pixel constitue le plus petit élément de l'image. Ainsi, une image numérique est représentée par une matrice de dimension  $(m \times n)$  dont les éléments sont appelés pixels « picture element ». La valeur de chaque pixel représente une couleur considérée dans l'image. Les pixels sont répartis régulièrement en lignes et en colonnes [9] [10].

### **I.2.2.2 Niveau de gris**

L'intensité lumineuse en un point spécifique d'une image est représentée par le niveau de gris. La teinte d'un pixel peut être noire ou blanche, en passant par plusieurs niveaux intermédiaires. Dans le cas d'une image en niveaux de gris, chaque pixel est donné une valeur qui correspond à la quantité de lumière qu'il reflète, généralement comprise entre 0 et 255. Il s'agit donc d'un octet entier pour chaque pixel et non d'un seul bit. Ainsi, le matériel employé pour représenter l'image doit être capable de reproduire ces diverses nuances de gris.

Le nombre de niveaux de gris possibles est proportionnel au nombre de bits utilisés pour coder la "couleur" de chaque pixel : plus il y a de bits, plus le nombre de niveaux de gris possibles est élevé.

### **I.2.2.3 Bruit**

La variation soudaine d'un pixel par rapport à ses voisins est un phénomène qu'on appelle bruit. Ce dernier peut provenir de l'éclairage, des dispositifs électroniques du capteur, voir même de l'image [11].

### **I.2.2.4 Contour et texture**

Les contours constituent une information essentielle pour certaines applications de traitement d'image. En particulier, les contours d'un objet permettent en générale de caractériser sa forme. La détection de contours peut être réalisée grâce à des filtres dont les coefficients ont été soigneusement choisis[12].

### I.2.2.5 Histogramme

L'histogramme peut être considéré comme une densité de probabilité des niveaux de gris de l'image. Pour le calcul d'un histogramme, on donne un nombre de niveaux de quantification et, pour chaque niveau, on compte le nombre de pixels de l'image correspondant à ce niveau [12].

L'histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris de l'image, ou ce qu'on appelle « La dynamique de l'image ». Il permet de voir entre quelles bornes est répartie la majorité des niveaux de gris dans le cas d'une image trop claire ou trop foncée [9].

### I.2.2.6 Obtention des données images

Afin de permettre l'utilisation de l'image par l'ordinateur, il est nécessaire de la numériser. Il s'agit de convertir l'image de base en une mosaïque de points, puis de convertir ces points en valeurs binaires que le système informatique peut traiter. La numérisation d'une image est la transformation d'une image de son état analogique (signal continu) vers l'état numérique. La procédure se déroule en trois étapes :

- **Echantillonnage**

L'échantillonnage d'un signal, qu'il soit unidimensionnel ou bidimensionnel (comme dans le cas d'une image), consiste à prélever des échantillons de ce signal à des intervalles de temps réguliers, spécifiés par la période d'échantillonnage (N<sub>T</sub>e). (Figure I- 5).



**Figure I- 5:** Echantillonnage d'une image

- **Quantification**

Elle consiste à assigner une valeur numérique à chaque échantillon. Cette quantification peut être effectuée en nuances de gris (monochrome) ou en couleurs. (Figure I- 6)



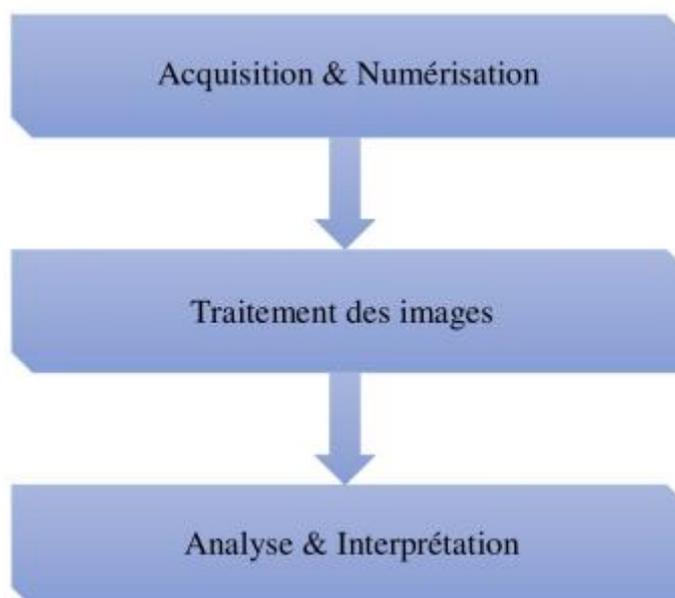
**Figure I- 6 :** Quantification d'une image

- **Codage**

Le codage est une méthode qui implique de fournir à chaque pixel de l'image un code qui reflète une information supplémentaire : sa teinte et sa précision. Cela dans le but d'améliorer son stockage ou sa transmission.

### **I.2.3 Le traitement d'images**

Le traitement d'images est l'ensemble des opérations effectuées sur l'image, afin d'en améliorer la lisibilité et d'en faciliter l'interprétation. L'analyse d'images est une chaîne de traitement de l'information [13] qui peut être décrite par la (Figure I- 7) :



**Figure I- 7:** Processus de traitement d'image

#### **I.2.3.1 Acquisition et numérisation des images**

La collecte d'images revêt une importance capitale lors de toutes les étapes de conception et de production d'images. Pour qu'une image puisse être utilisée sur un système informatique, il est essentiel de la modifier pour qu'elle puisse être lisible et manipulable du système. L'imagerie médicale commence au moment de l'acquisition. Depuis plusieurs années, il y a une

multitude de technologies et d'équipements, pour l'acquisition, il faut attendre plusieurs années, comme pour le scanner (tomographie par rayon X), l'IRM (résonance magnétique nucléaire) ou l'échographie (radiographie). Ces techniques conduisent généralement à une succession d'images en deux dimensions, les coupes étant placées les unes derrière les autres, reconstituant ainsi le volume du patient.

### I.2.3.2 Le traitement

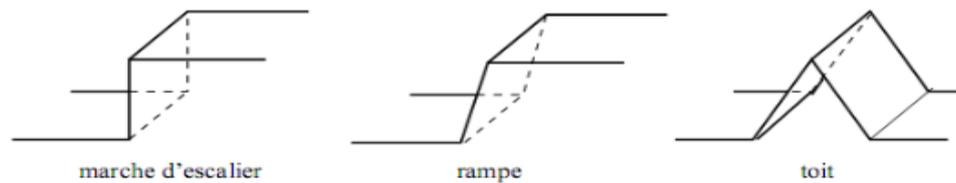
Cette étape intervient immédiatement après l'acquisition des images et vise à améliorer la qualité de l'image afin de la segmenter. Le délai de traitement revêt une grande importance. Il s'agit d'un élément déterminant, il doit être aussi petit que possible. Il est donc nécessaire que les opérateurs soient localisés, c'est-à-dire qu'ils doivent intervenir sur un nombre restreint de pixels et principalement sur les pixels situés à proximité du pixel courant. Les méthodes de prétraitement les plus répandues sont les suivantes :

- **Restauration** : Il s'agit de recréer l'image altérée en supprimant les imperfections causées par une source. L'objectif est de produire une image aussi proche que possible de l'image idéale, qui aurait été obtenue si le système d'acquisition était parfait.
- **L'amélioration** : Il s'agit de la transformation de l'image afin de la rendre plus plaisante à regarder.
- **La compression** : Réduction du volume de l'image.
- **La segmentation** : La segmentation d'une image implique de séparer une image en différents segments ou régions afin de simplifier ou modifier la représentation d'une image en quelque chose de plus important et plus facile à analyser. Elle offre la possibilité de séparer des objets ou des zones d'intérêt d'une image. Les méthodes fréquemment utilisées pour segmenter comprennent :
  - Selon un seuil de luminosité, qui permet de simplifier une image en convertissant les pixels en noir et blanc.
  - La segmentation par région consiste à regrouper des pixels proches présentant des propriétés similaires.
  - Selon les contours des objets, la segmentation par contours permet de les séparer du reste de l'image. Et elle est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermés, formant les frontières significatives de l'image.

Il existe deux (02) grandes approches de la détection des régions :

- Méthodes basées sur la détection des régions : approches régions.
- Méthodes basées sur la détection des contours : approches contours.

Les contours dans une image proviennent des : discontinuités de la fonction de réflectance (texture, ombre), où discontinuités de profondeur (bords de l'objet), dans l'image correspond à une variation brusque de l'intensité lumineuse de l'image. Plusieurs types de contours peuvent être trouvés (Figure I- 8).



**Figure I- 8 :** Quelques modèles de contours

- **Le filtrage :** L'utilisation du filtrage d'image permet d'améliorer la qualité de l'image en éliminant le bruit, en lissant les détails ou en accentuant certains aspects. Différents types de filtres existent, chacun ayant un but précis : les filtres passe-bas permettent de diminuer le bruit et de lisser l'image en atténuant les fréquences élevées, les filtres passe-haut mettent en évidence les détails et les contours en amplifiant les fréquences élevées, et les filtres médians permettent de diminuer le bruit impulsionnel tout en préservant les bords en substituant chaque pixel par la médiane des pixels aux alentours.

### I.2.3.3 Analyse et interprétation

#### I.2.3.3.1 Analyse

L'analyse vise à rendre possible l'utilisation du contenu de l'image à des fins d'interprétation. La phase d'analyse utilise un ou plusieurs types d'informations estimés à partir des pixels pour aider au diagnostic en imagerie satellitaire, médicale, de localisation et de reconnaissance (robotique, vidéosurveillance) ou de mesure des évolutions, en fonction de la nature de pixels.

#### I.2.3.3.2 Interprétation

Cette phase consiste à extraire une ou plusieurs informations d'une image, afin de comprendre son contenu, en associant une valeur symbolique aux régions précédemment définies. Pour cela, l'utilisateur doit disposer des connaissances à priori, qu'il a dû contenir de l'image pour définir les attributs pertinents des objets, permettant d'associer à chacun une valeur symbolique.

## **Conclusion**

Au cours de ce chapitre, nous avons exploré l'anatomie de l'œil et son fonctionnement, ainsi que les défis posés par la déficience visuelle. Nous avons également présenté une variété de Solutions, à la fois classiques et modernes, qui visent à compenser la perte de vision et à améliorer l'autonomie des personnes malvoyantes et aveugles, ainsi que des généralités sur le traitement d'images.

En outre, il est essentiel de souligner le rôle croissant de l'intelligence artificielle (IA) dans le développement de nouvelles solutions pour les personnes atteintes de déficience visuelle. L'IA, avec ses capacités d'apprentissage automatique et de traitement avancé des données, offre des opportunités sans précédent pour créer des outils plus efficaces et personnalisés.

## ***CHAPITRE.II Outils d'aide à la Vision Artificielle***

## Chapitre II. Outils d'aide à la Vision Artificielle

### Introduction

Quand on évoque l'apprentissage automatique et l'intelligence artificielle, on se projette dans un monde rapide et futuriste. Des modèles sophistiqués nous permettent de prendre des décisions intelligentes qui sont davantage adaptées à notre situation. De la même manière, il est possible d'optimiser les caractéristiques des algorithmes d'apprentissage automatique. De nos jours encore, il est possible de créer des modèles capables d'entendre, voir et réagir à l'environnement en utilisant des données d'apprentissage, et de générer de nouvelles données afin d'améliorer l'expérience de l'utilisateur.

Toutefois, il est difficile de concevoir de tels modèles d'apprentissage automatique. Il y aura de nombreux obstacles à surmonter lors de la finalisation du modèle. Cependant, il est possible d'utiliser différents outils et applications pour vous assister dans cette procédure.

Découvrons maintenant ce chapitre captivant sur le domaine fascinant de l'intelligence artificielle appliquée à la navigation, en explorant ses méthodes, modèles, architectures, ainsi que l'outil essentiel qu'est Python pour mettre en œuvre ces progrès.

### II.1 Apprentissage automatique

Fabien Benureau a proposé en 2015 une définition de l'apprentissage qui s'applique autant à un programme informatique qu'à un animal de compagnie ou à un être humain : "L'apprentissage est une modification d'un comportement sur la base d'une expérience" [14]. Lorsqu'il est question d'un programme informatique, nous utilisons le terme d'apprentissage automatique, ou machine Learning, pour désigner sa capacité à apprendre sans que cette modification ne soit explicitement programmée. Cette définition a été donnée pour la première fois par Arthur Samuel en 1959 [15]. L'approche d'un programme classique est différente de celle d'un programme d'apprentissage automatique. Dans un programme traditionnel, on utilise des données d'entrée afin de produire des réponses en sortie en suivant une procédure spécifique. D'autre part, un logiciel d'apprentissage automatique se sert de données et de réponses afin de créer la méthode permettant de générer les réponses à partir des données.

### II.1.1 En ce qui concerne l'intelligence artificielle

On peut considérer la machine Learning (l'apprentissage automatique) comme une branche de l'intelligence artificielle. En effet, un système incapable d'apprendre a du mal à être considéré comme intelligent. Il est crucial pour un système de pouvoir apprendre et tirer des leçons de ses expériences afin de s'adapter à un environnement en constante évolution. L'intelligence artificielle, qui se réfère à toutes les méthodes utilisées pour créer des machines capables d'adopter des comportements qualifiés d'intelligents, implique de multiples disciplines, comme les sciences cognitives, la neurobiologie, la logique, l'électronique, l'ingénierie, et bien d'autres encore. Selon toute probabilité, le terme « intelligence artificielle » est de plus en plus employé à la place de celui de « machine Learning ».

Pour vous aider à mieux comprendre la différence entre machine Learning et intelligence artificielle, voici un tableau récapitulatif [16] :

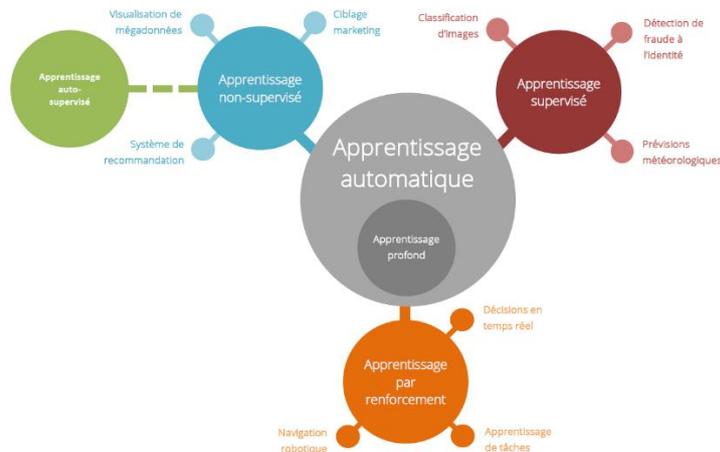
<b>Machine Learning</b>	<b>Intelligence artificielle</b>
L'apprentissage automatique est une technique permettant de développer l'IA.	L'IA est un concept visant à imiter l'intelligence humaine.
Le ML permet aux machines d'apprendre par elles-mêmes (sans raisonnement propre).	L'IA cherche à créer des modèles capables de raisonner par eux-mêmes.
Le ML permet aux machines de résoudre des problématiques spécifiques.	L'IA crée des modèles capables de résoudre des tâches complexes comme un humain.
Le ML n'utilise que des données structurées ou semi-structurées.	L'IA peut utiliser tous types de données.

**Tableau II- 1 :** La différence entre machine Learning et intelligence artificielle

### II.1.2 Méthode d'apprentissage

Afin d'aider une machine à apprendre, il est nécessaire de lui donner cette capacité qui correspond à un ensemble de méthodes d'apprentissage inspirées de la manière dont nous, les êtres humains, apprenons à faire des différentes choses. Le but de l'apprentissage est d'induire une fonction qui prédise les réponses associées à de nouvelles observations en commettant une erreur de prédiction la plus faible possible [17]. Parmi ces méthodes, on note : apprentissage

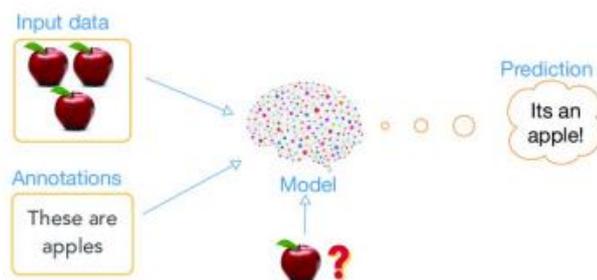
supervisé, non supervisé et par renforcement. Où chaque type repose sur un ensemble d'algorithmes.



