

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

Université Abderrahmane Mira Bejaia  
Faculté Des Sciences Exactes  
Département d'Informatique

Mémoire de fin d'études en vue de l'obtention du diplôme de Master  
en Intelligence Artificielle

---

Thème  
Dispositif Intelligent pour l'aide à la navigation  
des personnes aveugles et malvoyantes

---

*Jury :*

Présidente DR ALOUI SORAYA NÉE  
TIGHIDET

Examineur Dr ATMANI  
MOULOUD

Examineur Dr BOUCHEBBAH  
FATAH

*Réalisé par :*

Ramy BOUTEGRABET  
Maya KHALED KHODJA

*Encadré par :*

Ahror BELAID

12 juillet 2023

# *Remerciements*

*Nous tenons à exprimer notre profonde gratitude envers le Bon Dieu qui nous a donné la force nécessaire pour réaliser ce modeste travail ainsi qu'à toutes les personnes ayant contribué de près ou de loin à la réalisation de ce mémoire.*

*Tout d'abord, nous tenons à remercier notre encadrant Monsieur BELAID Ahror de nous avoir orienté et aidé à mener à bien ce projet. Nous lui exprimons toute notre reconnaissance pour son soutien constant, ses conseils précieux et son expertise. Sa disponibilité et sa passion pour la recherche nous ont inspirés et encouragés à repousser nos limites.*

*Nous sommes également reconnaissants envers les membres du jury, Madame. ALOUI Soraya, Monsieur. ATMANI Mouloud, Monsieur. BOUCHEBBAH Fatah, pour leur temps, leur attention et leurs commentaires constructifs qui ont grandement amélioré ce travail.*

*Nous souhaitons exprimer notre gratitude envers l'unité de recherche CERIST de Béjaïa pour leur confiance et pour nous avoir fourni les ressources nécessaires et pour toutes les connaissances et l'expérience que nous avons acquis grâce à eux. Pour cela, nous remercions également nos talentueux collègues du laboratoire, pour leur collaboration, leurs discussions stimulantes et leur aide précieuse. Leurs idées et leur expertise ont été inestimables dans la réalisation de ce mémoire.*

*Nous remercions nos très chers parents, amis et familles respectives pour leur soutien indéfectible tout au long de nos études. Leurs encouragements, leurs conseils et leur amour ont été une source de motivation et de réconfort dans lequel nous puisons notre énergie et notre force pour ne jamais abandonner face aux obstacles.*

*Enfin, nous sommes reconnaissants envers l'université A.Mira de Béjaïa pour avoir fourni les enseignements et les ressources nécessaires à la réalisation de cette recherche.*

*Nos sincères remerciements vont également à toutes les personnes que nous n'avons pas citées ici, mais qui ont joué un rôle important dans ce projet.*

## *\*Dédicaces\**

*Je voudrais commencer par remercier mes parents. Ma mère d'abord pour m'avoir inculqué une excellente éducation ainsi que pour avoir été une enseignante de la vie en plus d'être celle qui m'a tout appris ; de mes premiers mots à l'homme que je suis devenu aujourd'hui. Mon père, pour m'avoir offert une vie que peu aient eu la chance d'avoir. Je te remercie pour les sacrifices que tu as faits, j'espère que tu es fier de ce que je suis devenu. Maman, papa, merci d'avoir cru en moi, de m'avoir poussé et motivé à continuer, merci infiniment.*

*Je voudrais remercier ma famille, pour tout ce que vous avez fait pour moi depuis mon plus jeune âge et jusqu'à maintenant malgré les hauts et les bas de la vie. Un grand merci à mes grands-parents pour avoir été une source de bonheur, de nostalgie et de souvenirs.*

*Merci aux amis qui sont à mes côtés, vous serez à tous jamais gravés dans mon cœur non pas comme des amis, mais comme des frères et sœurs.*

*Merci aux enseignants qui ont su nourrir cette passion que j'ai en moi pour ce que je fais ainsi que ce que je souhaite accomplir.*

*Merci à notre encadrant de nous avoir donné cette opportunité, de nous avoir guidés, aidés ainsi que de nous avoir ouvert les portes du CERIST où nous avons pu rencontrer de merveilleuses personnes que je remercie pour leur aide et leur continuel soutien.*

*Enfin, merci à Allah de nous avoir ouvert toutes ces portes et de nous avoir aidé durant ce parcours. De nous avoir montré la voie afin de surmonter toutes les épreuves que nous avons pu rencontrer durant notre vie et pour ce qui nous attend*

*Une pensée pour toi Cyril. À tous jamais dans nos cœurs, où que tu sois, veille sur nous. J'aurais tellement aimé que tu puisses être encore à nos côtés.*

***M. Boutegrabet Ramy***

## *\*Dédicaces\**

*À mes parents, Merci pour votre amour indéfectible, votre soutien sans faille et vos encouragements constants pour faire de moi la personne que je suis aujourd'hui. Un grand merci à ma mère pour ses sacrifices et à ma soeur pour sa patience et son investissement personnel, je vous suis infiniment reconnaissante pour tout.*

*À ma nièce, Nelya merci pour le bonheur et la joie que tu m'apportes au quotidien.*

*À mon encadrant, merci infiniment pour votre confiance en moi et merci de m'avoir permis d'apprendre tant de choses au cours de cette expérience.*

*À mes amis, merci d'avoir toujours été là pour moi, me prodiguant vos conseils, partageant mes joies et me soutenant dans les moments difficiles. Votre présence a rendu ce voyage encore plus spécial, et je vous en suis reconnaissante.*

*À mes grands parents, et à tous mes proches pour chacune de vos petites attentions.*

***Mlle. Khaled Khodja Maya***

## Résumé

Le monde dans lequel nous vivons est peuplé d' $\frac{1}{4}$  de milliards de personnes atteintes de déficience visuelle. Ces personnes sont chaque jour confrontées à la difficulté de naviguer en sécurité dans un environnement extérieur.

Avec l'évolution exponentielle des techniques de vision par ordinateurs, il est désormais possible de croire à des solutions qui peuvent se servir de l'intelligence artificielle (IA) pour aider ces personnes à se déplacer plus aisément en améliorant ainsi leur quotidien et leur sécurité.

Aujourd'hui, le dilemme auquel ces personnes font face est soit de continuer à utiliser les aides à la navigation rustique telles que les chiens guides, les cannes ou autres moyens existants depuis lors, soit de faire confiance aux nouvelles technologies.

Ce dilemme nous offre donc un challenge qui est de mettre à bien ce projet intitulé "Outil intelligent d'aide à la navigation des personnes aveugles et malvoyantes" où nous avons mis au point un système intelligent embarqué qui permet à l'utilisateur de détecter et éviter les obstacles auxquels il pourrait être confronté.

Dans ce projet, nous avons entraîné des modèles de détection et de classification d'objets ainsi que des modèles de traitement du langage naturel en combinant l'IA, l'internet of things (IoT), et des techniques de vision par ordinateur, afin de réaliser un prototype capable d'assurer à l'utilisateur une navigation en temps réel et qui lui permet la détection d'objets/obstacle, la lecture d'une donnée textuelle et la description de scène.

## **ABSTRACT**

The world in which we live has one fourth of a billion persons suffering from visual deficiency. These persons are faced, every single day, with the difficulties of navigating in an outdoors environment in total safety.

With the exponential evolution of computer vision techniques, it is now possible to believe in solutions exploiting artificial intelligence (AI) for the purpose of helping people to get about in an easier manner and thus, improving their lifestyle and safety.

Today's dilemma to which these persons must face is to either keep using rustic navigation aids such as guide dogs, canes and other existing means or to start trusting the new technologies.

This dilemma offers us a challenge which is to conduct this project called "Smart tool for the navigation aid of the blind and the visually impaired" where we developed a portable smart device helping the user to detect and navigate through the obstacles which he could be confronted to.

In this project, we have trained several detection and classification models and other natural language processing models. We have combined both IA and internet of things (IoT) with computer vision techniques in order to create a prototype capable of real time navigation with detection of objects and obstacles, the reading of textual data in front of him and scene description.