**Figure II- 1:** Quelques algorithmes des 3 types d'apprentissage du Machine Learning :supervisé ou non supervisé et renforcement [18]

### II.1.3 Apprentissage supervisé

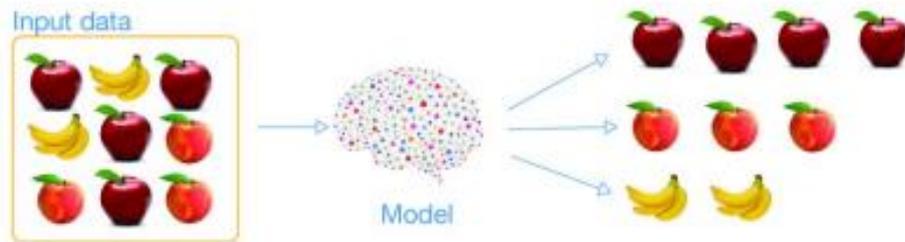
Dans l'apprentissage supervisé, l'agent observe quelques couples types entrée-sortie et apprend une fonction de l'entrée vers la sortie [17] . C'est-à-dire, notre échantillon de données se présente sous la forme d'un couple (X, Y) où X représente tous les aspects qui correspondent aux catégories de la cible Y. Dans la figure II- 2, nous avons donné des pommes au modèle comme données étiquetées. Après l'entraînement, le modèle sera capable de classer de nouvelles images de pommes comme des pommes.



**Figure II- 2:** Supervised learning

### II.1.4 Apprentissage non-supervisé

La notion d'apprentissage non supervisé est utilisée lorsqu'on ne connaît pas les valeurs de sortie et que l'algorithme doit prendre en compte tous les aspects X et reconnaître les structures communes entre ces derniers afin de prédire la cible Y. Dans la figure II- 3, nous avons donné des données non étiquetées au modèle. Après l'entraînement, le modèle a pu distinguer les 3 classes, et a également créé des classes en fonction des similitudes entre les échantillons.



**Figure II- 3:** Unsupervised learning

#### II.1.4.1 Différence entre apprentissage supervisé et non supervisé

Les deux types d'apprentissages se distinguent principalement par la manière dont les données d'entrée sont disposées. Mais il y a d'autres différences, comme le montre le tableau II- 2 [19].

	Les données d'entrée	Nombre de classes	Précision du résultat	Schéma
Supervised Learning	Utilise des données connues et étiquetées en tant qu'entrée	Connues	Résultat précis et fiable	
Unsupervised Learning	Utilise des données inconnues en tant qu'entrée non étiquetée	Inconnues	Précision et fiabilité Modér	

**Tableau II - 2 :** La différence entre apprentissage supervisé et non supervisé

## II.1.5 Apprentissage par renforcement

L'apprentissage par renforcement implique d'acquérir des compétences grâce à des essais et des erreurs. Dans cette approche, un individu perçoit son état et agit. Chaque action est récompensée par une récompense numérique. Le but de l'agent est de maximiser la récompense totale qu'il reçoit au cours du temps[20].

## II.1.6 Processus d'Apprentissage Automatique

Comme mentionné dans le paragraphe précédent, un projet de Machine Learning est un processus cyclique que l'on peut décrire par 8 grandes étapes. Chaque étape peut en influencer/remettre en question une autre, la finalité étant de fournir une réponse statistiquement robuste vis-à-vis d'une problématique initiale.

Le découpage que nous proposons en 8 étapes n'est pas unique et peut être sujet à discussion, notamment car chaque étape est interconnectée avec les autres. Cependant, cela permet de rendre explicite le cycle de vie d'un projet et d'articuler sa présentation de façon cohérente. Ce découpage est représenté dans la figure ci-dessous [21].

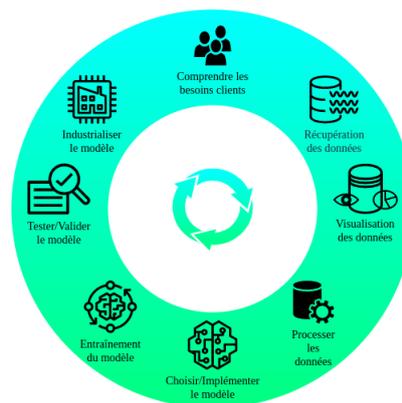


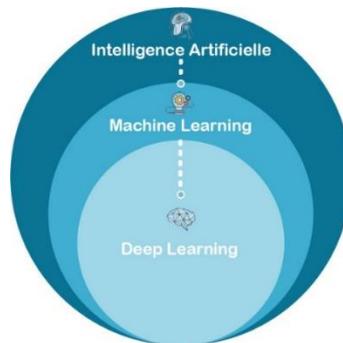
Figure II- 4 : Le cycle de vie d'un projet de Machine Learning en 8 étapes

## II.2 L'apprentissage en profondeur (Deep Learning)

Le Deep Learning (en anglais automatique) a été introduit dans le but de rapprocher le machine Learning de son objectif principal : l'intelligence artificielle. Il s'agit des algorithmes qui prennent leur inspiration de la structure et du fonctionnement du cerveau. Ils ont la capacité de modéliser des relations complexes entre les données dans leur objectif.

Le Deep Learning repose sur l'idée des réseaux de neurones artificiels et est conçu pour gérer de grandes quantités de données. En ajoutant des couches de traitement multiples, il a la capacité d'extraire des caractéristiques à partir des données brutes grâce à des transformations

linéaires et non linéaires, tout en apportant une intervention humaine minimale pour apprendre sur ces caractéristiques [22].



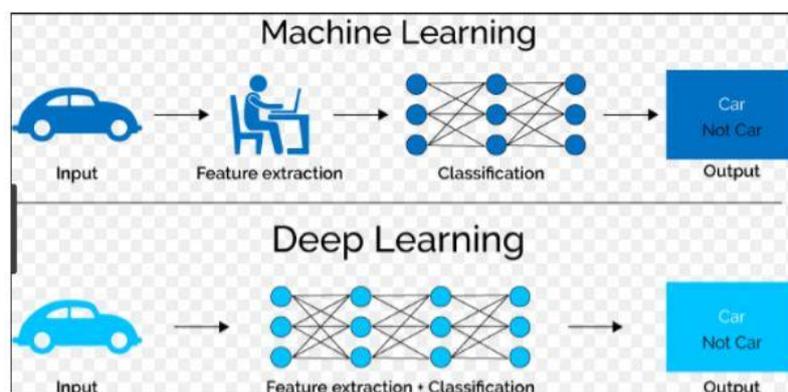
**Figure II- 5:** La relation entre l'intelligence artificielle, le ML et le Deep Learning

### II.2.1 Applications du Deep Learning

- Reconnaissance d'image.
- Traduction automatique.
- Voiture autonome.
- Diagnostic médical.
- Recommandations personnalisées.
- Modération automatique des réseaux sociaux.
- Exploration spatiale.
- Robots intelligents...exc.

### II.2.2 Des différences: Machine learning contre Deep Learning

Dans la figure 2, l'apprentissage automatique montre que les caractéristiques sont extraites manuellement alors que dans l'apprentissage profond, les caractéristiques sont extraites automatiquement parce qu'elles font partie de l'architecture.



**Figure II- 6:** La différence entre machine Learning et Deep Learning [23]

Et d'autres nous les résumons dans le tableau suivant [24] :

	Machine Learning	Deep Learning
De quoi s'agit-il?	Le ML est une méthodologie d'intelligence artificielle (IA). Le ML n'est pas forcément du deep learning.	Le deep learning est une méthodologie avancée de ML. Le deep learning est forcément du machine learning
Idéal pour	Le ML est la solution idéale pour les tâches bien définies avec des données structurées et étiquetées.	Le deep learning est idéal pour les tâches complexes qui nécessitent que les machines puissent interpréter des données non structurées.
Approche de résolution des problèmes	Le ML permet de résoudre des problèmes grâce aux statistiques et aux mathématiques.	Le deep learning associe les statistiques et les mathématiques à l'architecture des réseaux neuronaux.
Formation	Vous devez sélectionner et extraire manuellement des caractéristiques à partir de données brutes et leur attribuer des poids pour entraîner un modèle ML.	Les modèles de deep learning peuvent auto-apprendre en utilisant les commentaires provenant d'erreurs connues.
Ressources requises	Le ML est moins complexe et possède un volume de données plus faible.	Le deep learning est plus complexe et implique un volume de données très élevé.

**Tableau II - 3 : La différence entre Machine Learning et Deep Learning**

### II.3 Vision par Ordinateur

La vision par ordinateur associe caméras, edge computing, informatique en nuage, logiciels et intelligence artificielle (IA) pour aider les systèmes à « voir » et à identifier les objets. Intel possède un riche portefeuille de technologies permettant de mettre en œuvre l'IA, notamment des processeurs à usage général et des unités de traitement de la vision (VPU) qui accélèrent le traitement. Précieux dans un large éventail d'environnements, les systèmes de vision par

ordinateur peuvent reconnaître les objets et les personnes rapidement, analyser les données démographiques, inspecter les produits manufacturés et bien plus encore [25]. La vision par ordinateur est comparable aux yeux humains et à la capacité de notre cerveau à traiter et analyser les images perçues.

### **II.3.1 Utilisation de la vision par ordinateur**

Les technologies actuelles de vision par ordinateur sont adaptées à divers domaines, allant de la production à la vente au détail en passant par la finance. Ils soutiennent les entreprises dans leur expansion et leur amélioration de l'IA en périphérie. Les fonctions principales des systèmes de vision par ordinateur actuels sont la détection, la reconnaissance et la classification des objets.

- Détection et reconnaissance d'objets.
- Imagerie médicale .
- Communautés et parcs intelligents.
- Vision par ordinateur pour des villes plus intelligentes.
- Vision par ordinateur pour l'industrie manufacturière.
- Analytique dans la vente au détail.
- Systèmes de transport intelligents.

### **II.3.2 La détection d'objets**

La détection d'objets est un domaine crucial de la recherche, car les avancées visent à créer des systèmes qui se rapprochent des capacités humaines en matière de perception, de suivi et de reconnaissance d'objets. Son importance réside dans le fait que des résultats précis dans cette phase conduisent à une reconnaissance précise. Cependant, cette tâche est complexe en raison de divers défis tels que la taille, la forme, l'éclairage, la variété des objets, la réponse en temps réel et la complexité des arrière-plans.

Dans la vie quotidienne, la détection d'objets trouve des utilisations intéressantes. Par exemple, pour les malvoyants, un dispositif de détection d'obstacles intégré à une canne ou à des lunettes pourrait faciliter la navigation des utilisateurs en repérant les obstacles à proximité et en leur fournissant des alertes sonores ou haptiques pour les éviter. Cela leur donnerait la possibilité de se déplacer de façon autonome et sécurisée dans différents milieux.

### II.3.3 La détection et le suivi d'objets

La détection d'objets dans une image implique deux éléments principaux :

**Détection** : L'objectif est de dessiner des cadres (bounding boxes) autour des divers objets présents dans l'image afin de les repérer spatialement. Cette procédure permet de repérer la présence et la position des objets dans l'illustration.

**Suivi** : Après avoir repéré les objets, on leur attribue des étiquettes ou des labels afin de les identifier selon leur classe respective. Ceci permet de classer les objets repérés selon leur type ou leur classe, ce qui est crucial pour une meilleure compréhension du contenu de l'image.

Les approches les plus récentes en matière de détection d'objets peuvent être catégorisées en deux principales catégories :

**Méthodes à une étape** : Ces méthodes sont axées sur la rapidité de l'inférence, ce qui signifie qu'elles sont conçues pour détecter les objets rapidement.

**Méthodes à deux étapes** : Ces méthodes accordent une plus grande importance à la précision de la détection des objets. Elles sont généralement composées de deux étapes distinctes : la génération de régions d'intérêt suivie de la classification des objets dans ces régions.

### II.3.4 Outils utilisés en computer vision

Il existe plusieurs bibliothèques populaires utilisées en vision par ordinateur pour la programmation, l'analyse et le traitement des images [26] :

**OpenCV** : Bibliothèque open source en C++ offrant des outils pour l'acquisition, le traitement, la segmentation d'images, et la reconnaissance d'objets. Disponible en Python, Java, C#, et MATLAB.

**TensorFlow** : Bibliothèque de Deep Learning développée par Google, permettant la création et l'entraînement de modèles de réseaux de neurones. Disponible en Python, C++, Java, JavaScript, et Swift.

**PyTorch** : Bibliothèque de Deep Learning développée par Facebook pour la création et l'entraînement de modèles de réseaux de neurones. Disponible en Python.

**Keras** : Interface conviviale de Deep Learning pour la création, l'entraînement et le déploiement de modèles de réseaux de neurones. Disponible en Python.

**Caffe** : Bibliothèque de Deep Learning écrite en C++ principalement utilisée pour la classification d'images, la détection d'objets et la segmentation d'images. Disponible en Python et MATLAB.

### II.3.5 Algorithmes de Vision par Ordinateur

Les algorithmes classiques de vision par ordinateur comprennent des méthodes comme le filtrage d'images, la segmentation et la détection de contenus. Néanmoins, grâce à l'émergence du Deep Learning, les réseaux de neurones convolutifs (CNN) sont devenus la méthode de choix pour de nombreuses tâches de vision par ordinateur.

Il existe quatre grandes catégories d'algorithmes utilisés par le computer vision [26]

### II.3.6 Algorithmes de détection d'objet

L'utilisation d'algorithmes de détection d'objets permet de repérer la présence d'objets particuliers dans des images ou des vidéos. Ces technologies sont fréquemment utilisées dans des secteurs tels que la surveillance, la reconnaissance faciale et la reconnaissance de plaques d'immatriculation. Voici une liste des algorithmes les plus appréciés :

**R-CNN** (Region-based Convolutional Neural Networks) :

- Un des premiers modèles de détection d'objets utilisant des CNN.
- Utilise des régions proposées pour extraire des caractéristiques et les classifier.

**YOLO** (You Only Look Once):

- Modèle basé sur des CNN utilisant un réseau unique pour détecter des objets.
- Prédit simultanément les boîtes englobantes et les classes d'objets.

**SSD** (Single Shot Detector) :

- Conçu pour la détection en temps réel.
- Utilise des filtres de convolution à différentes échelles pour détecter des objets de diverses tailles.

**R-CNN Faster** :

- Variante de R-CNN utilisant des méthodes de régionalisation basées sur des CNN.
- Améliore l'efficacité et la rapidité de la détection.

**R-CNN Mask :**

- Extension de Faster R-CNN.
- Ajoute la segmentation d'images à la détection d'objets, permettant une approche multi-tâches pour la détection, la prédiction des boîtes englobantes et la segmentation des images.

**II.3.7 Algorithmes de segmentation d'images**

Les techniques de segmentation d'images sont employées afin de séparer une image en différents segments ou régions, ce qui facilite l'analyse et la compréhension de celle-ci. On les utilise fréquemment dans des secteurs tels que le traitement d'images médicales, la surveillance et la détection de formes. Voici certains exemples courants :

**K-means clustering**

- Utilise l'algorithme de clustering K-means pour segmenter une image en fonction des couleurs ou des intensités des pixels.
- Les régions sont ensuite assignées à des étiquettes pour identifier les différentes parties de l'image.

**Watershed segmentation**

- Utilise les gradients d'intensité pour diviser l'image en différentes régions.
- Souvent employée pour segmenter des régions contenant des bordures ou des lignes.

**Graph cut**

- Utilise des graphes pour définir les frontières entre les différentes régions.
- Couramment utilisée pour la segmentation d'objets dans les images.

**Convolutional neural networks (CNN)**

- Utilise des réseaux de neurones pour identifier les régions d'intérêt dans une image.
- Fréquemment utilisés pour la segmentation d'images médicales, comme la segmentation de tumeurs dans des images de tomographie par ordinateur (CT) ou d'IRM.