# Table des matières

<b>Table des matières</b>	<b>I</b>
<b>Table des figures</b>	<b>III</b>
<b>Liste des tableaux</b>	<b>IV</b>
<b>Liste des abréviations</b>	<b>V</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Rappels et outils pour un système intelligent embarqué</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Les systèmes embarqués . . . . .	3
1.2.1 Systèmes intelligents embarqués . . . . .	3
1.3 Définitions et généralités . . . . .	3
1.4 Réseaux de neurones pour les systèmes embarqués . . . . .	5
1.4.1 Algorithmes de détection d'objets . . . . .	6
1.4.2 Modèles de réseaux de neurones pré-entraînés . . . . .	7
1.5 Définition d'un jeu de données (dataset) . . . . .	8
1.5.1 Dataset COCO . . . . .	8
1.6 Outils d'aide à la navigation . . . . .	9
1.6.1 Les types d'entrées sensorielles . . . . .	9
1.6.2 Les types de sorties sensorielles . . . . .	11
1.7 Quelques matériels utilisés pour l'aide à la navigation . . . . .	12
<b>2 État de l'Art</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Qu'es-ce que O et M? . . . . .	16
2.3 Revue des différents articles . . . . .	16
2.3.1 Systèmes à imagerie visuelle . . . . .	16
2.3.2 Systèmes de données non-visuelles . . . . .	20
2.3.3 Soutuion basée sur les smartphones . . . . .	23
2.3.4 Systèmes utilisant les méthodes d'apprentissage profond . . . . .	26
<b>3 Conception et prototypage</b>	<b>30</b>
3.1 Introduction . . . . .	30
3.2 <b>La contribution du ML dans un système intelligent embarqué</b> . . . . .	31
3.3 Cycle de vie d'un projet Machine Learning . . . . .	31
3.4 Comparaison des différents modèles étudiés . . . . .	32
3.4.1 Comparaison des modèles de réseaux de neurones utilisés . . . . .	32
3.4.2 Comparaison des modèles de NLP utilisés . . . . .	34
3.5 Définition des métriques de performance . . . . .	34
3.6 Plan du prototype à mettre en œuvre . . . . .	36
3.6.1 Besoins matériels . . . . .	36
3.6.2 Besoins Logiciels . . . . .	37
3.7 Méthode et implémentation de l'ultrason . . . . .	38
3.7.1 Les capteurs ultrason . . . . .	38
3.8 Manipulation des modèles . . . . .	41
3.8.1 MobileNet SSD V2 . . . . .	41
3.8.2 Google's Tesseract . . . . .	44

3.8.3	GPT-2 . . . . .	45
3.9	Bibliothèques utilisées . . . . .	46
3.10	Plan du montage du projet . . . . .	47
3.11	Fonctionnalités du prototype . . . . .	48
3.12	Diagramme de cas d'utilisation . . . . .	49
<b>4</b>	<b>Évaluation et tests</b>	<b>51</b>
4.1	Introduction . . . . .	51
4.2	Motivation sur l'usage de COCO . . . . .	52
4.3	Versions des logiciels . . . . .	52
4.4	Résultats Obtenus . . . . .	53
4.4.1	Détection . . . . .	53
4.4.2	Temps de réponse du système . . . . .	54
4.4.3	Coût matériel . . . . .	55
4.4.4	Coût Logiciel . . . . .	56
4.5	Discussion . . . . .	56
4.5.1	Avantages et limites de la méthode développée . . . . .	56
4.5.2	Discussion des futures améliorations . . . . .	57
	<b>Conclusion générale et perspectives</b>	<b>59</b>
	<b>Bibliographie</b>	<b>60</b>

# Table des figures

1.1	Exemple du dataset COCO [9]	9
1.2	Capteur infrarouge	10
1.3	Capteur ultrason	11
1.4	Raspberry Pi	12
1.5	Smart canne	13
1.6	Lunettes Intelligentes	13
1.7	Module Gps	14
2.1	Tyflo (Adapté de [5])	16
2.2	Binaural Rendering (Adapté de [38])	17
2.3	Réseau de caméras IP (adapté de [16])	17
2.4	VS-LAM	18
2.5	ISANA (Adapté de [4])	18
2.6	Microsoft Kinect	19
2.7	Participant utilisant le système LASS (Adapté de [37])	19
2.8	BLE Beacons (Adapté de [26])	21
2.9	Indriya (Adapté de [15])	21
2.10	Blind guide (Adapté de [39])	22
2.11	Google Tango	23
2.12	The sound of vision (Adapté de [6])	23
2.13	NavCog3 (Adapté de [31])	24
2.14	Percept 2 (Adapté de [11])	24
2.15	Envision (Adapté de [18])	26
2.16	A Deep learning based model to assist blind people in their navigation (Adapté de [29])	27
2.17	Smart Glass System Using Deep Learning for the Blind and Visually Impaired (Asapté de [32])	27
3.1	Cycle de vie d'un projet machine learning	32
3.2	Pont diviseur de tension.	38
3.3	Plan du montage du capteur ultrason	39
3.4	Architecture principale de MobileNet V2.	42
3.5	Entrée des tensors dans le réseau	42
3.6	Architecture du classificateur de MobileNet.	43
3.7	Architecture de la partie SSD de MobileNet	43
3.8	Architecture de Google's Tesseract.	44
3.9	Montage du prototype.	47
3.10	Entraînement du modèle.	47
3.11	Plan du fonctionnement du prototype	48
3.12	Diagramme de cas d'utilisation	49
4.1	Capture d'écran de l'interface utilisateur illustrant les résultats de la détection d'une classe	53
4.2	Capture d'écran de l'interface utilisateur illustrant les résultats de la détection de plus d'une classe	53
4.3	Autre exemple illustrant les résultats de la détection d'une classe	54

# Liste des tableaux

1.2	Tableau comparatif des différents outputs possible . . . . .	12
2.1	Tableau comparatif des systèmes à imagerie visuelle . . . . .	20
2.2	Tableau comparatif des systèmes données non-visuelles et systèmes basées sur les smartphones. . . . .	25
2.3	Tableau comparatif des systèmes utilisant les méthodes d'apprentissages profond . . . . .	28
3.1	Tableau comparatif des performances des modèles testés avec le modèle adopté. . . . .	33
3.2	Tableau comparatif des architectures et performances des modèles NLP testé ainsi que du modèle adopté. . . . .	34
4.1	Tableau comparatif des différents logiciels utilisés . . . . .	52
4.2	Tableau résumant le temps de réponse de chaque processus au lancement du système avant optimisation de l'algorithme . . . . .	54
4.3	Tableau résumant le temps de réponse de chaque processus au lancement du système après optimisation de l'algorithme . . . . .	55
4.4	Tableau récapitulatif du coût matériel . . . . .	56

# Liste des abréviations

<b>AIoT</b>	<i>Artificial Intelligence Of Things</i>
<b>BERT</b>	<i>Bidirectional Encoder Representation from Transformers</i>
<b>CNN</b>	<i>Convolutional Neural Network</i>
<b>COCO</b>	<i>Common Objects in COntext</i>
<b>DNN</b>	<i>Deep Neural Network</i>
<b>EOA's</b>	<i>Electronic Orientation Aids</i>
<b>ETA's</b>	<i>Electronic Travel Aids</i>
<b>FPS</b>	<i>Frames per second</i>
<b>Gnd</b>	<i>Ground</i>
<b>GRU</b>	<i>Gated Recurrent Units</i>
<b>GPT</b>	<i>Generative Pre-Trained Transformer</i>
<b>GPS</b>	<i>Global Positioning Position</i>
<b>HC-SR04</b>	<i>High-Conductance Ultrasonic Sensor</i>
<b>IA</b>	<i>Intelligence Artificielle</i>
<b>IoT</b>	<i>Internet of Things</i>
<b>IR</b>	<i>Infrarouge</i>
<b>LSTM</b>	<i>Long Short-Term Memory</i>
<b>OCR</b>	<i>Optical character Recognition</i>
<b>ML</b>	<i>Machine Learning</i>
<b>NLP</b>	<i>Natural Language Processing</i>
<b>NFC</b>	<i>Near-Field Communication</i>
<b>PLDs</b>	<i>Position Locator Devices</i>
<b>QA</b>	<i>Question Answering</i>
<b>RNN</b>	<i>Réseaux de neurones récurrents</i>
<b>R-CNN</b>	<i>Region-based Convolutional Neural Network</i>
<b>ResNet</b>	<i>Residual Neutral Network</i>
<b>RFID</b>	<i>Radio Frequency IDentification</i>
<b>SSD</b>	<i>Single Shot Detector</i>
<b>VIN</b>	<i>INput Voltage</i>
<b>VOUT</b>	<i>OUTput Voltage</i>
<b>YOLO</b>	<i>You Only Look Once</i>

# Introduction générale

Le monde dans lequel nous vivons actuellement ne cesse de connaître des changements constants et réguliers, que ce soit en termes d'infrastructures ou de technologies, et avoir une image statique d'un environnement externe est devenu quasi impossible.

Être une personne aveugle ou malvoyante dans un environnement pareil devient rapidement un défi très pesant, surtout avec l'expansion continue de ce dernier, à savoir l'ajout de nouveaux obstacles perturbant la navigation en extérieur. Cela est un réel problème pour des millions de personnes à travers le monde pour qui, il est de plus en plus difficile de se déplacer en sécurité.

En 2022, l'organisation mondiale de la santé a estimé qu'environ 2.2 milliards de personnes souffrent de problèmes de la vision [41]. Des méthodes traditionnelles comme les cannes et les chiens guides ont été très utiles et ont prouvé leur efficacité afin de subvenir aux besoins des malades, néanmoins elles se sont avérées très limitées.

L'avancée technologique continue que connaît notre époque, et notamment avec le développement du secteur informatique et électronique, permet d'offrir de nouvelles possibilités de développement de l'assistance à la navigation.

Bien qu'il existe déjà plusieurs méthodes conçues pour l'aide à la navigation des personnes aveugles et malvoyantes, leur capacité à fournir des informations précises sur la localisation en temps réel est limitée, certaines méthodes sont aussi jugées rustiques avec l'évolution de cette ère technologique.

Nous visons, par cette étude, la conception d'une technologie d'assistance intelligente, portable et d'actualité qui pourrait offrir une meilleure qualité de vie aux individus sujets à un déficit de la vision. Le système proposé est constitué d'algorithmes d'apprentissage automatique et de techniques de vision par ordinateur pour la localisation et l'identification des objets en temps réel, le tout combiné dans une technologie intelligente qui servira de dispositif d'aide à la navigation des personnes ayant une acuité visuelle réduite.

Dans ce mémoire, nous allons parler de la contribution d'une combinaison d'intelligence artificielle ainsi que d'internet des objets dans le but d'aider les personnes avec une capacité visuelle limitée à naviguer en temps réel, nous définirons ce qu'est un système intelligent embarqué et ses outils. Nous discuterons également des méthodes déjà exploitées et publiées afin de réaliser une étude comparative et critique en comparant les performances des méthodes traditionnelles aux méthodes récentes. Nous parlerons ensuite de notre solution ainsi que des besoins matériels et logiciels nécessaires à sa conception. Enfin, nous exposerons les résultats obtenus et nous listerons les futures perspectives.

# Chapitre 1

## Rappels et outils pour un système intelligent embarqué

### 1.1 Introduction

La liberté d'observer et de se déplacer font partie des droits fondamentaux de la vie. La conception d'un système intelligent embarqué pour aider des personnes atteintes de cécité visuelle nécessite l'intégration de plusieurs technologies telles que l'intelligence artificielle (IA), les capteurs, l'informatique embarquée, etc.

Dans ce chapitre, nous commencerons par introduire les systèmes intelligents embarqués, l'IA, et son application dans l'IoT (internet of things). Nous définirons également les différents matériels, logiciels et plusieurs concepts essentiels pour le suivi de ce mémoire.

## 1.2 Les systèmes embarqués

Un système embarqué est un système à la fois électronique et informatique. Sa conception combine à la fois le matériel et le logiciel afin qu'il puisse réaliser des tâches spécifiques. Ce système fonctionne de manière automatique et autonome. La plupart des systèmes embarqués se caractérisent également par leur faible consommation énergétique et leur petite taille. Avec l'avènement de l'intelligence artificielle, les systèmes embarqués intelligents ont un potentiel infini.

### 1.2.1 Systèmes intelligents embarqués

Un système intelligent embarqué est un système électronique conçu pour effectuer des tâches spécifiques de manière autonome et intelligente. Il est intégré à un appareil ou à une machine et combine ainsi les domaines de l'intelligence artificielle et de l'électronique. Tout cela permet d'intégrer des capacités de traitement de données, d'analyse et de prise de décision en temps réel.

Les systèmes intelligents embarqués sont souvent utilisés dans des applications mobiles, et on les retrouve dans divers domaines tels que la téléphonie, l'automobile, la médecine et bien d'autres encore.

## 1.3 Définitions et généralités

### 1. Définition de l'intelligence artificielle

L'intelligence artificielle (IA) [8] est un domaine de la technologie qui vise à créer des systèmes informatiques capables de réaliser des tâches qui nécessitent normalement l'intelligence humaine, que ce soit vision par ordinateur, reconnaissance vocale ou encore traduction de langues et même prise de décision. Utilisant des algorithmes sophistiqués afin de traiter de grandes quantités de données et identifier des modèles cachés permettant aux machines d'apprendre de manière autonome tout en améliorant leur performance au fil du temps. On retrouve l'IA dans de nombreux domaines tels que la santé, les finances, la sécurité et l'industrie afin d'avoir des résultats plus efficaces, précis et de qualité supérieure.

### 2. Définition de l'IIoT

L'IIoT (Internet des objets) est un réseau de dispositifs physiques connectés à Internet pouvant contenir des capteurs, appareils ménagers et des machines capables de collecter et échanger les données en temps réel. Il permet de surveiller l'efficacité opérationnelle, d'améliorer les expériences utilisateur ainsi que d'automatiser les processus. On retrouve son application dans de nombreux secteurs, mais nécessite également de relever des défis de sécurité des données ainsi que de la gestion de la complexité réseau.

### 3. Définition de l'AIoT

L'AIoT (Artificial Intelligence of Things) est la combinaison de l'IIoT et de l'IA pour créer des systèmes intelligents et connectés. Cette combinaison donne la capacité aux dispositifs IIoT de faire la collecte et l'analyse des données massives en temps réel, ce qui permet à des modèles de l'IA de prendre des décisions intelligentes sans intervention humaine. L'AIoT peut améliorer l'efficacité, la sécurité et l'expérience utilisateur dans différents secteurs, mais nécessite une attention particulière pour la sécurité des données et la protection de la vie privée des utilisateurs.

### 4. Réseaux de neurones artificiels

Les réseaux de neurones artificiels [27] sont des modèles computationnels inspirés du fonctionnement du système nerveux humain. Utilisés en intelligence artificielle dans le but de résoudre des tâches complexes telles que la classification, la prédiction ou encore la reconnaissance de patterns en apprenant à partir des données. Ces derniers sont constitués

de nœuds interconnectés nommés neurones. Traitant les entrées et générant des sorties en fonction de poids et de fonctions d'activation.

## 5. Machine Learning

Le Machine Learning ou encore apprentissage automatique est une branche de l'intelligence artificielle se concentrant sur le développement de techniques donnant la capacité aux ordinateurs d'apprendre à partir des données. Reposant sur des algorithmes et des modèles mathématiques permettant aux systèmes informatiques d'extraire des connaissances et leur donner la capacité de prendre des décisions en analysant de grandes quantités de données.

## 6. Deep Learning

Le Deep Learning, ou l'apprentissage profond, est une sous-branche du machine learning se concentrant sur l'apprentissage automatique de modèles de données complexes à l'aide de réseaux de neurones artificiels. Utilisant des architectures de réseaux de neurones avec plusieurs couches afin d'extraire les représentations hiérarchiques des données, il permet ainsi de modéliser les relations non linéaires et l'apprentissage de caractéristiques abstraites. Avec l'obtention de performances remarquables, le deep learning est utilisé dans divers domaines tels que la vision par ordinateur, le traitement de langage naturel et beaucoup d'autres. Tout cela ouvre de nouvelles perspectives dans le domaine de l'intelligence artificielle.