**U-Net**

- Architecture de réseau de neurones conçue spécifiquement pour la segmentation d'images biomédicales.

- Utilise une structure d'encodeur-décodeur pour capturer les caractéristiques de l'image à plusieurs niveaux de résolution et produire des masques de segmentation précis.

### **II.3.8 Algorithmes de classification d'images**

Les algorithmes de classification d'images permettent de catégoriser les images en différentes classes. Ils sont couramment utilisés dans des applications telles que la reconnaissance d'objets, la détection de fraudes et le tri automatique des déchets. Voici quelques exemples de ces algorithmes :

#### **Réseaux de neurones profonds (DNN)**

- Utilisent des réseaux de neurones à plusieurs couches pour apprendre les caractéristiques des images à partir des pixels bruts.
- Les réseaux de neurones convolutifs (CNN) sont fréquemment employés pour la classification d'images.

#### **Support Vector Machine (SVM)**

- Utilisé pour la classification binaire et multi-classes en utilisant un hyperplan de séparation.
- Souvent appliqué à la classification basée sur des caractéristiques extraites des images, comme les descripteurs SIFT ou HOG.

#### **Random Forest**

- Ensemble d'arbres de décision utilisé pour la classification d'images.
- Généralement utilisé pour la classification basée sur des caractéristiques extraites des images, comme les descripteurs SIFT ou HOG.

#### **Réseaux de neurones récurrents (RNN)**

- Utilisés pour la classification de séquences d'images, telles que des vidéos.
- Capturent la dépendance temporelle entre les images pour améliorer la classification.

#### **Réseaux de neurones résiduels**

- Utilisent des blocs résiduels pour améliorer la performance des réseaux de neurones profonds.
- Souvent utilisés pour la classification d'images à grande échelle.

### II.3.9 Algorithmes de reconnaissance

Les algorithmes de reconnaissance de formes, d'images et de mouvements sont essentiels dans des applications telles que la surveillance vidéo, la reconnaissance de gestes, la reconnaissance d'activités humaines et la réalité augmentée. Voici quelques exemples de ces algorithmes :

#### Template Matching

- Utilise une image modèle pour trouver des occurrences de cette forme dans une image cible.
- Méthode simple mais peu robuste aux variations d'échelle, de rotation et d'éclairage.

#### Contours Actifs (Active Contours)

- Utilise des contours actifs, ou "snakes", pour détecter les bords des objets dans une image.
- Les contours actifs se déforment pour épouser la forme de l'objet à détecter.

### II.3.10 Algorithmes de Détection de Mouvement

- Utilisés pour détecter les mouvements dans des séquences d'images.
- Techniques courantes : différence d'images et flux optique pour détecter les changements dans la scène.

#### Réseaux de Neurones Convolutifs (CNN)

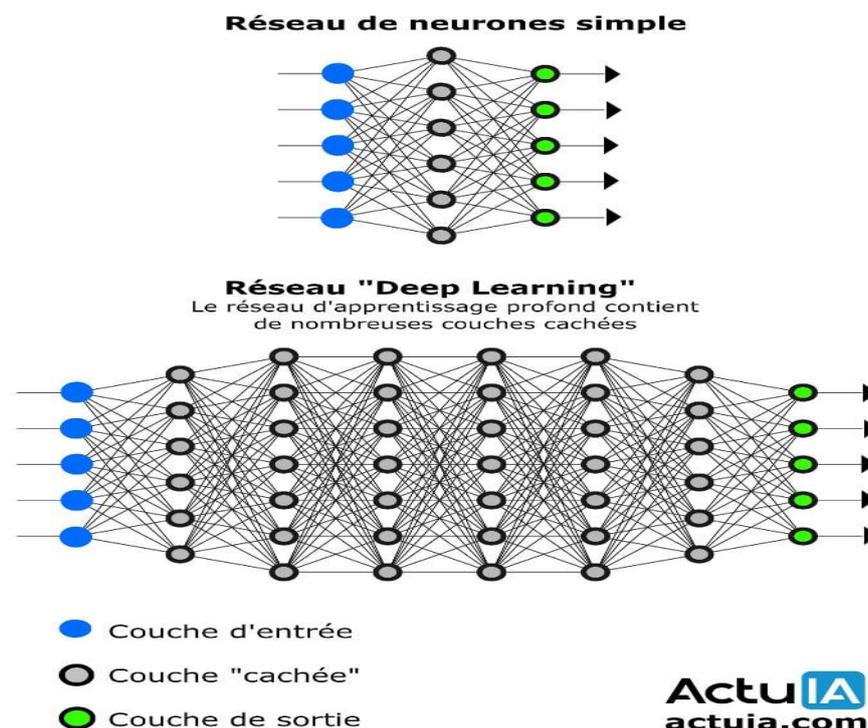
- Utilisés pour la reconnaissance de formes et la détection de mouvements.
- Les CNNs peuvent apprendre à détecter des caractéristiques pertinentes dans une image ou une séquence d'images pour identifier des objets ou des mouvements spécifiques.

### II.3.11 Algorithmes de Reconnaissance de Mouvements Humains

- Spécifiquement conçus pour la reconnaissance de mouvements humains, comme la détection de gestes ou la reconnaissance d'activités.
- Utilisent souvent des réseaux de neurones, des modèles de Markov cachés ou des SVM pour la reconnaissance de mouvements.

## II.4 Réseaux de neurones profonds

Un réseau neuronal profond (RNP) est une technique d'apprentissage automatique qui permet à un ordinateur, en l'entraînant, d'effectuer des tâches qui seraient très difficiles à réaliser à l'aide de techniques de programmation conventionnelles. IL est un système beaucoup plus complexe qu'un système neuronal "simple". Réseaux neuronaux traditionnels, généralement composés de 2 ou 3 couches cachées par contre les RNP, qui peut comporter 150 couches cachées [27].



**Figure II- 7 :** Réseau neuronal simple et réseau de neurones profonds [28]

### II.4.1 Importance des réseaux de neurones profonds

Voici quelques-unes des raisons pour lesquelles les réseaux de neurones profonds sont si importants :

- Ils sont capables d'apprendre des données et modéliser des relations complexes
- Ils sont très précis.
- Ils sont polyvalents et évolutifs.
- Hiérarchisation des caractéristiques

## II.4.2 Réseaux de neurones artificiels

Le perceptron est le réseau neuronal le plus ancien, créé par Frank Rosenblatt en 1958[29]. C'est un système dont la conception est à l'origine inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques [30]. Les réseaux artificiels de neurones sont des réseaux étroitement liés grâce à des processeurs élémentaires qui fonctionnent simultanément. Chaque processeur fondamental (neurone artificiel) évalue une sortie spécifique en fonction des données qu'il reçoit.

Ainsi qu'il copie le cerveau humain pour favoriser l'apprentissage. Il s'agit donc d'un système qui se base sur le fonctionnement du cerveau humain pour l'adapter à des ordinateurs équipés de fonctions d'intelligence artificielle. Grâce au réseau de neurones artificiels, l'ordinateur arrive à résoudre des problèmes de manière autonome. Le réseau améliore aussi les capacités de l'ordinateur [31].

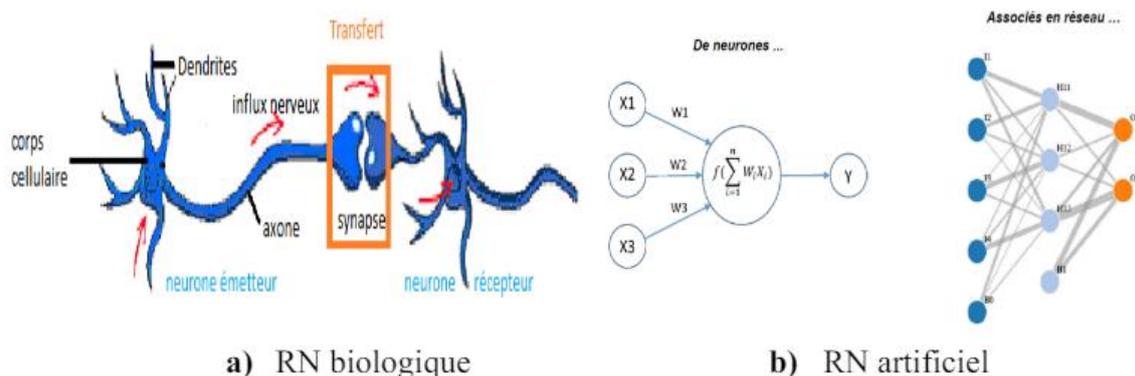


Figure II- 8 : Définition d'un réseau de neurones [32]

## II.4.3 Fonctionnement de réseaux de neurones

Les réseaux neuronaux artificiels se composent de diverses couches de neurones artificiels interconnectés, alimentés par des fonctions d'activation qui permettent de les activer/désactiver. À l'instar des algorithmes des machines classiques, les réseaux neuronaux acquièrent également certaines valeurs lors de leur formation.

En résumé, chaque neurone reçoit une série d'entrées et de poids aléatoires additionnées d'une valeur de biais statique (spécifique à chaque couche de neurones). Cette valeur est ensuite transmise à une fonction d'activation appropriée qui détermine la valeur finale à donner par le neurone.

Il existe différentes fonctions d'activation disponibles selon la nature des valeurs d'entrée. Une fois la sortie générée à partir de la couche finale du réseau neuronal, la fonction de perte (entrée vs sortie) est calculée et une rétropropagation est effectuée où les poids sont ajustés pour minimiser la perte. La recherche des valeurs optimales des poids est le point sur lequel se concentre l'opération globale.

Différentes fonctions d'activation peuvent être utilisées en fonction de la nature des valeurs d'entrée. Après avoir généré la sortie à partir de la couche finale du réseau neuronal, on calcule la fonction de perte (entrée/sortie) et on procède à une rétropropagation où les poids sont ajustés afin de réduire la perte. Le but principal de l'opération globale est de trouver les valeurs optimales des poids[33].

**Les poids** sont des valeurs numériques multipliées par les entrées. En rétropropagation, ils sont modifiés pour réduire la perte. En termes simples, les poids sont des valeurs apprises automatiquement à partir des réseaux de neurones.

Ils s'auto-ajustent en fonction de la différence entre les résultats prévus et les apports de formation.

**La fonction d'activation** est une formule mathématique qui aide le neurone à s'allumer/s'éteindre.

**La couche d'entrée** représente les dimensions du vecteur d'entrée.

**La couche cachée** représente les nœuds intermédiaires qui divisent l'espace d'entrée en régions avec des limites (douces). Il prend en compte un ensemble d'entrées pondérées et produit une sortie via une fonction d'activation.

**La couche de sortie** représente la sortie du réseau neuronal.

#### II.4.4 Types de réseaux neurone

Il y a de multiples types de réseaux de neurones disponibles ou qui pourraient encore être développés. On peut les classer selon différentes caractéristiques telles que leur structure, leur flux de données, l'utilisation des neurones et leur densité, ainsi que leurs couches et filtres d'activation en profondeur, entre autres : Perceptron, Réseaux de neurones à action directe, Perceptron multicouche, Réseau neuronal convolutif, Réseaux de neurones à fonction de base radiale, Réseaux de neurones récurrents, Modèles séquence à séquence, Réseau neuronal modulaire [33].

Notre attention s'est principalement portée sur les réseaux neuronaux à propagation, également connus sous le nom de perceptrons multicouches (MLP). Ils se constituent d'un niveau d'entrée, d'une ou de plusieurs couches masquées et d'un niveau de sortie. Même s'ils

sont aussi couramment désignés sous le nom de MLP, il convient de souligner qu'ils sont en réalité constitués de neurones sigmoïdes, et non de perceptrons, car la majorité des problèmes du monde réel ne sont pas linéaires. En général, ces modèles sont entraînés à l'aide de données, qui sont les bases de la vision par ordinateur, du traitement du langage naturel et d'autres réseaux neuronaux [29].

### 1. Les réseaux de neurones récurrents (RNN)

Ils sont identifiés par leurs boucles de rétroaction. Ces algorithmes d'apprentissage sont principalement utilisés lorsque vous utilisez des données de séries temporelles pour faire des prédictions de résultats, telles que les prévisions boursières ou les prévisionnels de vente[29]. Parmi ses applications : traitement de texte comme suggestion automatique, vérifications grammaticales, traitement de la synthèse vocale, tagueur d'images, analyse des sentiments, traduction.

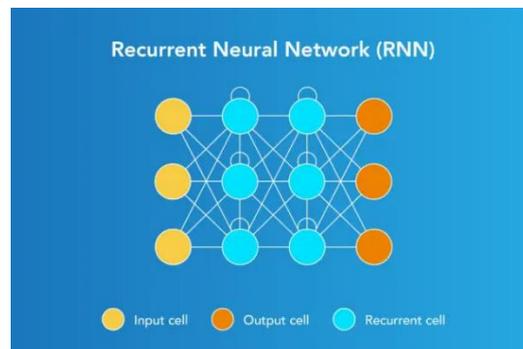
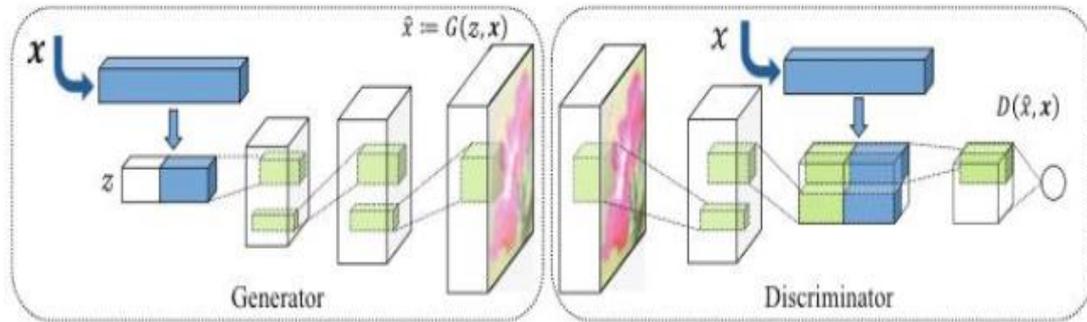


Figure II- 9 : RNN

### 2. Réseaux Adversarial Génératifs (GAN)

Ce réseau combine les modèles génératifs et discriminatifs pour analyser la distribution des données produites par le modèle génératif et différencier les données d'entrée originales des données générées. Dans ce jeu à somme nulle, le modèle génératif tente de produire des données ressemblant aux données d'entrée originales, tandis que le modèle discriminatif s'efforce de distinguer les données d'entrée réelles des données générées par le modèle.

La force de ce réseau réside dans sa capacité à maintenir la cohérence après avoir atteint l'équilibre, à ne pas avoir besoin d'inférence approximative ou de chaînes de Markov, et à pouvoir être entraîné avec des données manquantes ou restreintes. Par ailleurs, le désavantage de l'utilisation du GAN réside dans la difficulté de trouver un équilibre entre les modèles génératifs et discriminatifs [34].



**Figure II- 10 :** Architecture de Réseaux Adversariaux Génératifs

### 3. Les réseaux neuronaux convolutifs (CNN)

Aujourd'hui, les réseaux de neurones convolutionnels sont considérés comme les modèles les plus efficaces pour organiser des images. Connus sous le nom de CNN, signifiant convolutional Neural Network en anglais, ils se composent de deux parties distinctes. Une image est présentée en entrée sous la forme d'une matrice de pixels. Elle possède deux dimensions pour une image en couleurs. La troisième dimension, de profondeur 3, est utilisée pour représenter les couleurs fondamentales telles que le rouge, le vert et le bleu [35].

La partie convolutive proprement dite d'un réseau de neurones convolutionnels est la première partie. Elle opère en tant qu'outil d'extraction des caractéristiques des images. Une image a été traversée par une série de fils, ou noyaux de convolution, créant ainsi de nouvelles images appelées cartes convolutionnelles. Certaines options de filtrage temporaire réduisent la résolution de l'image en utilisant une option de maximum local. Finalement, on met en place des cartes de convolutions et on les concatène en un vecteur de caractéristiques, appelé code CNN, ce code dernier en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image [35].

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories [35].