## 7. Transfert Learning

Le transfert d'apprentissage (ou transfer learning en anglais) est une technique en apprentissage automatique qui consiste à utiliser les connaissances acquises par un modèle préalablement entraîné sur une tâche source pour améliorer les performances d'un modèle sur une tâche cible similaire. Au lieu de construire un modèle à partir de zéro et de l'entraîner entièrement sur la tâche cible, le transfert d'apprentissage permet de bénéficier des représentations apprises par le modèle source, qui peuvent être généralisables et utiles pour la tâche cible. Cela permet de réduire la nécessité de grandes quantités de données d'entraînement et de temps de calcul, tout en améliorant souvent les performances du modèle sur la tâche cible.

## 8. Réseaux de neurones convolutifs

Les CNN ou réseaux de neurones convolutionnels sont un type spécifique de réseau de neurones largement utilisé dans la vision par ordinateur. Ils exploitent la structure spatiale des données d'entrée, pouvant être des images par exemple, en utilisant des opérations de convolution et de pooling. Ayant la capacité d'extraire des caractéristiques visuelles hiérarchiques à partir d'images, il devient très efficace de faire appel à eux dans la détection ou reconnaissance d'objets et même la segmentation d'images.

## 9. Réseaux de neurones récurrents

Les Réseaux de neurones récurrents (ou Recurrent Neural Networks RNN) sont des modèles conçus pour comprendre et traiter des séquences de données textuelles, ils sont adaptés pour le traitement du langage naturel.

Les RNN sont des outils puissants capables de capturer des connexions complexes et à long terme entre les mots. Des versions améliorées telles que les LSTM (Long Short-Term Memory) et les GRU (Gated Recurrent Units) ont une mémoire spéciale pour retenir des informations importantes à plus longue échéance, ce qui leur permet de mieux saisir les susceptibilités du langage.

## 10. Traitement du langage naturel

Le Traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle qui vise à permettre aux machines de comprendre et de traiter le langage humain de manière efficace. Il utilise des techniques telles que l'analyse syntaxique, la sémantique et la reconnaissance vocale pour analyser et interpréter les phrases et les textes. Pour extraire des informations et faciliter la communication entre les humains et les machines. Il utilise des

modèles statistiques et d'apprentissage automatique. On peut citer de nombreuses applications pratiques qui nécessitent un NLP, telles que les assistants virtuels et l'analyse des sentiments.

### 11. Transformers

Les Transformers sont un type de modèle de réseau de neurones qui ont révolutionné le domaine du NLP. Introduits par le modèle « Attention Is All You Need », ils utilisent des mécanismes d'attention pour comprendre les relations entre les mots dans une phrase afin de modéliser les dépendances à long terme. Grâce à cette approche, ils ont été largement adoptés dans plusieurs domaines tels que Les chatbots et assistants virtuels et la traduction automatique.

### 12. OCR(Optical Chracter Recognition)

Technique qui permet la conversion des images/documents contenant des données textuelles en texte numérique modifiable. Cette méthode utilise des algorithmes pour analyser les images et détecter les caractères afin de les transcrire en texte.

l'OCR est utilisé pour la transformation du contenu imprimé ou manuscrit en text éditable pour faciliter l'accès et le traitement des informations, il est utilisé dans les tâches telles que la reconnaissance des plaques d'immatriculations, l'extraction des informations à partir d'un livre en format papier ou électronique et d'autres tâches similaires.

### 13. Modèle de langages pré-entraînés

Un modèle de langage pré-entraîné est un modèle de réseau de neurones entraîné sur de larges bases de données textuelles dans le but d'apprendre les structures linguistiques et les modèles de langage. Ces modèles sont utilisés comme point de départ pour résoudre différentes tâches, telles que la traduction automatique ou la génération de texte. L'utilisation de modèles pré-entraînés permet de bénéficier des connaissances linguistiques acquises lors de l'entraînement initial, l'entraînement sur des jeux de données plus restreint leur permet d'améliorer leur performance dans la réalisation de tâches spécifiques.

### 14. Modèle de génération de langage

Un modèle de génération de langage est un système d'intelligence artificielle capable de produire du texte de manière autonome. Il est entraîné sur une quantité massive de données textuelles. En utilisant des techniques d'apprentissage profond, il apprend à générer des phrases ou des paragraphes qui ont une structure et une signification similaires à celles du langage humain. Le modèle est capable de créer du contenu cohérent et pertinent en se basant sur un contexte donné, il peut être utilisé dans diverses applications telles que la rédaction automatique, la traduction automatique, la génération de dialogues ou la création de contenu artistique.

## 1.4 Réseaux de neurones pour les systèmes embarqués

Un modèle de réseaux de neurones est une configuration précise du réseau de neurones qui définit la connexion et l'organisation des neurones dans la réalisation d'une tâche spécifique, le modèle est construit sur une architecture appropriée.

Un modèle de réseau de neurones peut être de type réseaux convolutionnel ou récurrent, il est possible de combiner ces deux types ou de les utiliser séparément. Pour effectuer des tâches telles que la prédiction, la détection ou la génération de texte, un modèle est entraîné sur un jeu de données de type visuel pour les modèles convolutionnel et textuel pour les modèles récurrents.

Voici les principaux modèles de réseaux de neurones qui vont apparaître par la suite dans ce rapport :

### 1. MobileNet [1]

MobileNet est une architecture de réseau de neurones convolutifs (CNN) développée pour l'usage sur des appareils mobiles et autres appareils à faible puissance de calcul, tout comme les smartphones, tablettes et systèmes embarqués. Cette architecture est souvent utilisée pour des tâches telles que la classification d'images, détection d'objets et la segmentation sémantique. On retrouve plusieurs versions de MobileNet, notamment MobileNet V1, V2 ou V3 qui diffèrent par leurs architectures et leur performance.

MobileNetV2 est très similaire à MobileNet V1 la différence réside dans le fait qu'il utilise des blocs résiduels inversés avec des caractéristiques de goulot d'étranglement. Il a un nombre de paramètres nettement inférieur à celui de MobileNet V1. Nous parlons plus en détails de l'aspect architectural de ces CNN dans le chapitre 3.

Les versions de MobileNets prennent en charge toute taille d'entrée supérieure à 32 x 32, et les tailles d'image plus grandes offrant de meilleures performances. Ce qui fait d'eux un excellent choix pour les systèmes embarqués.

## 2. SSD MobileNetV2 [40]

MobileNet SSD V2 est une architecture de réseau de neurones convolutifs conçue spécialement pour la détection d'objets en temps réel sur des dispositifs mobiles. Se basant sur deux technologies ; MobileNetV2 et SSD (Single Shot Detector). MobileNet V2 permet une classification rapide et précise des images sur des appareils mobiles grâce à un réseau de neurones convolutifs léger. D'un autre côté, SSD est un algorithme de détection d'objets capable de détecter plusieurs objets dans une image en une seule passe de réseau <sup>1</sup>.

<sup>1</sup>. En combinant ces deux technologies, on obtient la capacité de réaliser une détection d'objets en temps réel avec une précision élevée tout en ayant une consommation d'énergie basse et diminuer l'utilisation de la mémoire. Tout cela permet au modèle d'être utilisé dans de nombreuses applications de détection d'objets, que ça soit sur smartphones ou tout autre dispositif mobile pour la reconnaissance faciale, détection de plaque d'immatriculation ou encore détection de piétons.

## 3. Resnet (Residual Neural Network)

Un réseau de neurones résiduel (ResNet) est un réseau de neurones artificiels utilisant des connexions saute-couche ou raccourcis afin de sauter par-dessus certaines couches tout en étant combiné avec les couches de sortie en addition.

## 4. Darknet

Darknet est une plate-forme open source de réseau de neurones développée pour l'apprentissage en profondeur de la vision pour ordinateur. Centrée principalement autour de la détection d'objets, segmentation sémantique et classification d'image. Ce framework utilise une architecture modulaire et flexible ainsi qu'une syntaxe simple afin de décrire et configurer les architectures de réseau de neurones. Il est capable de prendre en charge plusieurs architectures réseau populaires tel que YOLO tout comme de nombreux jeux de données.

## 5. EfficientDet

EfficientDet est une architecture de réseau de neurones convolutifs conçue pour la détection d'objets en temps réel. Elle est basée sur EfficientNet, une architecture efficace et légère pour la classification d'images. Cette dernière utilise une approche de détection d'objets en une seule passe de réseau similaire à SSD, mais avec d'importantes améliorations de précision et de vitesse de traitement.

### 1.4.1 Algorithmes de détection d'objets

Un algorithme de détection d'objets est un type d'algorithme d'apprentissage automatique qui permet de localiser et d'identifier les objets présents dans une image ou une vidéo. Les

1. Opération de propagation des données dans un réseau de neurones convolutif

algorithmes de détection d'objets sont basés sur des réseaux de neurones profonds qui sont entraînés à reconnaître les caractéristiques visuelles des objets et à les distinguer du reste de l'image. Ces algorithmes sont capables de détecter plusieurs objets dans une seule image et de fournir des informations sur leur emplacement et leur catégorie. Les algorithmes de détection d'objets sont largement utilisés dans de nombreuses applications, telles que la surveillance de sécurité, la reconnaissance de visages ou encore la conduite autonome.

### 1. **YOLO(You Only Look Once)**

C'est un algorithme populaire pour les applications nécessitant une détection rapide d'objets en temps réel, il peut reconnaître plusieurs objets dans une image en temps réel. Contrairement aux algorithmes traditionnels qui travaillent sur plusieurs régions de l'image, YOLO utilise un seul réseau de neurones pour prédire simultanément les bounding box et les probabilités de classe, ce qui le rend très rapide.

### 2. **DETR(DÉtection TRansformer)**

C'est un algorithme de détection d'objets récent qui utilise un transformer pour encoder l'image, il est capable de prédire simultanément les emplacements des objets ainsi que leurs classes, ainsi que d'effectuer une correspondance globale des objets détectés, cet algorithme est certes performant et rapide mais n'a pas encore été assez adopté dans les applications pratiques).

### 3. **R-CNN**

R-CNN (Regions with CNN features) est un algorithme de détection d'objets qui fonctionne en extrayant des propositions de régions d'intérêt à partir d'une image, puis en utilisant un réseau de neurones convolutifs pour extraire des caractéristiques de chaque région. Ensuite, un classificateur est utilisé pour déterminer la classe de chaque région. Cette méthode a été une avancée importante dans la détection d'objets, mais a depuis été remplacée par des algorithmes plus rapides et plus précis.

### 4. **Fast R-CNN**

C'est un algorithme de détection d'objets qui étend le modèle R-CNN, il utilise une couche de rétropropagation pour régulariser les caractéristiques partagées, ce qui améliore la précision de la détection. Ces améliorations ont permis à Fast R-CNN d'améliorer les performances de l'algorithme le rendant plus rapide et plus précis qu'R-CNN.

### 5. **Faster R-CNN (Region-based Convolutional Neural Network) Et Mask R-CNN**

C'est un algorithme de détection d'objets largement utilisé qui se compose de deux étapes. La première étape génère des propositions de régions et la deuxième étape classe et affine ces propositions. Il est connu pour sa grande précision et a été largement utilisé dans diverses applications. Mask R-CNN étend Faster R-CNN en ajoutant une branche pour prédire également des masques de segmentation pour chaque objet détecté. Cela permet non seulement de détecter des objets dans une image, mais aussi de les segmenter précisément.

Tous ces algorithmes sont utilisés pour la détection et la localisation d'objets, mais ils diffèrent dans leur approche et leur performance dans différentes applications.

## 1.4.2 Modèles de réseaux de neurones pré-entraînés

### 1. **GPT-2(Generative Pre-trained Transformer 2)**

Modèle de réseau de neurone basé sur l'attention développé par OpenAI, sa fonction principale est de générer des données textuelles cohérentes, de haute qualité selon un contexte donné. Son architecture est de type Transformer et il peut utiliser un mécanisme de décodage uniquement.

## 2. GPT-2 Neo

Gpt-2 Neo est une version dérivée de GPT-2, de ce fait ces deux modèles ont la même architecture, néanmoins la version Neo a une empreinte mémoire réduite et une puissance de calcul moins performante que le modèle initial, ce modèle garde les mêmes fonctionnalités principales que GPT2, mais il est plus adapté à des environnements avec des ressources informatiques limitées tels que les smartphones.

## 3. BERT (Bidirectional Encoder Representation from Transformers)

Développé par Google AI, bert est un modèle de traitement du langage basé sur l'architecture transformer, conçu pour trouver les relations contextuelles entre les mots dans une base de donnée de type texte. Il utilise des mécanismes d'encodage et d'attention, sa particularité réside dans le fait qu'il soit un modèle bidirectionnel ce qui signifie qu'il prend en compte le contexte avant et après chaque mot d'une phrase lors de l'entraînement.

## 4. Tiny Bert

Version compressée et allégée de BERT, Tiny bert est conçu pour être moins gourmand en termes de taille et de ressources requises tout en gardant les mêmes fonctionnalités et la même architecture que le modèle initial, il utilise des techniques de compression telle que le transfert learning pour réduire l'empreinte mémoire et la charge de calcul, ce qui le rend idéal pour les systèmes embarqués.

# 1.5 Définition d'un jeu de données (dataset)

Un dataset est un ensemble de données structurées qui peut être constitué d'images, de vidéos, de données textuelles ou audio. Il est composé de deux parties principales : les données d'entraînement et données de test, son but est d'entraîner et tester des algorithmes d'apprentissage automatique. On lui attribue des tâches telles que la classification, la segmentation, la détection d'objets, la reconnaissance de voix et de caractères.

La qualité et la quantité des données d'un dataset sont des facteurs importants pour la performance des algorithmes d'apprentissage automatique. Plus le dataset est grand et diversifié, plus l'algorithme offrira de meilleures performances sur de nouveaux exemples. De plus, le choix d'un dataset approprié pour un problème d'apprentissage automatique spécifique est crucial afin d'obtenir des résultats précis et fiables.

### 1.5.1 Dataset COCO

COCO (Common Objects in COntext) [33] est un jeu de données très populaire en vision par ordinateur et en reconnaissance d'objets. Contenant plus de 330 000 images et plus de 2.5 millions d'instances d'objets annotées sur 80 classes différentes. On y trouve des animaux, des personnes, des véhicules, des meubles, des outils ou encore des appareils électroniques. Chaque image est accompagnée d'annotations identifiant les objets présents, les contours précis ainsi que les boîtes englobantes ( Bounding Box) autour de ces derniers. Ce jeu de données a été conçu pour encourager la recherche sur la reconnaissance d'objets au sein de scènes complexes et différentes afin d'être utilisé dans de nombreux défis ou encore compétitions de vision par ordinateur. Comme on peut le voir dans la figure 1.1, le dataset coco est adapté pour plusieurs types de tâches telles que classification, segmentation ou encore détection de contour.

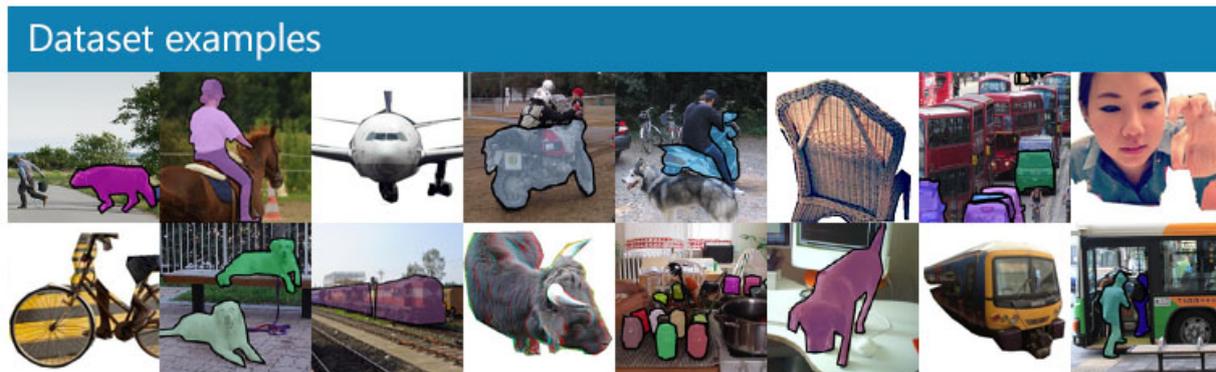


FIGURE 1.1 – Exemple du dataset COCO [9]

## 1.6 Outils d'aide à la navigation

### 1. Electronic orientation aids (EOAs)

Caméra et capteurs sensoriels pour la détection d'obstacles, mais manque d'informations par difficulté d'incorporation d'outils électroniques légers et le traitement en temps réel. Il serait intéressant de trouver un moyen de perdre du poids sur les différents moyens de récolte de données.