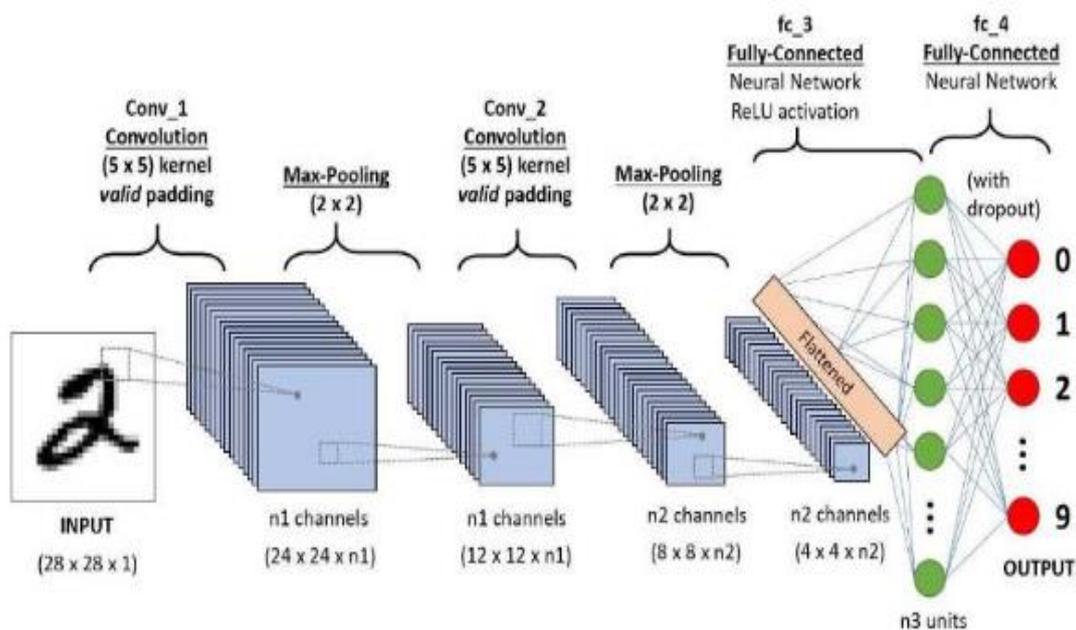


Figure II- 11 : Architecture d'un réseau de neurone convolutif [36]

**La couche Convolution :** Dans cette opération, une partie de l'image est soumise à une convolution avec un filtre, produisant un entier unique dans le volume de sortie. Ensuite, le filtre se déplace sur le champ récepteur suivant de l'image d'entrée, répétant cette opération jusqu'à ce que toute l'image soit parcourue.

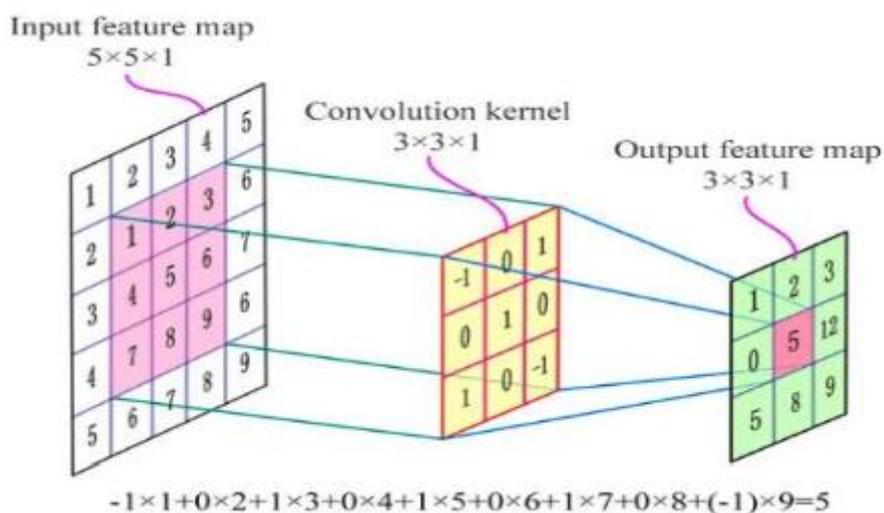


Figure II- 1 : La couche de convolution

**Couche de mise en commun (Pooling) :** C'est une étape de sous-échantillonnage souvent insérée entre deux couches de convolution.

Sa fonction est de réduire progressivement la taille de la carte de fonctionnalités (matrice de convolution) pour réduire les paramètres et les calculs réseau, tout en conservant les informations importantes [37].

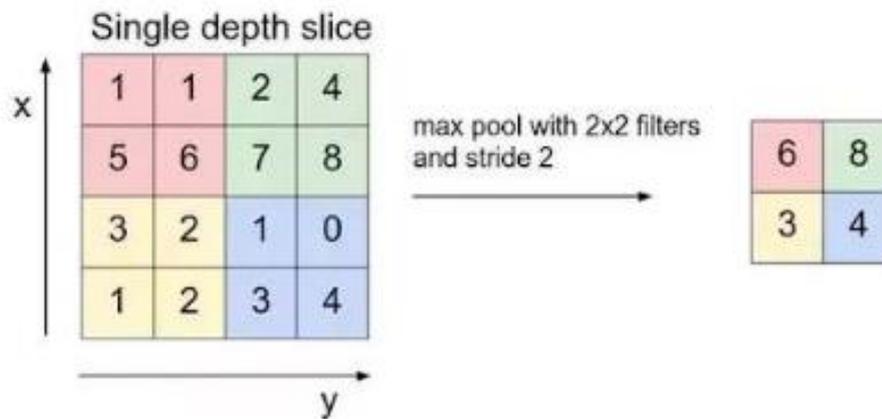


Figure II- 12 : La couche pooling

**Couche entièrement connecté (Fully Connected)** : reliées effectuent les mêmes fonctions que les ANN classiques et chercheront à générer des notes de classe à partir des activations, afin de les utiliser pour la classification. Il est aussi recommandé d'utiliser ReLu entre ces différentes couches afin d'améliorer les performances [38].

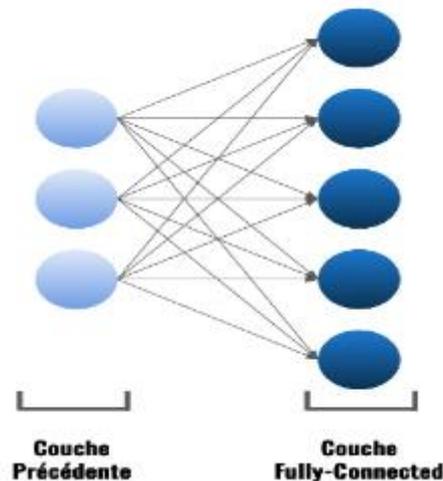


Figure II- 13 : Couche entièrement connecté (Fully Connected) [39]

**Couche de sortie** : située à la fin du réseau, contient des neurones chargés d'identifier les classes du modèle, le nombre de neurones dans cette couche dépendant du nombre de classes à distinguer. Dans le schéma suivant nous allons voir les meilleurs model dans les deux demain précédent :

: Couche entièrement connecté (Fully Connected) [39]

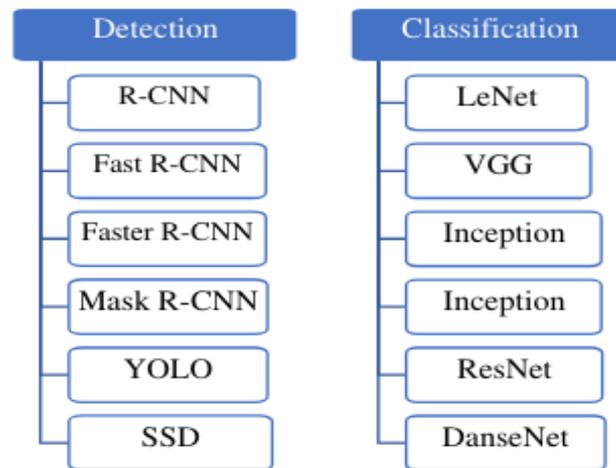


Figure II- 14 : Schéma des modèles de détection et classification par CNN

## II.5 Les modèles de détection par CNN

Les architectures CNN sont capables d'apprendre des caractéristiques plus complexes. Il existe deux types de modèles de détection d'objets par les architectures CNN. Le premier type de détection en deux coups est basé sur la proposition de région et comprend des modèles tels que RCNN, SPP-NET, FRCNN, Faster RCNN et le second type la détection à un coup est basé sur la régression et comprend MultiBox, AttentionNet, G-CNN, YOLO, SSD...etc[40].

✓ **Détection en deux coups** : cette se décompose en deux phases distinctes. Tout d'abord, elle génère des propositions de régions. Ensuite, dans la deuxième phase, elle effectue la classification de ces régions tout en affinant les prédictions de localisation.

✓ **Au contraire, la détection à un coup** saute l'étape de la proposition de région et produit la localisation finale et la prédiction du contenu en une seule fois [41].

Il existe plusieurs model de détection d'objet, nous avons présenté dans schéma suivant, dans un ordre chronologique :

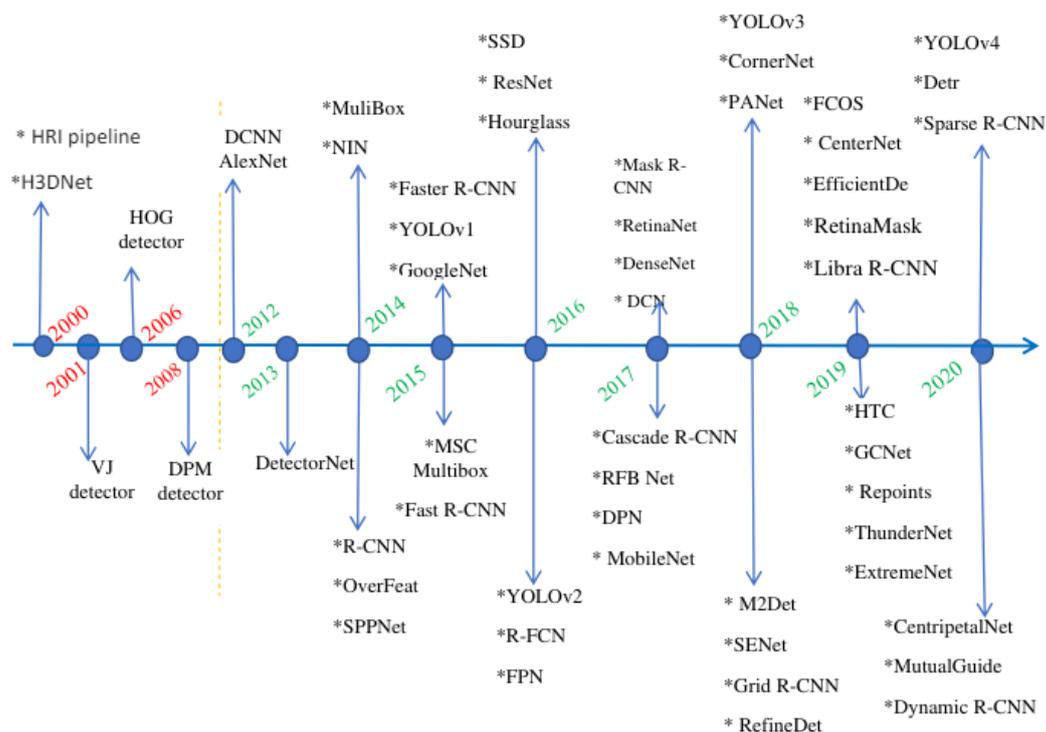


Figure II- 15 : Étapes importantes de la détection et de la reconnaissance des objets

## II.6 YOLO

You Only Look Once ou YOLO est l'un des algorithmes populaires de détection d'objets utilisé par les chercheurs du monde entier. Il a été décrit pour la première fois dans en 2015 dans l'article de Joseph Redmon et al [42] .

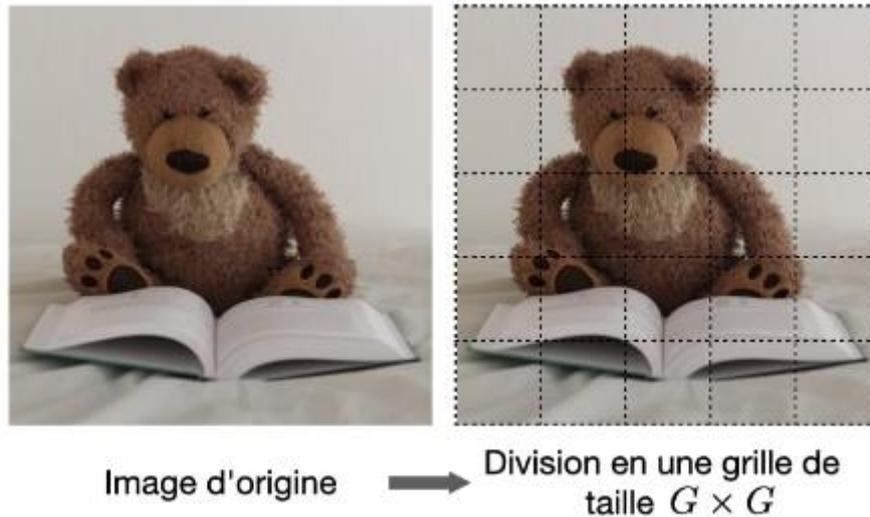
L'ensemble des caractéristiques de l'image est utilisé pour prédire chaque boîte englobante et toutes les boîtes englobantes de toutes les classes d'une image en même temps, ce qui signifie que le réseau raisonne globalement sur l'image et sur tous les objets qu'elle contient. La conception YOLO permet un apprentissage complet et des vitesses en temps réel tout en maintenant une précision moyenne élevée [42].

L'architecture de ce réseau est inspirée du modèle GoogLeNet pour la classification d'images. Ce réseau comporte 24 couches convolutives suivies de 2 couches entièrement connectées. Au lieu des modules d'initialisation utilisés par GoogLeNet, elle a utilisé simplement des couches de réduction  $1 \times 1$  suivies de  $3 \times 3$  couches convolutives [42].

L'algorithme de modèle YOLO est divisé en 3 étapes :

- ✚ **Diviser l'image en cellules avec une taille  $G \times G$**

On divise l'image en une grille de taille  $G \times G$  (en exemple  $5 \times 5$ ), ce qui donne  $N$  cellules au total. Cette cellule de la grille est responsable de la détection de cet objet.



**Figure II- 16 :** Diviser l'image en ( $G \times G$ ) grille

#### ✚ Chaque cellule prédit $B$ boîtes englobante

Après avoir divisé l'image en  $N$  cellules, chaque cellule de la grille prédit des boîtes englobantes  $B$  ainsi que des scores de confiance pour ces boîtes. Chaque boîte englobante comprend 5 prédictions :  $x$ ,  $y$ ,  $w$ ,  $h$  et un score de confiance. Les coordonnées  $(x, y)$  représentent le centre de la boîte par rapport aux limites de la cellule de la grille. La largeur ( $w$ ) et la hauteur ( $h$ ) sont prédites par rapport à l'image entière. Le score de confiance indique la probabilité que la boîte englobante contienne un objet et la précision de cette prédiction.

✚ **Suppression non maximale :** Cette étape est la dernière dans l'algorithme de détection, est utilisée pour gérer les situations où le même objet est détecté par plusieurs boîtes englobantes. Elle vise à "supprimer" les boîtes englobantes les moins probables, ne conservant que la meilleure. Le processus de cette technique se déroule en cinq étapes :

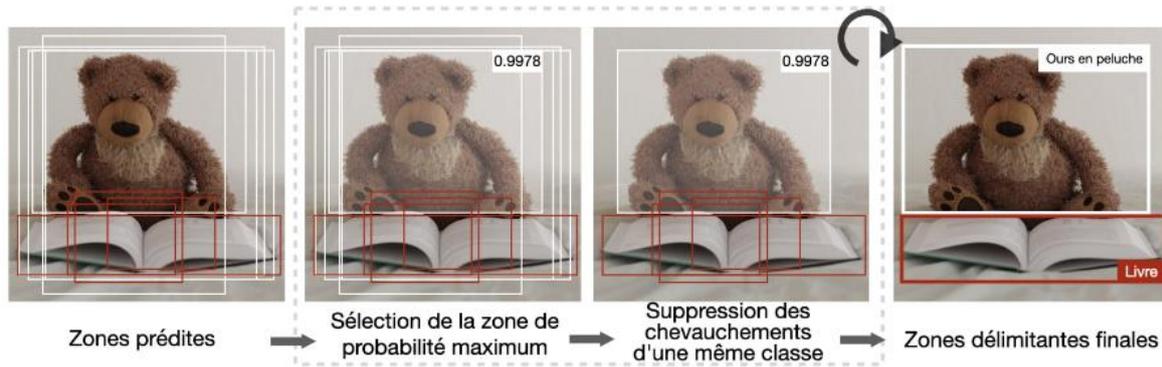
**Étape 1 :** Sélectionner la boîte avec le score d'objectivité le plus élevé.