### 2. Position locator devices (PLDs)

Combinaison de technologies GPS / GIS pour connaître la position exacte de l'appareil, mais difficulté et souvent impossibilité de détecter les obstacles en face du sujet ainsi que l'impossibilité de fonctionner en intérieur par besoin de recevoir des signaux satellites pour le bon fonctionnement. Il serait intéressant de le combiner à des capteurs sensoriels afin d'aider avec la détection d'obstacles et le pathfinding / ajouter une carte réseau pour rendre l'accès au satellite continu.

### 3. Electronic travel aids (ETAs)

Outil d'aide à l'amélioration de l'évasion d'obstacle en améliorant le calcul de distance jusqu'à ce dernier et la focalisation sur des repères. grâce à des détecteurs sensoriels de l'environnement et une ou plusieurs modalités de sorties pour faciliter la navigation.

### 1.6.1 Les types d'entrées sensorielles

#### 1. Caméra de smartphone moderne

Petit en taille et de bonne qualité, cette caméra a tout pour plaire, mais malheureusement, il manque les données de profondeur afin de calculer les distances entre le dispositif et l'obstacle.

#### 2. Caméra stéréo avec une intégration de profondeur

Cette caméra possède un avantage très grand sur les autres caméras standards puisqu'elle permet de prendre en considération la profondeur de l'image et par ce fait nous permet de calculer la distance entre le dispositif et l'objet, mais reste peu avantageuse par le prix très élevé et parfois même la taille de ce dernier (ie : Microsoft Kinect)

#### 3. Identification radiofréquence

Dans l'identification radiofréquence (Radio Frequency Identification RFID), les données sont enregistrées comme Labels et Tags à partir de vagues radios. Malheureusement, cette technologie souffre des changements de fréquences et autres parasites qui peuvent nuire aux résultats et les performances.

Technologie très utilisée pour l'échange de données à très petite distance, principalement pour le paiement bancaire, mais malheureusement, avec une distance maximale de 10 cm, il n'est pas intéressant de l'implémenter ici.

#### 4. Communication en champ proche

Communication en champ proche (ou Near-Field Communication NFC) est une technologie très utilisée pour l'échange de données à très petite distance, principalement pour le paiement bancaire, mais malheureusement, avec une distance maximale de 10 cm, il n'est pas intéressant de l'implémenter ici.

#### 5. Navigation basée sur les ultrasons

Permettant de mesurer la distance entre l'utilisateur et les obstacles en se basant sur des ultrasons. Cette technologie souffre malheureusement de problèmes de précision puisqu'elle n'arrive pas à déterminer avec exactitude les positions des obstacles ni même les types d'obstacles.

#### 6. Capteurs infrarouges

Capteurs infrarouges (ou Infrared sensors IR) sont utilisés pour ressentir certaines caractéristiques de ses alentours en émettant des signaux infrarouges ou en les détectant. Les IR peuvent aussi ressentir les chaleurs émises par les objets. L'un des problèmes principaux de ces capteurs est l'interférence des lumières naturelles ou artificielles, tout comme le coût d'installation et de maintenance.

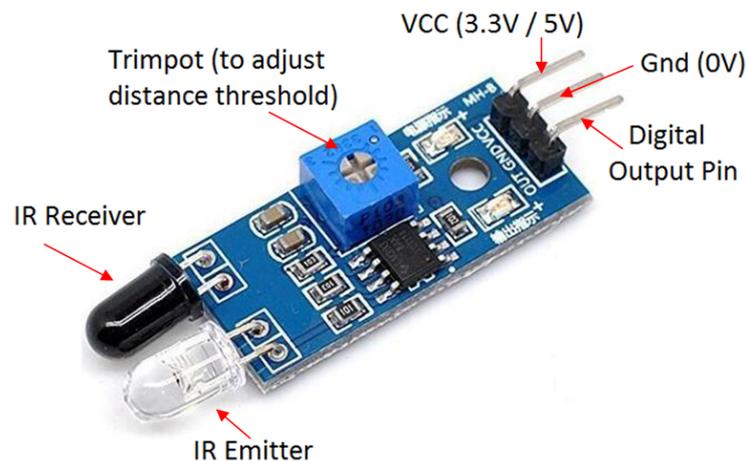


FIGURE 1.2 – Capteur infrarouge

#### 7. Ultrasonic sensor (capteur ultrason)

Les capteurs ultrasons mesurent le parcours du son entre le détecteur et l'objet cible pour fournir des mesures de précision, bien que la précision puisse diminuer avec la distance. Ils peuvent être utilisés pour mesurer une distance, détecter des objets mal positionnés ou de forme irrégulière, etc. Les capteurs émettent des impulsions qui se propagent dans l'air jusqu'à ce qu'elles rencontrent un objet et retournent vers le capteur. En mesurant la durée du trajet aller-retour des impulsions, la distance peut être déterminée en se basant sur la vitesse constante du son dans l'air.



FIGURE 1.3 – Capteur ultrason

## 1.6.2 Les types de sorties sensorielles

### 1. Audio

Le feed-back audio utilisé dans les systèmes de navigation s’usant d’écouteurs ou casques audio ou encore de baffles et autres sorties audio. Le désavantage de ce dernier est le dérangement causé lorsque l’information est surchargée ou encore l’agacement quand l’utilisateur rate les sons de l’environnement dû à des indices auditifs. Des solutions ont été proposées pour remédier à ça, notamment

### 2. Conduction osseuse

Des écouteurs à conduction osseuse (Bonne conduction headphones) permettent de conduire le son à l’oreille interne tout en permettant à l’utilisateur de recevoir des signaux audio sans avoir à bloquer le canal de l’oreille.

### 3. Tactile (haptique)

Le feed-back ici est donné à l’utilisateur à travers l’une des parties où le sens du touché est éprouvé, que ça soit au niveau du pied, bras, doigts ou n’importe quelle autre partie de corps. Elle permet à l’utilisateur d’être notifié sur les indices directionnels et d’également éviter les obstacles en ressentant des sensations au niveau des différents points de pressions du corps. Contrairement au retour audio, cette méthode tactile peut être utilisée afin d’éviter de distraire l’utilisateur dans des environnements dans lesquels il a besoin du son.

### 4. Électrique

Le retour se fait sous forme de décharges électriques juste assez puissante pour stimuler le muscle que nous souhaitons notifier, à savoir : La langue, les doigts, la paume de la main.

### 5. Thermale

Le retour se fait sous forme d’augmentation de la chaleur sur certains points de la peau du corps humain afin de le notifier d’obstacles.

### 6. Vibrations

Ce type de feedback a été utilisé dans plusieurs systèmes. Un smartphone est utilisé ou encore d’autres types de dispositifs mobiles afin de donner des retours en forme de vibration à l’utilisateur en lui donnant des indices de navigation. Malgré tout, ce n’est pas si utile ou confortable pour les utilisateurs. Ainsi, il peut y avoir parfois confusion ou un temps de traitement trop important.

Types de sortie	Audio	Conduction Osseuse	Électrique	Thermale	Vibration
<b>Difficulté de déploiement</b>	Facile	Facile	Difficile	Difficile	Facile à Moyen
<b>Coût</b>	Peu onéreux à très onéreux	Très onéreux	Peu onéreux à très onéreux	Moyennement onéreux à très onéreux	Peu onéreux
<b>Ergonomie</b>	Prise en main facile	Prise en main facile	Prise en main facile	Prise en main facile	Prise en main facile
<b>Consommation Énergétique</b>	Petite	Petite	Grande	Grande	Petite
<b>Intrusion</b>	Aucune mais peut désorienter l'utilisateur	Aucune	Dérange beaucoup l'utilisateur surtout dans des parties sensibles telles que la langue	Peut être agréable à utiliser ou l'inverse selon la température ou le nombre d'obstacles à notifier	Si le nombre d'obstacles est élevé, les constantes notifications peuvent devenir très désagréables et même douloureuses

TABLE 1.2 – Tableau comparatif des différents outputs possible

## 1.7 Quelques matériels utilisés pour l'aide à la navigation

### 1. Raspberry Pi

La Raspberry Pi est un mini-ordinateur peu coûteux conçu pour être accessible aux amateurs, étudiants et éducateurs. Il est capable d'exécuter différents systèmes d'exploitation et peut être utilisé pour une grande variété de projets, tels que l'automatisation domestique, les centres multimédias, la robotique, et bien plus encore. Son nom provient du fruit, car il est petit et a une forme et une taille similaires à une framboise.

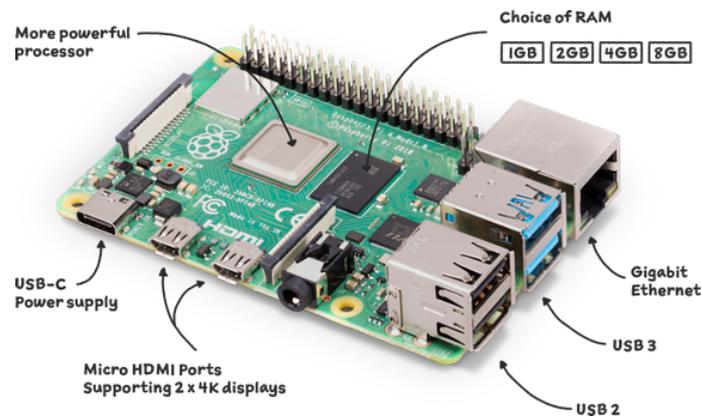


FIGURE 1.4 – Raspberry Pi

### 2. Canne intelligente

Une smart canne est un dispositif électronique doté de capteurs et de technologies de navigation qui aide les personnes aveugles et malvoyantes à se déplacer en toute sécurité en détectant les obstacles. Elle peut aussi être munie d'émetteurs afin de leur fournir des

informations sur leur environnement.

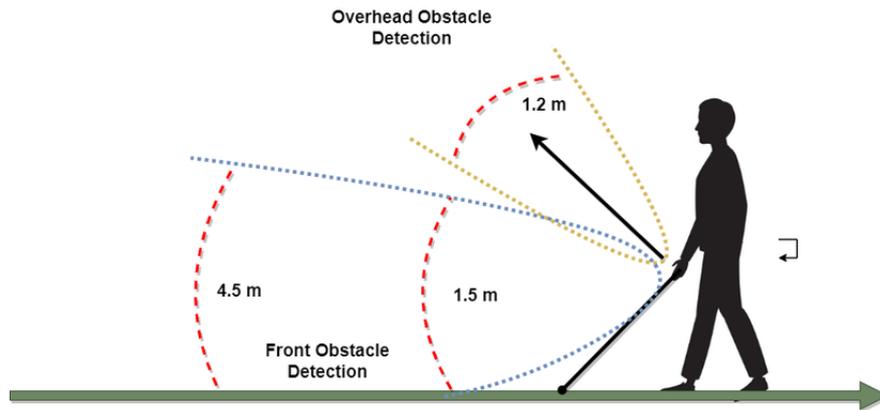


FIGURE 1.5 – Smart canne

### 3. Lunettes intelligentes

C'est des dispositifs portables qui ont des technologies intégrés telles que des capteurs, des caméras et des écrans, permettant à l'utilisateur d'interagir avec son environnement de manière plus intuitive et personnalisée. Ils peuvent être utilisés pour une variété d'applications, telles que la réalité augmentée, la reconnaissance faciale et la navigation assistée.



FIGURE 1.6 – Lunettes Intelligentes

### 4. Module GPS

Les modules GPS sont des dispositifs électroniques compacts qui utilisent des signaux satellites pour déterminer la position géographique d'un objet ou d'une personne en temps réel.



FIGURE 1.7 – Module Gps

## Conclusion

Dans ce chapitre, nous avons défini les notions et concepts essentiels au bon suivi de ce rapport. Dans le chapitre suivant, nous allons faire une étude des différents travaux déjà abordés sur le thème en question suivi d'une étude comparative et de commentaire sur nos recherches.

## Chapitre 2

# État de l'Art

### 2.1 Introduction

Dans le monde d'aujourd'hui et avec toutes les avancées technologiques actuelles, il est de plus en plus difficile de se déplacer pour une personne aveugle ou malvoyante et cela revient principalement à l'ajout constant de nouveaux obstacles. Plusieurs outils d'aide à la navigation ont déjà été proposés par le passé et très peu d'entre eux ont été pris en considération ou même mis sur le marché. Nous allons discuter dans ce qui suit ces différentes solutions et les critiquer.

## 2.2 Qu'es-ce que O et M ?

Les personnes malvoyantes utilisent souvent des aptitudes d'orientation et de mobilité qui les aident à naviguer dans les environnements non-familiers (Orientation and Mobility skills O et M). Ces capacités leur permettent de développer des compétences clés et requises pour une navigation efficace et sécurisée. **L'orientation** fait référence à l'habilité de connaître sa localisation courante et la destination vers laquelle la personne veut se déplacer, elle inclut des informations telles que la volonté de se déplacer d'une pièce à une autre ou veut se déplacer vers un établissement particulier. **La mobilité** fait référence à la capacité qu'a une personne de se déplacer de façon sécurisée et efficace d'une position à une autre, cela inclut le déplacement vers un arrêt de transport sans tomber ou encore couper la route et l'usage sécurisé des moyens de transport publics.

## 2.3 Revue des différents articles

### 2.3.1 Systèmes à imagerie visuelle

#### 1. Caméra stéréo

##### (a) Tyflo

Deux caméras [5] qui capturent des images de l'environnement soit par la demande de l'utilisateur, soit par un mode vidéo continu). Le système convertit ensuite les images en descriptions verbales afin de les communiquer à l'utilisateur.



FIGURE 2.1 – Tyflo (Adapté de [5])

##### (b) Binaural rendering

Le système [38] de navigation intègre une caméra jumelle ainsi qu'un IMU (inertial measurement unit) et des écouteurs dans le casque d'un vélo. Quand l'objet détecte une position particulière, celle-ci sera convertie en une source sonore et envoyée aux écouteurs. c'est une technique de création de sons qui peuvent être localisées en directions ainsi que distance en s'aidant d'écouteurs. Ce système peut uniquement fonctionner en extérieur.



FIGURE 2.2 – Binaural Rendering (Adapté de [38])

## 2. Réseau de caméras IP

Plusieurs caméras [17] sont installées dans les plafonds de chaque pièce. Les images de l'environnement capturées sont analysées par un ordinateur s'aidant d'algorithmes de vision. En utilisant une simple application mobile interactive, l'utilisateur serait capable d'atteindre sa destination. Le problème principal serait le coût d'installation des différentes caméras IP dans un environnement IP.

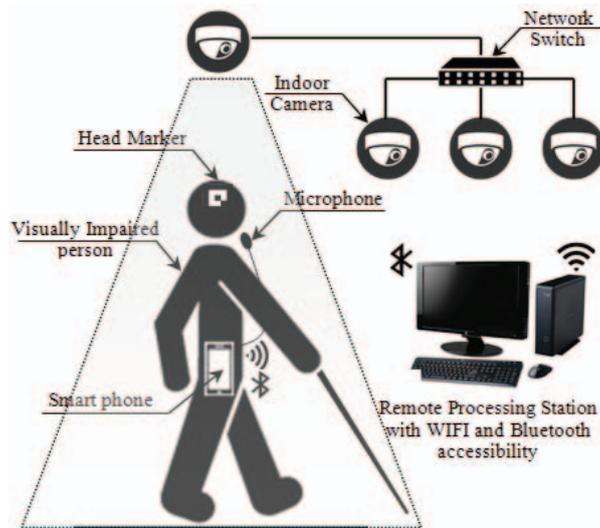


FIGURE 2.3 – Réseau de caméras IP (adapté de [16])

## 3. VS-LAM (Visual Simultaneous Localization and Mapping)

Technologie [16] qui peut être utilisée pour le positionnement et la localisation s'aidant d'entrée visuelle d'une caméra. Elle est devenue populaire grâce au fait qu'elle n'ait besoin que d'une seule caméra sensorielle et peut fonctionner sans aucun autre capteur.