**Étape 2 :** Ensuite, on compare le chevauchement (intersection sur union) de cette boîte avec d'autres boîtes.

**Étape 3 :** Supprimez les boîtes englobantes dont le chevauchement est  $>50\%$ .

**Étape 4 :** Passez ensuite au score d'objectivité le plus élevé suivant.

**Étape 5 :** Enfin, répétez les étapes 2 à 4 jusqu'à finir tous les objets dans l'image.



**Figure II- 17 :** Le résultat de Suppression non maximale [43]

Il y a eu 6 versions du modèle jusqu'à présent, chaque nouvelle version améliorant la précédente en termes de vitesse et de précision [36]. (Le model yolov2, yolov3, yolov4, yolov5, yolovX).

### II.6.1 Avantages

- **Vitesse** : YOLO est extrêmement rapide car il fait toutes les prédictions en une seule évaluation du réseau.
- **Précision** : YOLO est très précis pour la détection d'objets avec des taux de faux positifs relativement bas.
- **End-to-End** : YOLO traite la détection d'objets comme un problème de régression unique, ce qui simplifie le processus et réduit les erreurs d'accumulation.
- **Performance en temps réel** : YOLO est idéal pour les applications nécessitant une détection rapide, comme la surveillance vidéo en direct ou les systèmes de navigation pour les robots.
- **Robustesse** : YOLO est capable de détecter plusieurs objets de différentes classes dans une même image, ce qui le rend polyvalent pour diverses applications.

### Conclusion

L'IA est un outil puissant qui a le potentiel de transformer la façon dont les personnes malvoyantes naviguent dans le monde. En poursuivant la recherche et le développement dans ce domaine, nous pouvons créer des solutions innovantes qui améliorent considérablement la qualité de vie des personnes malvoyantes.

***CHAPITRE.III*** *Détection d'objets en utilisant une pi*  
*camera*

## Chapitre III. Détection d'objet en utilisant un pi camera

### Introduction

Dans ce chapitre, nous présentons notre projet de développer un système embarquée basée sur l'utilisation d'une carte Raspberry PI 4. En commençant d'abord par présenter le cahier des charges et les contraintes liées au développement de notre projet, puis nous expliquerons comment utiliser les différentes fonctions de notre application.

### III.1 Matériels et ressources utilisés

Cette section se concentre sur l'environnement de développement de notre système, (PC Portable, Raspberry Pi 4, Camera Raspberry, Carte mémoire, Alimentation, un écouteur)

#### III.1.1 PC Portable

- Processeur Intel(R) Core (TM) i5-7300 HQ CPU @ 2.50GHz 2.50 GHz.
- Processeurs logiques.
- Mémoire RAM : 8,00 Go
- Système d'exploitation : Windows 10.
- Environnement de programmation utilisé : Python 3.9.13
- Bibliothèques utilisées : numpy, time, openCv, os, imutils, imutils.video, gtts, pygame, tempfile, uuid, trading

##### III.1.1.1 Outils et langages de programmation

Pour la réalisation de notre projet sur un système de navigation pour les malvoyants, nous avons utilisé l'environnement de développement Visual Studio Code (VS Code) et le langage de programmation Python, Cette section détaille l'utilisation de ces outils et langages, ainsi que les bibliothèques spécifiques qui ont été intégrées au projet.

##### III.1.1.1 Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft. Il est devenu l'un des outils les plus populaires parmi les développeurs en raison de sa légèreté, de sa flexibilité et de ses nombreuses fonctionnalités.

### III.1.1.2 Python

Python est un langage de programmation interprété, polyvalent et très populaire pour le développement rapide d'applications. Il est connu pour sa syntaxe simple et lisible, ce qui le rend accessible aux débutants tout en étant puissant pour les développeurs expérimentés.

#### III.1.1.2.1 Caractéristiques Principales

- ✓ **Simplicité et Lisibilité** : La syntaxe claire de Python permet de se concentrer sur la résolution des problèmes plutôt que sur les complexités du langage.
- ✓ **Large Écosystème de Bibliothèques** : Python dispose d'un vaste écosystème de bibliothèques et de frameworks pour diverses applications, notamment la science des données, le développement web, l'automatisation, etc.
- ✓ **Communauté Active** : Une grande communauté de développeurs contribue à l'enrichissement continu du langage et à la disponibilité de ressources d'apprentissage.

#### III.1.1.2.2 Version de Python Utilisée

Nous avons utilisé Python version 3.9 pour ce projet. Cette version offre des fonctionnalités et des améliorations de performance pertinentes pour nos besoins.

#### III.1.1.2 Bibliothèques Utilisées

- **NumPy (np)**

NumPy est une bibliothèque fondamentale pour le calcul scientifique en Python. Elle permet de travailler efficacement avec des tableaux multidimensionnels (arrays) et fournit une large collection de fonctions mathématiques pour effectuer des opérations complexes sur ces tableaux.

**Utilisation dans le projet** : NumPy est utilisé pour le traitement et la manipulation des données numériques, notamment pour les calculs mathématiques liés au traitement des images et des données des capteurs.

- **Time**

La bibliothèque `time` fournit des fonctions pour manipuler le temps et les dates en Python. Elle permet d'obtenir l'heure actuelle, de mesurer des intervalles de temps, et de gérer les pauses dans l'exécution du programme.

**Utilisation dans le projet** : `time` est utilisée pour gérer les temporisations et les délais nécessaires dans les différentes opérations, comme l'attente de la réponse des capteurs ou la synchronisation des tâches.

- **OpenCV (cv2)**

OpenCV (Open Source Computer Vision Library) est une bibliothèque open-source très utilisée pour les applications de vision par ordinateur. Elle permet de capturer, traiter et analyser des images et des vidéos.

**Utilisation dans le projet :** OpenCV est utilisée pour la détection d'obstacles, le traitement des images capturées par la caméra, et l'analyse des flux vidéo en temps réel.

- **Os**

La bibliothèque `os` fournit une manière d'interagir avec le système d'exploitation, permettant d'effectuer des opérations telles que la manipulation de fichiers et de répertoires, et l'exécution de commandes système.

**Utilisation dans le projet :** 'os' est utilisée pour :

- Gérer les chemins de fichiers.
- Créer, lire et supprimer des répertoires et fichiers.
- Exécuter des commandes système nécessaires au bon fonctionnement de l'application.
- Récupérer des informations sur l'environnement d'exécution, comme les variables d'environnement et les identifiants de processus.

- **Imutils**

Imutils est une série de fonctions de commodité pour simplifier les opérations de traitement d'image avec OpenCV. Elle fournit des fonctions pour redimensionner, faire pivoter, et ajuster les images, ainsi que pour gérer les flux vidéo.

**Utilisation dans le projet :** Imutils est utilisée pour simplifier les opérations courantes de traitement d'image, comme le redimensionnement et la rotation des images capturées par la caméra.

- **Imutils.video**

La sous-bibliothèque 'imutils.video' fournit des classes et des fonctions pour travailler facilement avec les flux vidéo, notamment pour capturer des images à partir de webcams ou de caméras IP.

**Utilisation dans le projet :** 'imutils. video' est utilisée pour gérer le flux vidéo de la caméra en temps réel, notamment à travers la classe 'VideoStream pour simplifier la capture vidéo.

- **Gtts (gTTS)**

Google Text-to-Speech (gTTS) est une bibliothèque qui permet de convertir du texte en discours en utilisant le service Google Text-to-Speech.

**Utilisation dans le projet :** gTTS est utilisée pour générer des messages vocaux à partir de texte, fournissant ainsi des instructions verbales aux utilisateurs malvoyants.

- **Pygame**

Pygame est une bibliothèque pour le développement de jeux vidéo en Python, mais elle peut aussi être utilisée pour la gestion du son, de l'image, et des entrées utilisateurs.

**Utilisation dans le projet :** Pygame est utilisée pour jouer les fichiers audios générés par gTTS, offrant ainsi des retours sonores aux utilisateurs.

- **tempfile**

La bibliothèque 'tempfile' fournit des fonctions pour créer des fichiers et des répertoires temporaires de manière sécurisée, qui sont automatiquement supprimés après utilisation.

- Utilisation dans le projet : 'tempfile' est utilisée pour créer des fichiers temporaires pour stocker les fichiers audios générés par gTTS avant leur lecture.

- **uuid**

La bibliothèque 'uuid' est utilisée pour générer des identifiants uniques universels (UUID), souvent nécessaires pour garantir l'unicité des objets dans un système distribué.

**Utilisation dans le projet :** 'uuid' est utilisée pour générer des noms uniques pour les fichiers temporaires, évitant ainsi les conflits de nommage.

- **Threading**

La bibliothèque 'threading' fournit un support pour le multi-threading en Python, permettant l'exécution concurrente de plusieurs threads (légers processus).

**Utilisation dans le projet :** 'threading' est utilisée pour exécuter des tâches en parallèle, comme le traitement des flux vidéo et la génération de messages vocaux, améliorant ainsi la réactivité du système.

### III.1.2 Raspberry Pi 4



**Figure III - 1 :** Carte Raspberry Pi 4 B

### III.1.2.1 Définition

C'est un nano-ordinateur pouvant se connecter à un moniteur, à un ensemble clavier/souris et disposant d'interfaces Wi-Fi, Bluetooth et Ethernet.

Il fonctionne depuis une carte micro-SD et fonctionne avec un système d'exploitation basé sur Linux ou Windows 10 IoT. Etc. La Fondation Raspberry Pi propose une gamme de modèles Raspberry Pi, chacun offrant des spécifications matérielles différentes adaptées à divers besoins et cas d'utilisation (Raspberry Pi Zero, Raspberry Pi 3 Model B/B+, Raspberry Pi Compute Module, dans notre projet, nous avons utilisé le Raspberry Pi 4 Model B en raison de ses performances améliorées, de sa connectivité étendue et de sa disponibilité de mémoire RAM. Cela nous a permis d'exécuter efficacement les algorithmes de traitement d'image et de fournir une expérience utilisateur fluide et réactive.

### III.1.2.2 Caractéristiques

- Alimentation à prévoir : 5 Vcc/maxi 3 A\* via prise USB Type C (\* intensité maxi si toutes les fonctions sont utilisées)
- CPU : ARM Cortex-A72 quatre coeurs 1,5 GHz
- Wi-Fi: Dual-band 2,4 et 5 GHz, 802.11b/g/n/ac (Broadcom BCM43438)
- Bluetooth 5 compatible BLE (Broadcom BCM43438)
- Mémoire : 4 GB LPDDR4
- Circuit vidéo : VideoCore VI à 500 MHz
- 2 ports USB 2.0
- 2 ports USB 3.0
- Connecteur GPIO avec 40 broches d'E/S
- Port Ethernet Gigabit : RJ45
- Bus : SPI, I2C, série
- Support pour cartes micro-SD
- Sorties audios :
  - 2 x micro-HDMI avec gestion du 5.1
  - Jack 3,5 mm en stéréo (partagé avec vidéo)
- Sorties vidéo :
  - 2 x micro-HDMI (4K @ 60 fps maxi)
  - Jack 3,5 mm (partagé avec audio)
- Interface CSI pour caméra
- Interface DSI pour écran
- Dimensions : 88 x 58 x 17 mm
- Poids : 46 g

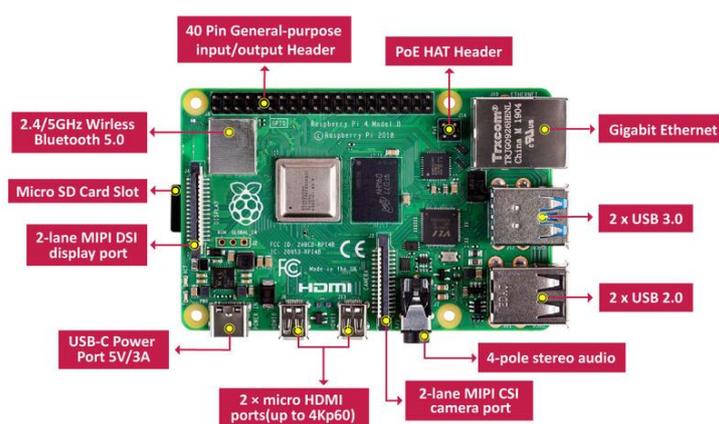


Figure III - 1 : Schéma du montage de circuit de commande

### III.1.2.3 Systèmes d'exploitation

#### - Préparation du Raspberry Pi

Sortez le Raspberry Pi de la boîte, puis de son sachet protecteur s'il y en a un. Le Raspberry Pi, comme tous les composants actuels, est sensible à l'électricité statique. Évitez de le manipuler, ce qui pourrait détériorer le SoC ou un autre circuit intégré.

Un objet présent sur l'établi ou le bureau (agrafe, trombone, ciseaux, stylo métallique...) peut provoquer un court-circuit sur la carte de circuit imprimé. Ceci peut endommager définitivement la carte. Il est important de poser le Raspberry Pi sur une surface propre et dégagée ou mieux, d'utiliser un boîtier adapté [44].

#### - Mise en place de la carte SD

Sur les Raspberry Pi 3 et 4, la carte est simplement maintenue par friction, il n'y a pas de système de verrouillage.

- Présentez la carte micro SD devant le connecteur situé sous la carte du Raspberry Pi 4. Les contacts de la carte micro SD doivent être tournés vers le circuit imprimé du Raspberry Pi.
- Insérez la carte micro SD dans son connecteur. Enfoncez la carte, elle ne se verrouille pas. Le maintien de la carte micro SD se fait par friction. Si la carte est à l'envers ou insérée en biais, vous sentirez une résistance importante. Ne forcez pas pour ne pas endommager la carte ou le connecteur [44].

#### - Connexion des câbles

La carte micro SD est prête, Raspberry Pi OS est installé, il est temps de brancher les câbles sur le Raspberry Pi et de découvrir toutes ses possibilités.

Branchez tous les câbles sur le Raspberry Pi, le bloc d'alimentation ne doit pas être relié au secteur pour le moment.

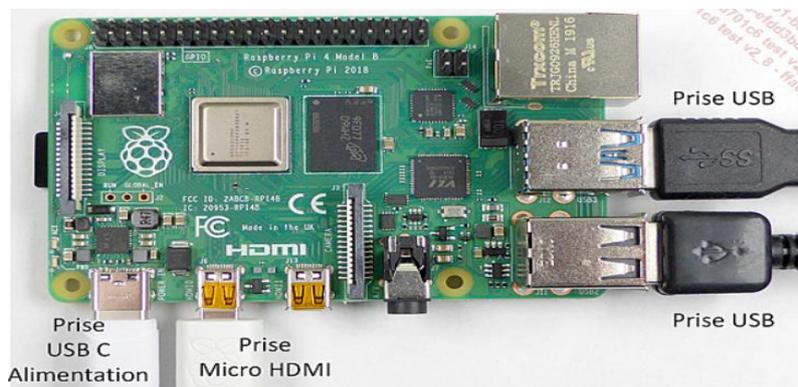


Figure III - 2 : Branchement des câbles avec la Raspberry

### - Mise sous tension

Le moment attendu est venu : la mise sous tension. Branchez le bloc d'alimentation dans une prise secteur.

L'écran relié au Raspberry Pi affiche d'abord une plage colorée, puis le logo du Raspberry Pi (4 framboises pour 4 cœurs) s'affiche en haut à gauche de l'écran... et au bout de quelques secondes le bureau de Raspberry Pi OS apparaît.

### - Démarrage de Raspberry Pi OS Desktop

Configuration initiale

- La configuration de Raspberry Pi OS Desktop se fait au démarrage grâce à un assistant.



- L'assistant vous informe de l'adresse IP attribuée au Raspberry Pi (s'il est connecté à un réseau local). Si vous cliquez sur le bouton **Cancel**, vous pouvez ignorer cette

étape de configuration. Le système sera alors configuré au minimum (pas de Wi-Fi, en anglais, clavier QWERTY...). Pour configurer le système avant de l'utiliser, cliquez sur **Next**.

Configurez le pays d'utilisation (**Country**), le clavier (**Language**) et le fuseau horaire (**Timezone**). Passez à la fenêtre suivante en cliquant sur le bouton **Next**.



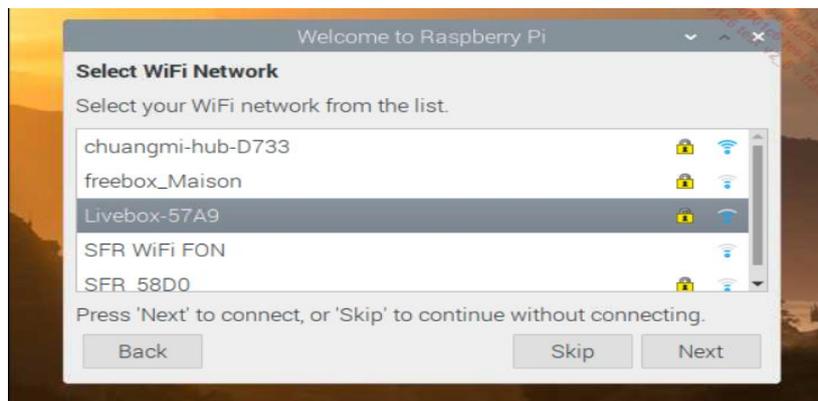
- Par défaut, le mot de passe du système est Raspberry. Si vous ne le changez pas, vous verrez apparaître une fenêtre d'avertissement à chaque démarrage. De plus, si votre Raspberry Pi est connecté à Internet, il est prudent de changer le mot de passe.



- La fenêtre suivante vous permet de régler la taille du bureau par rapport à l'écran et d'éliminer la marge noire qui pourrait apparaître. Cochez la case **This Screen...** si c'est le cas.



- Configurez ensuite le Wi-Fi. Sélectionnez votre point d'accès (ici **Livebox-57A9**). Si vous ne souhaitez pas utiliser le Wi-Fi, cliquez sur **Skip**.



- Saisissez le mot de passe pour la connexion...

Vous disposez maintenant d'un système Raspberry Pi OS opérationnel. Les chapitres suivants vous fourniront les bases pour utiliser le bureau en mode graphique ou pour découvrir l'interface texte de Raspberry Pi OS Linux.

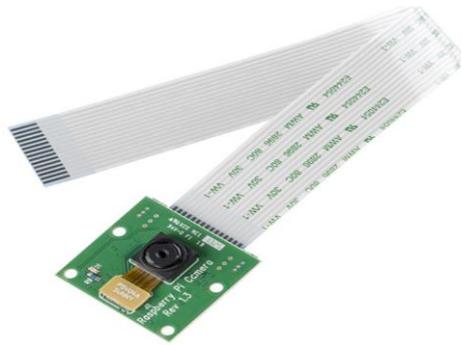


Figure III - 3 : Alimentation d'une carte raspberry avec le PC

### III.1.3 Camera Raspberry 5mp RV1.3

#### III.1.3.1 Définition

Ce module de caméra est un plug and play conçu pour Raspberry Pi. Il se fixe au Raspberry Pi via l'une des deux petites prises de la surface supérieure du tableau. Ce bus se déplace le long du câble ruban qui relie la carte de la caméra au Pi. Le capteur lui-même a une résolution native de 5 mégapixels, Résolution d'image fixe: 2592 x 1944, prend également en charge la vidéo 1080p30, 720p60 et 640x480p60 / 90. La caméra est prise en charge dans la dernière version de Raspbian, le système d'exploitation préféré de Raspberry Pi.



**Figure III - 4 :** Caméra Raspberry [45]

### III.1.3.2 Caractéristiques

Taille CCD: 1/4 pouce.

Ouverture (F): 1.8.

Longueur focale: 3,6 mm (réglable).

Champ de vision: 72 degrés.

Résolution de capteur préférée: 1080p.

Dimension: 1.42inch x1.04inch x0.94inch.

4 trous de vis utilisés pour une position fixe.

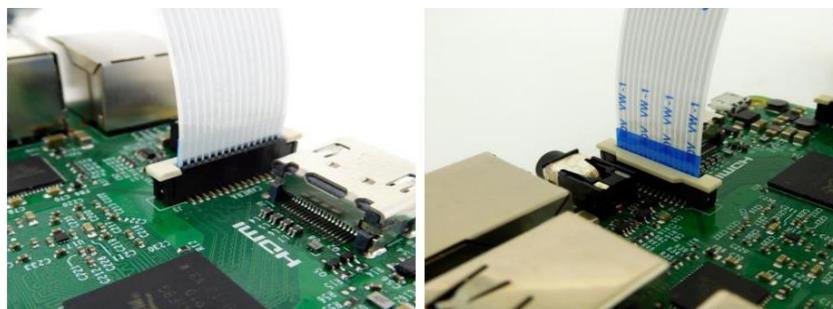
Prise en charge de la puissance de 3,3 V.

Prend en charge la connexion infrarouge LED ou flash LED.

### III.1.3.3 Connexion de caméra

La caméra se connecte au port CSI, situé entre les connecteurs HDMI et jack.

- 1) Tirez doucement le verrou du connecter (morceau blanc) vers le haut pour le débloquer.
- 2) Insérez-y la nappe de la caméra sans forcer, la partie bleue tournée vers la prise jack [46] .
- 3) Refermez le verrou, la nappe est maintenant maintenue.



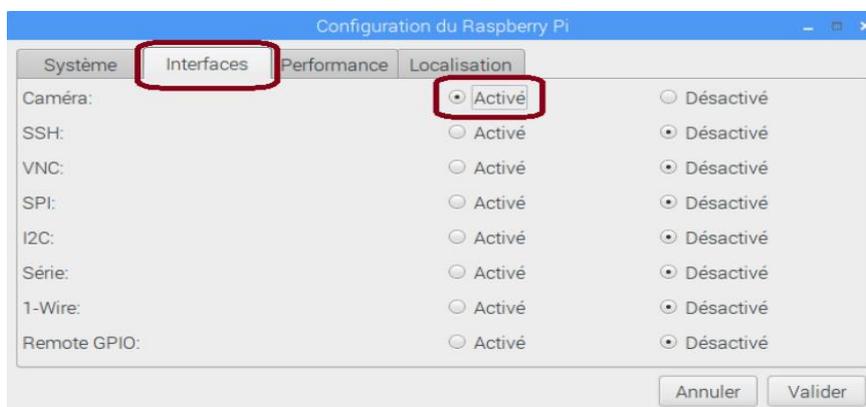
**Figure III - 5 :** Connecter la caméra avec la carte raspberry

### III.1.3.4 Configurer la caméra

Le système Raspbian n'est pas configuré pour utiliser la caméra d'origine, vous devez activer l'option dans les configurations : Allez dans le menu démarrer > Préférences > Configuration du Raspberry Pi.



Dans la fenêtre de configuration, sélectionnez l'onglet Interfaces et cochez la case Activé en face de Caméra.



Redémarrez ensuite votre Raspberry Pi.

### III.1.4 Carte mémoire

La carte micro SD de 16 Go de classe 10 est un support de stockage idéal pour une utilisation avec un Raspberry Pi.

La carte micro SD est un petit dispositif de stockage de données utilisé dans de nombreux appareils électroniques, y compris le Raspberry Pi. Avec une capacité de stockage de 16 Go, cette carte offre un espace suffisant pour installer et exécuter des systèmes d'exploitation, des applications et stocker des fichiers sur votre Raspberry Pi.

La classe 10 fait référence à la classification de vitesse de la carte. Les classes de vitesse sont utilisées pour déterminer la vitesse d'écriture minimale garantie par la carte. La classe 10 est la plus élevée dans la classification SD, ce qui signifie que la carte offre une vitesse d'écriture continue d'au moins 10 Mo/s. Cette vitesse élevée est particulièrement importante pour les applications exigeantes en termes de lecture et d'écriture, telles que l'exécution d'un système d'exploitation sur un Raspberry Pi.

La carte micro SD de classe 10 est spécifiquement recommandée pour une utilisation avec un Raspberry Pi en raison de sa capacité à fournir des performances optimales. Elle permet un chargement rapide du système d'exploitation, une exécution fluide des applications et une lecture/écriture efficace des données.

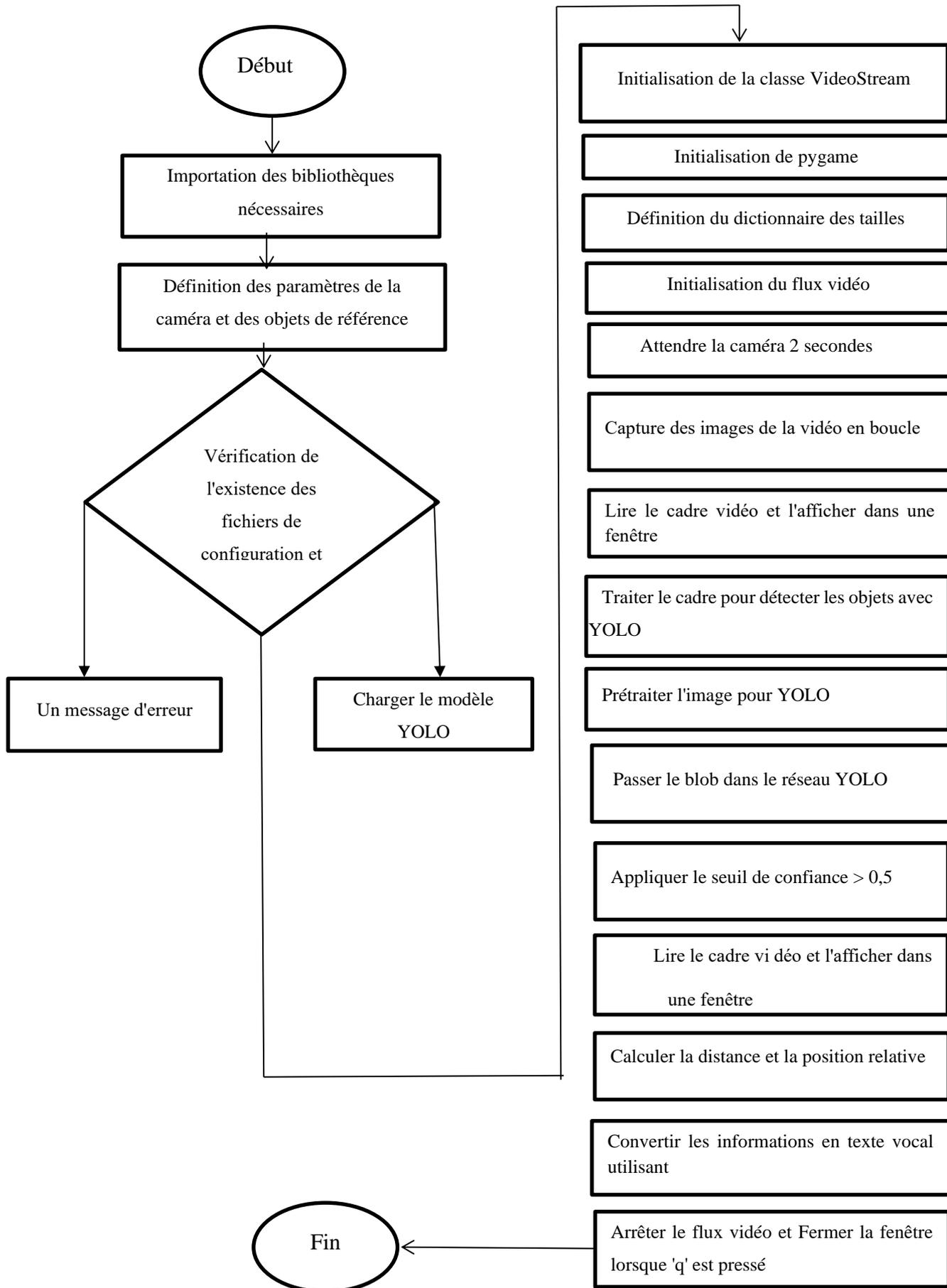
En utilisant une carte micro SD de classe 10, vous pouvez tirer pleinement parti des fonctionnalités et des performances de votre Raspberry Pi. Que vous utilisiez le Raspberry Pi pour des projets éducatifs, des médias, des serveurs ou d'autres applications, une carte de cette classe garantit une expérience utilisateur plus fluide et réactive.



**Figure III - 6 :** Carte mémoire SD

- ❖ Malheureusement, nous n'avons pas eu suffisamment de temps pour finaliser notre conception finale. Nous souhaitons que ce système fasse partie de lunettes intelligentes capables de détecter des objets et de convertir la vidéo en messages audio transmis via des écouteurs Bluetooth. Ceux-ci permettraient à la personne malvoyante de se déplacer confortablement à l'intérieur et à l'extérieur de son environnement.

### III.2 Organigramme du code



### III.3 Explication détaillée du code pour la détection d'objets

Le code fourni, écrit en Python sur VS code, utilise diverses bibliothèques pour le traitement d'images, la vision par ordinateur, la conversion texte-parole et la lecture audio afin d'effectuer la détection d'objets s. Décomposons le code en ses composants clés :

**Partie 1** : Contient toutes les importations et la configuration de base du YOLO.

#### 1 - Importation des bibliothèques et paramètre

Le code commence par importer les bibliothèques nécessaires à son fonctionnement :

```
# Importation des bibliothèques nécessaires.  
import numpy as np  
import time  
import cv2  
import os  
import threading  
from imutils.video import Videostream  
from gtts import gTTS  
import pygame  
import tempfile  
import uuid
```

Après les paramètres de la caméra et de l'objet de référence

```
# Paramètres de la caméra et de l'objet de référence  
FOCAL_LENGTH = 200 # Distance focale en pixels (à calibrer pour votre caméra)
```

**2 - La distance focale** est une caractéristique optique d'un objectif qui détermine la distance à laquelle il peut focaliser la lumière pour former une image nette. En photographie et en vision par ordinateur, la distance focale est souvent mesurée en millimètres (mm), mais dans le contexte de la vision par ordinateur, elle est souvent exprimée en pixels. La distance focale en pixels dépend de la résolution de la caméra et de la taille du capteur.

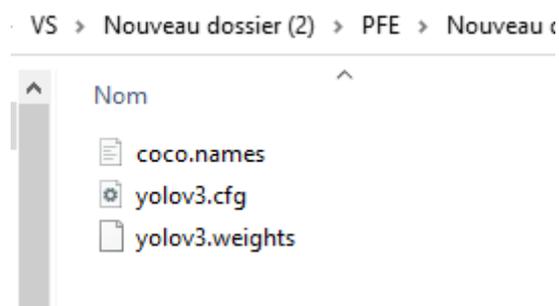
#### ➤ Pourquoi la Distance Focale en Pixels ?

Lorsqu'on utilise une caméra pour la vision par ordinateur, il est courant de travailler avec des images numériques où les dimensions des objets sont mesurées en pixels. Par conséquent, la distance focale en pixels est un paramètre crucial pour calculer des distances réelles à partir de mesures en pixels.

### 3 - Vérification des fichiers du modèle YOLO

Le code vérifie l'existence des fichiers du modèle YOLO requis :

- **coco.names** : Ce fichier contient la liste des étiquettes de classe d'objet elle contient 50 objets: person , bicycle, car, motorbike , aeroplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, cat, dog, horse, sheep, cow, umbrella, handbag, bottle, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, pizza, cake, chair, sofa,bed, diningtable, toilet, tvmonitor, laptop, mouse , remote, keyboard , cell phone , oven , sink, refrigerator, book, clock ,vase, scissors, hair drier , toothbrush ect.
- **yolov3.cfg** : Ce fichier définit la configuration de l'architecture du modèle YOLO.
- **yolov3.weights** : Ce fichier contient les poids entraînés du modèle YOLO.



```
# Vérification de l'existence des fichiers de configuration et des poids du modèle YOLO.
LABELS = open("C:/Users/A/Desktop/VS/Nouveau dossier (2)/PFE/coco.names").read().strip().split("\n")
config_path = "C:/Users/A/Desktop/VS/Nouveau dossier (2)/PFE/yolov3.cfg"
weights_path = "C:/Users/A/Desktop/VS/Nouveau dossier (2)/PFE/yolov3.weights"

# Verify if the files exist
if not os.path.exists(config_path):
    print(f"[ERROR] Configuration file not found: {config_path}")
if not os.path.exists(weights_path):
    print(f"[ERROR] Weights file not found: {weights_path}")
```

### 4 - Chargement du modèle YOLO

Le modèle YOLO est chargé à partir des fichiers de configuration et de poids en utilisant la fonction `cv2.dnn.readNetFromDarknet()`. Elle fait partie du module `cv2.dnn` de la bibliothèque OpenCV et est utilisée pour charger un modèle de réseau de neurones profond (DNN) qui a été pré-entraîné avec l'architecture YOLO (You Only Look Once) et les poids correspondants. Cette fonction charge l'architecture du modèle et ses paramètres entraînés, la rendant prête pour la détection d'objets.

### ➤ Paramètres

- **cfg** : Chemin vers le fichier de configuration du modèle YOLO. Ce fichier contient la définition de l'architecture du réseau de neurones, y compris les couches, les filtres, les tailles de noyau, les fonctions d'activation, etc.

- **weights** : Chemin vers le fichier des poids pré-entraînés. Ce fichier contient les poids et les biais des neurones du réseau, obtenus après l'entraînement du modèle sur un ensemble de données.

### ➤ Retour

- **net** : Un objet de type `cv2.dnn_Net` qui représente le réseau de neurones YOLO chargé en mémoire.

## Partie 2 : Initialisation des ressources

### 1 - Classe VideoStream

La classe `VideoStream` est conçue pour capturer des vidéos en temps réel depuis la caméra Raspberry Pi. Elle utilise la bibliothèque `OpenCV` pour ce faire et gère le flux vidéo dans un thread séparé afin de ne pas bloquer l'exécution du programme principal. Cette classe gère le flux vidéo et fournit des images pour la détection d'objets.

### 2 - Initialisation de `pygame.mixer`

Cette ligne initialise le module `pygame.mixer`, qui est utilisé pour gérer les effets sonores et permettre la lecture des fichiers audio.

Ces composants permettent à notre code de capturer des images en temps réel et de fournir des notifications audios basées sur les objets détectés.

## Partie 3 : Détection et estimation de la distance

### 1 - Dictionnaire des tailles réelles des objets

**OBJECT\_SIZES** : Ce dictionnaire contient les largeurs réelles (en centimètres) des objets courants. Cela permet de comparer la largeur apparente détectée dans l'image à la largeur réelle pour estimer la distance.

## 2 - Initialisation et préparation de l'image

La fonction reçoit une image (frame), le modèle YOLO chargé (net), les noms des couches de sortie (ln) et les labels des classes (LABELS). Elle détermine les dimensions de l'image (hauteur H et largeur W), et crée un blob à partir de l'image pour le traitement par le modèle YOLO.

## 3 - Passage du blob à travers le réseau

Un "blob" (Binary Large Object) dans le contexte de la vision par ordinateur et de l'utilisation de réseaux de neurones convolutifs (CNN) est une représentation binaire de l'image, transformée pour être utilisée comme entrée dans le réseau de neurones.

Le blob est passé à travers le réseau avec `net.setInput(blob)` et `net.forward(ln)` pour obtenir les sorties des couches.

## 4 - Parcours des détections

Pour chaque détection, les scores de confiance pour chaque classe sont extraits.

## 5 - Suppression des boîtes non maximales

La suppression des non-maxima est appliquée pour éliminer les boîtes redondantes avec `cv2.dnn.NMSBoxes`.

## 6 - Calcul de la distance et calibrage de la Distance Focale

Pour chaque boîte retenue après suppression des non-maxima, la distance de l'objet à la caméra est calculée.

On utilise la distance focale dans nos calculs, nous devons la calibrer pour notre caméra spécifique. Voici un aperçu du processus de calibrage :

1. **Mesurez une Distance Réelle :** Placez un objet de taille connue à une distance fixe de la caméra.
2. **Capturez une Image :** Prenez une photo de l'objet avec la caméra.
3. **Mesurez la Taille en Pixels :** Mesurez la largeur de l'objet dans l'image en pixels.

4. **Calculez la Distance focale en Pixels** : Utilisez la formule suivante pour calculer la distance focale en pixels :

$$f = \frac{P \times D}{W}$$

Où :

- f est la distance focale en pixels.
- P est la largeur de l'objet en pixels dans l'image.
- D est la distance réelle entre l'objet et la caméra.
- W est la largeur réelle de l'objet.

Dans notre code, la distance focale est utilisée pour calculer la distance réelle d'un objet détecté par rapport à la caméra :

$$distance\_cm = \frac{REAL\_WIDTH \times FOCAL\_LENGTH}{detected\_width}$$

Où :

- REAL\_WIDTH est la largeur réelle de l'objet en centimètres.
- FOCAL\_LENGTH est la distance focale en pixels.
- detected\_width est la largeur de l'objet détectée dans l'image en pixels.

En utilisant cette formule, nous pouvons estimer la distance de l'objet par rapport à la caméra en fonction de sa taille apparente dans l'image et de la distance focale calibrée.

Les objets détectés leur distance estimée, et leur position horizontale sont ajoutés à la liste detected\_objects.

## 7 - Retour des objets détectés

La fonction retourne une liste des objets détectés avec leur label, la distance estimée, et la position horizontale.

Cette fonction permet donc de détecter des objets dans une image et d'estimer leur distance par rapport à la caméra en utilisant un modèle YOLO pré-entraîné.

### Partie 5 : Boucle principale

Dans cette partie du code, nous avons la boucle principale qui gère la capture continue de la vidéo à partir de la caméra, la détection des objets dans chaque frame capturée, l'estimation de la distance des objets détectés, et la génération de feedback audio en fonction des objets détectés.

- Initialisation du flux vidéo : Cela initialise un objet VideoStream qui commence à capturer la vidéo à partir de la caméra

```
vs = VideoStream(src=0).start()
```

- Attend 2 secondes pour que la caméra se stabilise après son initialisation.

```
time.sleep(2.0)
```

- Initialise un compteur pour suivre le nombre de frames traitées.

```
frame_count = 0
```

- Boucle principale de traitement de la vidéo.

```
while True:  
    frame = vs.read()  
    frame_count += 1  
    frame = cv2.flip(frame, 1)
```

- Affiche le frame capturé dans une fenêtre nommée 'video'.

```
cv2.imshow('video', frame)
```

- Quitte la boucle si la touche 'q' est pressée dans la fenêtre affichant la vidéo.

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

- Traitement périodique basé sur le nombre de frames.

```
if frame_count % 400 == 0:  
    detected_objects = detect_objects_and_estimate_distance(frame, net, ln, LABELS)  
    give_audio_feedback(detected_objects, frame.shape[1])
```

- Arrête le flux vidéo de la caméra et ferme toutes les fenêtres affichées.

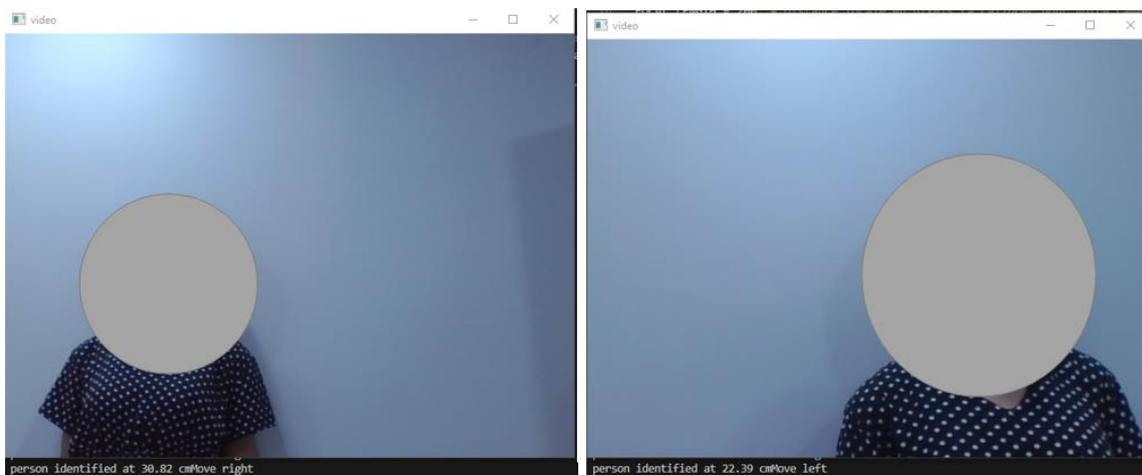
```
vs.stop()  
cv2.destroyAllWindows()
```

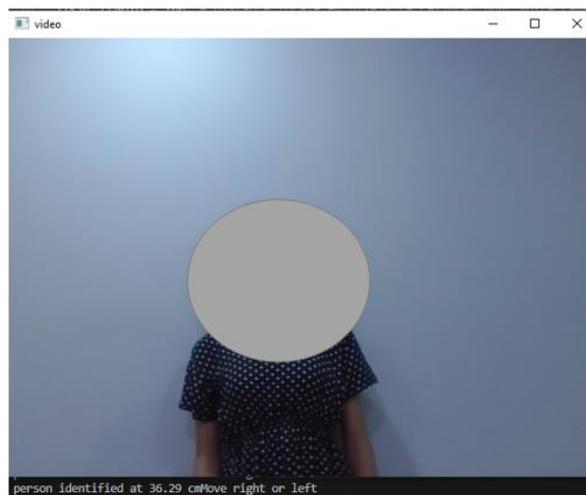
Cette partie du code gère la capture continue de la vidéo à partir de la caméra, effectue la détection d'objets tous les 400 frames, estime la distance des objets détectés, et fournit un feedback audio en conséquence. Elle s'assure également de terminer proprement le flux vidéo et de fermer toutes les fenêtres d'affichage à la fin de l'exécution.

### III.4 Tests et validation

La commande affiche les résultats de la détection de personnes dans l'image s'affichent. Les résultats indiquent que trois personnes ont été détectées dans l'image et que leur emplacement et leur distance ont été estimés. Lorsque la personne est au centre ou en face de la caméra, le malvoyant a le choix de se diriger vers la droite ou vers la gauche et lorsqu'elle est à gauche de la caméra, il se dirige vers la droite et lorsqu'elle est à droite, il se dirige vers la gauche pour éviter une collision.

```
person identified at 16.85 cmMove right or left  
person identified at 26.87 cmMove left  
person identified at 26.09 cmMove right
```



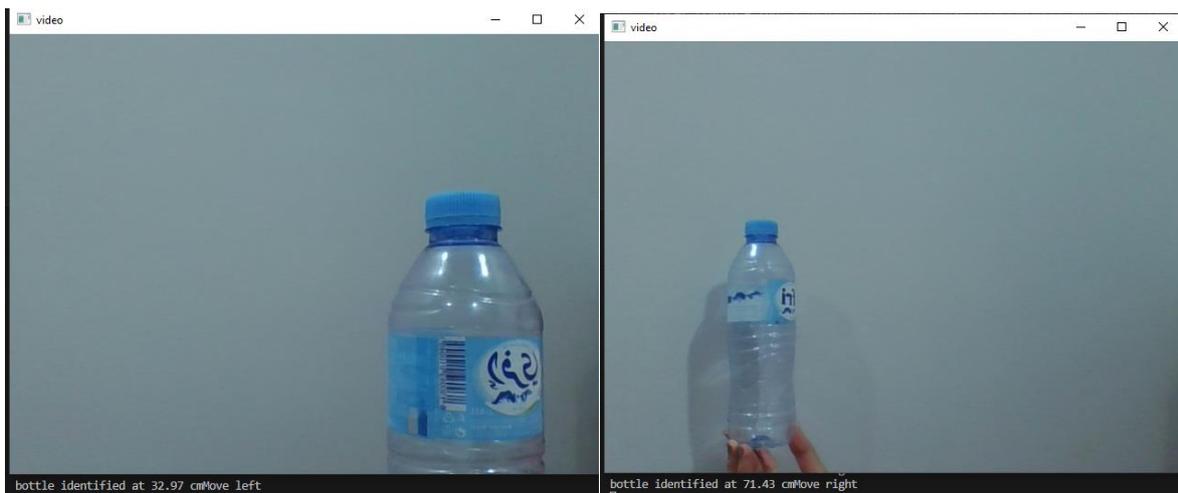


Cette commande nous montre clairement la mesure de la distance. La détection de la distance des objets est basée sur la taille apparente de l'objet détecté par YOLO et la distance focale de la caméra, nous avons défini un dictionnaire `OBJECT_SIZES` qui associe des étiquettes d'objets à leurs tailles réelles en centimètres. Par exemple pour l'étiquette "personne", la taille est définie comme 45 cm. Actuellement le seuil de confiance défini à 0.5, cela contrôle la sensibilité de la détection. Nous pouvons ajuster ce seuil pour changer la manière dont les objets sont détectés en fonction de leur distance.

**Détection des objets proches :** Pour les objets situés à une distance relativement proche de la caméra, un seuil de 0.5 est généralement adéquat. Cela signifie que YOLO ne prendra en compte que les détections où il est assez sûr (à 50% ou plus) que l'objet est présent. Les objets proches sont généralement plus grands dans le cadre de la caméra, ce qui facilite une détection précise avec ce seuil.

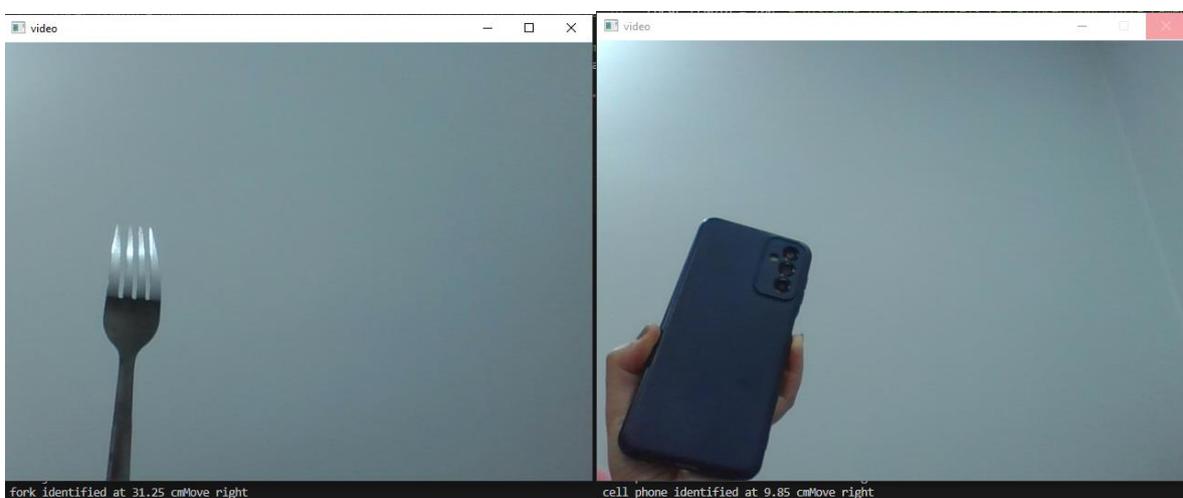
**Détection des objets éloignés :** À mesure que la distance entre la caméra et l'objet augmente, les objets apparaissent plus petits et peuvent être moins distincts dans l'image. Un seuil de 0.5 peut entraîner une diminution du nombre de détections d'objets éloignés, car la confiance dans la détection peut diminuer avec la distance.

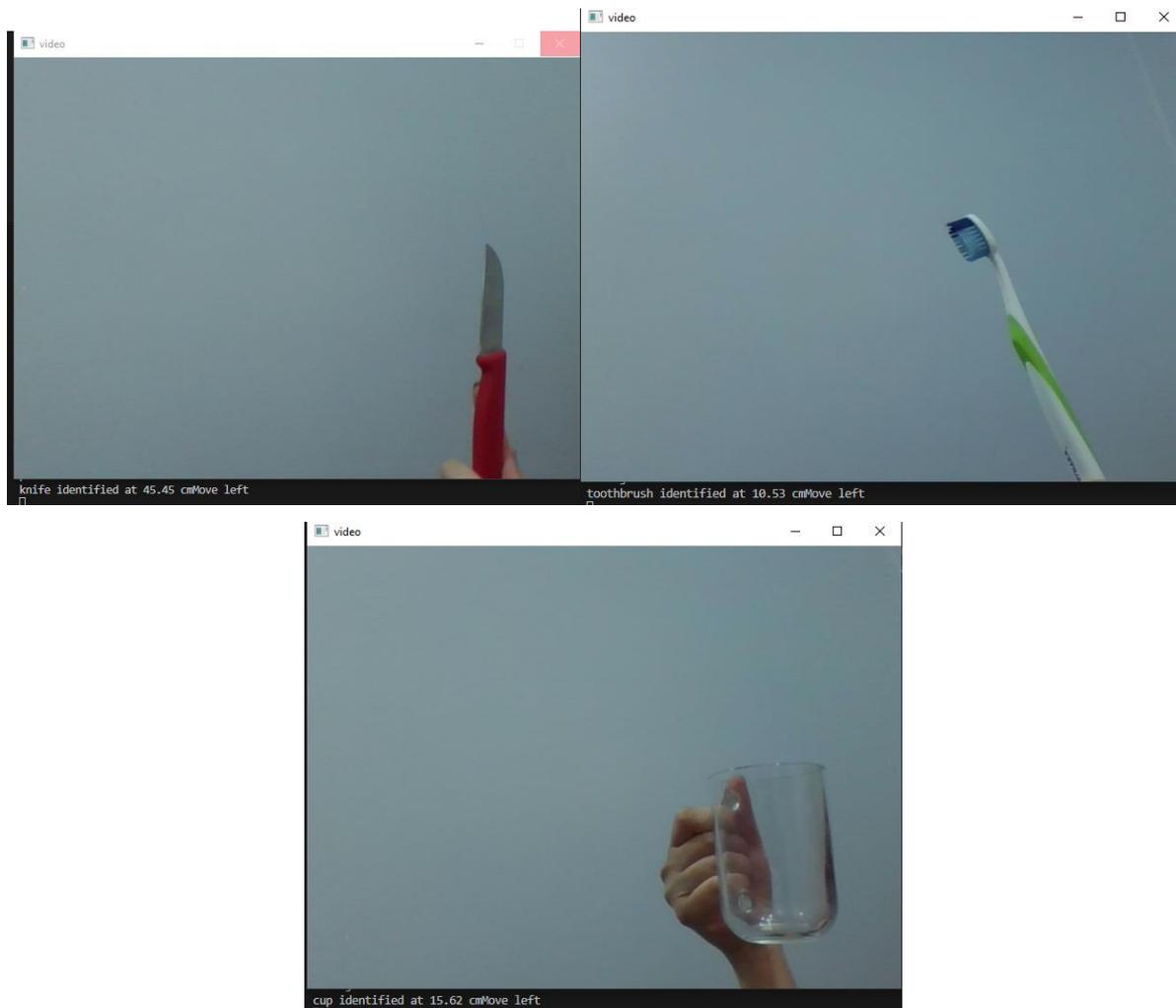
Un seuil de 0.5 aide à limiter les faux positifs en ne considérant que les détections où la probabilité d'erreur est inférieure à 50%. Cela contribue à maintenir une précision raisonnable tout en minimisant les erreurs de détection.



Et celle-ci nous montre quelques d'autres objets que notre programme peut détecter pour aider les personnes malvoyantes dans leur vie quotidienne.

```
PS C:\Users\A> & C:/Users/A/anaconda3/python.exe c:/Users/A/Downloads/simple_detection.py
pygame 2.5.2 (SDL 2.28.3, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
[INFO] loading YOLO from disk...
[INFO] Output layer names: ['yolo_82', 'yolo_94', 'yolo_106']
No objects identified
No objects identified
No objects identified
toothbrush identified at 39.22 cmMove left
No objects identified
fork identified at 23.53 cmMove left
No objects identified
knife identified at 42.55 cmMove left
bottle identified at 29.56 cmMove left
No objects identified
cell phone identified at 9.39 cmMove left
```





Le programme complet et les résultats fonctionnent avec une caméra d'ordinateur, accompagnés d'un message vocal pour reconnaître les objets et arrêter le flux vidéo et fermer la fenêtre lorsque 'q' est pressé.

- ✚ Pour rendre ce programme pratique et utile pour les personnes malvoyantes, nous suggérons d'intégrer un Raspberry Pi et une caméra Pi. Cette solution permettra de transformer le programme en un dispositif portable pour l'aide à la vie quotidienne et l'indépendance. En intégrant ces technologies sur des lunettes, cela offrira une assistance visuelle directement accessible et facile à utiliser dans diverses situations.

## **Conclusion**

En conclusion de ce chapitre, ce programme utilisant une caméra et une sortie vocale représente une avancée significative pour aider les personnes malvoyantes dans leur vie quotidienne. Il permet l'identification des objets essentiels sans se préoccuper des détails trop éloignés, répondant ainsi efficacement aux besoins spécifiques de cette communauté.

# ***CONCLUSION GENERALE***

## Conclusion Général

Dans le cadre de ce projet de fin d'études, nous avons exploré les systèmes embarqués pour la détection des objets et la mesure de distance, en utilisant une caméra Raspberry Pi. À travers notre recherche, nous avons pu constater les avancées significatives apportées par l'intelligence artificielle dans ce domaine. En intégrant des algorithmes de vision par ordinateur et des techniques de traitement d'image avancées, en intégrant l'algorithme de détection d'objets YOLO (You Only Look Once) et des techniques de traitement d'image avancées, notre système a démontré sa capacité à identifier avec précision plus de 50 objets essentiels pour les personnes malvoyantes. Pour l'estimation de la distance, nous avons utilisé des méthodes basées sur la reconnaissance des tailles des objets et la distance focale de la caméra avec une intégration d'une sortie vocale permet une interaction intuitive, facilitant ainsi la navigation quotidienne des utilisateurs. En résumé, ce projet illustre comment la technologie peut être utilisée de manière innovante pour améliorer la qualité de vie des individus en situation de handicap visuel, ouvrant la voie à de futures applications prometteuses dans le domaine de l'assistance assistée par l'IA.

## *Bibliographie*

## Bibliographie

- [1] « Cécité et déficience visuelle ». Consulté le: 25 juin 2024. [En ligne]. Disponible sur: <https://www.who.int/fr/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- [2] OphthalAdmin, « Troubles visuels et maladies », Ophtalmocéan. Consulté le: 24 mai 2024. [En ligne]. Disponible sur: <https://ophtalmocean.fr/troubles-visuels-et-maladies/>
- [3] « Institut Bordelais de la Vision, centre ophtalmologique bordeaux », Institut Bordelais de la Vision, centre ophtalmologique bordeaux. Consulté le: 20 mai 2024. [En ligne]. Disponible sur: [https://www.ibvision.fr/systeme\\_visuel](https://www.ibvision.fr/systeme_visuel)
- [4] « Structure et fonction des yeux - Troubles oculaires », Manuels MSD pour le grand public. Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.msdmanuals.com/fr/accueil/troubles-oculaires/biologie-de-l'œil/structure-et-fonction-des-yeux>
- [5] « Le cristallin et ses fonctions - Presbytie et cataracte », <https://www.qualidoc.fr/>. Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.qualidoc.fr/specialites/ophtalmologie/cataracte/cristallin/>
- [6] « Nerf optique : définition, schéma ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://sante.journaldesfemmes.fr/fiches-anatomie-et-examens/2789397-nerf-optique-definition-schema/>
- [7] A. Brilhault, « Vision artificielle pour les non-voyants: une approche bio-inspirée pour la reconnaissance de formes ».
- [8] « S. A. DRURY, « Image interpretation in geology, Tome 1 », Chapman and Hall second Edition 1993. - Recherche Google ». Consulté le: 24 mai 2024
- [9] « Image numérique », *Wikipédia*. 19 mai 2024. Consulté le: 24 mai 2024. [En ligne]. Disponible sur: [https://fr.wikipedia.org/w/index.php?title=Image\\_num%C3%A9rique&oldid=215229962](https://fr.wikipedia.org/w/index.php?title=Image_num%C3%A9rique&oldid=215229962)
- [10] « A. KERIOUI et M. DJANI, « Application des réseaux de neurones à la compression d'images », mémoire de fin d'étude d'ingénieur, Editions 1998. - Recherche Google ». Consulté le: 24 mai 2024.
- [11] « A. MARION, « Introduction aux techniques de traitement d'images », Editions Eyrolles. Paris, 1987. - Recherche Google ». Consulté le: 24 mai 2024. [
- [12] A. Maoutsi, K. Harrar, S. Massout, et N. Bourkache, « Président Rapporteur Examineur Examineur ».

- [13] « G. BUREL, « Introduction au traitement d'images », Edition Hermès Octobre 2001. - Recherche Google ». Consulté le: 24 mai 2024.
- [14] L. Legrand, « Le traitement d'images médicales : Principes, principales applications et perspectives », 2013, p. 101-123. doi: 10.1007/978-2-8178-0338-8\_5.
- [15] « Delporte, J. (2014). Factorisation Matricielle, Application à la Recommandation Personnalisée de Préférences. Thèse de doctorat en Informatique, Institut National des Sciences Appliquées de Rouen - Recherche Google ». Consulté le: 26 mai 2024.
- [16] « Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 44(1.2), 206-226. - Recherche Google ». Consulté le: 26 mai 2024.
- [17] J. Robert, « Machine Learning : Définition, fonctionnement, utilisations », Formation Data Science | DataScientest.com. Consulté le: 26 mai 2024. [En ligne]. Disponible sur: <https://datascientest.com/machine-learning-tout-savoir>
- [18] « M. R. Amini. Apprentissage machine de la théorie à la pratique. Eyrolles, 2015. - Recherche Google ». Consulté le: 26 mai 2024.
- [19] « Glossaire - Intelligence artificielle - www.coe.int », Intelligence artificielle. Consulté le: 27 mai 2024. [En ligne]. Disponible sur: <https://www.coe.int/fr/web/artificial-intelligence/glossary>
- [20] « Supervisé vs non supervisé. <https://brightcape.co/apprentissage-supervise-vs-non-supervise/>. - Recherche Google ». Consulté le: 26 mai 2024.
- [21] « M. R. Coulon. Apprentissage par renforcement utilisant des réseaux de neurones, avec des application au contrôle moteur. PhD thesis, l'institut national polytechnique de GRENOBLE, 2002. - Recherche Google ». Consulté le: 26 mai 2024.
- [22] Celuga.fr, « Le cycle de vie d'un projet de Machine Learning en 8 étapes | Kaizen Solutions ». Consulté le: 26 mai 2024.
- [23] « Deep-Learning-pour-la-classification..pdf ». Consulté le: 2 juin 2024. [En ligne]. Disponible sur: <http://dspace.univ-tlemcen.dz/bitstream/112/12583/1/Deep-Learning-pour-la-classification..pdf>
- [24] « Machine Learning versus Deep Learning - Studytonight ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.studytonight.com/post/machine-learning-versus-deep-learning>
- [25] « Deep Learning et machine learning : différence entre les technologies de données – AWS », Amazon Web Services, Inc. Consulté le: 27 mai 2024. [En ligne]. Disponible sur:

<https://aws.amazon.com/fr/compare/the-difference-between-machine-learning-and-deep-learning/>

[26] « Qu'est-ce que la vision par ordinateur ? », Intel. Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.intel.com/content/www/fr/fr/internet-of-things/computer-vision/overview.html>

[27] Marie-Laurence, « Computer vision : tout connaître sur cette technique de l'IA ! », LePont. Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.lepont-learning.com/fr/computer-vision/>

[28] « Qu'est-ce qu'un réseau neuronal profond ? | Botpress Blog ». Consulté le: 27 mai 2024. [En ligne]. Disponible sur: <https://botpress.com/fr/blog/reseau-neuronal-profond>

[29] « Qu'est-ce que le Deep Learning ? », ActuIA. Consulté le: 27 mai 2024. [En ligne]. Disponible sur: <https://www.actuia.com/actualite/quest-deep-learning/>

[30] « Que sont les réseaux neuronaux ? | IBM ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.ibm.com/fr-fr/topics/neural-networks>

[31] « Djerioui.M, « Contribution au Développement de Systèmes Multicapteurs Intelligents Dédiés à la Surveillance et au Contrôle de la Qualité des Eaux Propres », Thèse de Doctorat en électronique, Université de M'sila, 2019. - Recherche Google ». Consulté le: 26 mai 2024.

[32] N. Humblot, « Deep learning : comprendre les réseaux de neurones artificiels (artificial neural networks) », Ryax Technologies. Consulté le: 27 mai 2024. [En ligne]. Disponible sur: <https://ryax.tech/fr/deep-learning-comprendre-les-reseaux-de-neurones-artificiels-artificial-neural-networks/>

[33] Y. Ben Atia, « Deep Auto-Encodeur pour la Reconnaissance de Visage », juin 2019, Consulté le: 2 juin 2024. [En ligne]. Disponible sur: <http://archives.univ-biskra.dz:80/handle/123456789/13187>

[34] G. L. Team, « Types of Neural Networks and Definition of Neural Network », Great Learning Blog: Free Resources what Matters to shape your Career! Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.mygreatlearning.com/blog/types-of-neural-networks/>

[35] « K. Kim et al. « Network Intrusion Detection using Deep Learning» page 32,2018 - Recherche Google ». Consulté le: 28 mai 2024.

[36] « 3165643223752092.pdf ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://bucket.theses-algerie.com/files/repositories-dz/3165643223752092.pdf>

[37] « 2489250461957366.pdf ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://bucket.theses-algerie.com/files/repositories-dz/2489250461957366.pdf>

- [38] « Trad Houssein Eddine «La détection d'objet avec OpenCV et deep learning» Mémoire De Master Sciences et Technologies Electronique Réseaux Télécommunication, Université Mohamed Khider de Biskra - Recherche Google ». Consulté le: 28 mai 2024. [En ligne].
- [39] « Oulmi Mehdi et Kaloune Salim «Classification d'objets avec le Deep Learning» Mémoire de Master En Informatique, Université Akli Mohand Oulhadj de Bouira, 2018 - Recherche Google ». Consulté le: 28 mai 2024. [En ligne].
- [40] « Couche de Convolution: Tout savoir sur les CNN - Meilleur Guide ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://inside-machinelearning.com/cnn-couche-de-convolution/>
- [41] « MLK«6 Different Types of Object Detection Algorithms in Nutshell», <https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms> - Recherche Google ». Consulté le: 28 mai 2024.
- [42] G. L. Team, « YOLO object detection using OpenCV », Great Learning Blog: Free Resources what Matters to shape your Career! Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://www.mygreatlearning.com/blog/yolo-object-detection-using-opencv/>
- [43] « J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. «You only look once: Unified, realtime object detection». Preprint sur arXiv:1506.02640, 2015 - Recherche Google ». Consulté le: 28 mai 2024.
- [44] « CS 230 - Pense-bête de réseaux de neurones convolutionnels ». Consulté le: 28 mai 2024. [En ligne]. Disponible sur: <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>
- [45] « Raspberry Pi : Notre guide complet pour débiter », Raspberry Pi France. Consulté le: 2 juin 2024. [En ligne]. Disponible sur: <https://www.raspberrypi-france.fr/guide/>
- [46] « Page non trouvée | iTechnoFrance ». Consulté le: 2 juin 2024. [En ligne]. Disponible sur: <https://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-aveclarduino/>
- [47] G. Tronic, « dans les configurations »:, 2017.