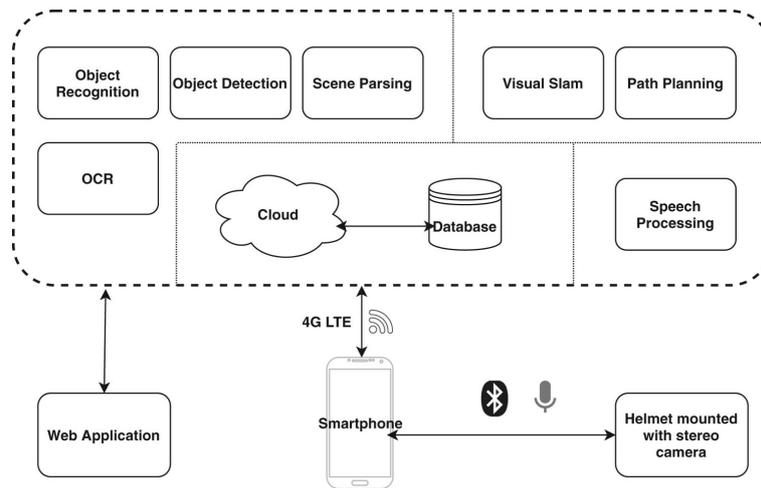


FIGURE 2.4 – VS-LAM

#### 4. Caméra RGB-D

##### (a) ISANA (Intelligent situational awareness and navigation aid)

Prototype [4] de canne intelligente qui se base sur la tablette Google Tango. S'aidant des algorithmes, des filtres de Kalman et des caméras de profondeur à bord de la tablette pour avoir une détection d'obstacles efficace.



FIGURE 2.5 – ISANA (Adapté de [4])

##### (b) Microsoft Kinect

Type de caméra RGB-D utilisée pour le design de systèmes de navigation pour les malvoyants. Kinect [4] est un produit de Microsoft qui détecte les lignes de mouvements et plus précisément les objets et, de ce fait, permet d'être un outil de navigation. Ce dernier supporte également plusieurs fonctionnalités et peut aussi fonctionner dans des environnements avec faible luminosité. Le système proposé par les auteurs a utilisé un

algorithme qui prend ses données de la Kinect d'une Xbox 360 de Microsoft. Elle aide à créer une carte 3D des zones d'intérieur et détecte la profondeur d'un obstacle et/ou d'un humain.

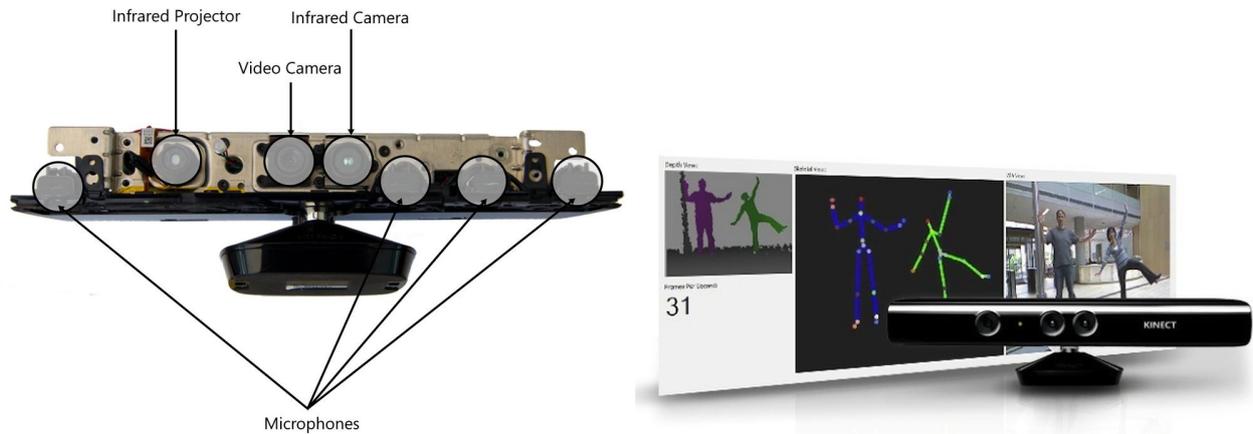


FIGURE 2.6 – Microsoft Kinect

## 5. Lidar

Prototype nommé LASS [37] (Lidar Assist Spatial Sensing system) qui permet d'obtenir les informations spatiales de l'environnement de l'utilisateur à travers plusieurs entrées audio de fréquences différentes (écholocation). Le système est capable d'identifier quantitativement les obstacles à proximité, différencier leur distance relative ainsi que distinguer la position de multiples objets avec un entraînement minime.

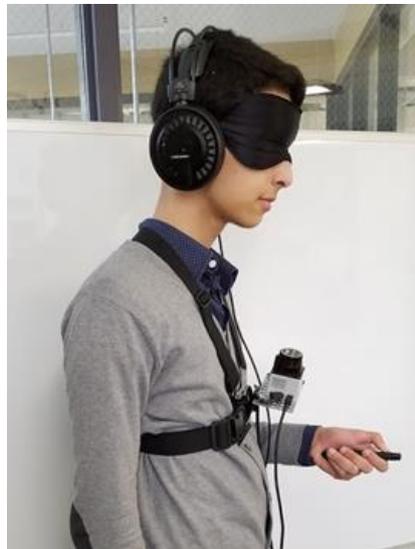


FIGURE 2.7 – Partissant utilisant le système LASS (Adapté de [37])

Le tableau ci-dessous résume et compare les différents systèmes cités :

Nom du système	Matériel utilisé	Avantages	Inconvénients
Tyflo	Deux caméras, gilet haptique (vibrations)	Permet d'avoir une idée de l'environnement. Sorties audio en signal	Constantes vibrations intrusives. Encombrement de l'ouïe.
Binaural Rendering	Caméra, IMU, écouteurs, casque de vélo	Immersion sonore réaliste Localisation précise du son.	Sensibilité aux variations individuelles (la perception varie d'une personne à une autre). L'utilisation d'un casque audio peut parfois être un handicap.
Réseau de caméras IP	Plusieurs caméras, ordinateur, smartphone	Flexibilité et évolutivité. Possibilité d'accéder aux flux vidéos en temps réel.	L'utilisation d'un grand nombre de caméras IP simultanément peut mettre le réseau à rudes épreuves. Les réseaux de caméras IP peuvent être vulnérables t cibles de pirates.
VSLAM	Caméra, casque de vélo, smartphone, sortie et entrée audio	Localisation en temps réel. Peut fonctionner dans des environnements variés et dynamique. S'adapte aux changements d'éclairage.	Sensible à l'environnement (une faible luminosité ou de simples reflets peuvent perturber la précision et la fiabilité). Consommation en énergie et coût de calcul élevés.
Isana	Canne, tablette Google Tango.	Fournit des informations de localisation et instructions vocales en temps réel. Utilisation des balises Bluetooth à faible consommation d'énergie.	Dépendance aux balises bluetooth et à leurs couvertures. Complexité de mise en œuvre.
Mirosoft Kinect	Caméra RGB-D Kinect	Interaction immersive et naturelle (sans prise de contrôle physique). Grande capacité à suivre les mouvements du corps et à reconnaître les expressions faciales.	Précision limitée (en cas de mouvements trop rapides). Configuration et espace requis.
LASS	Lidar, casque audio, microphone, capteur IR.	Précision et résolution élevée. Fonctionnement indépendant de l'éclairage. Calcul de distance précis.	Peut être coûteux. Sensibilité aux conditions atmosphériques.

TABLE 2.1 – Tableau comparatif des systèmes à imagerie visuelle

### 2.3.2 Systèmes de données non-visuelles

Ces systèmes N'utilisent pas les "Algorithmes de Vision" et "Capteurs Optiques" mais se basent plutôt sur des composants comme les capteurs ultrasons, balises (beacons) et capteurs infrarouges.

#### 1. Ble Beacons

Système utilisant des balises Bluetooth (signal), Nair et al [26] ont proposé une solution hybride qui combine le Bluetooth à faible utilisation de batterie et Google tango , un smartphone fut utilisé pour l'interaction. La BLE-Tango Hybrid System utilise Google tango

pour enregistrer un ADF (carte de caractéristiques). Pour chaque étage d'un bâtiment, elle transforme les données récoltées en 3D en un plan 2D. La position de l'utilisateur est alors suivie sur une interface d'application adaptative composée d'un plan 2D de l'environnement dans lequel il se trouve. Une assistance vibro tactile vient compléter le système, à l'aide d'un capteur infrarouge placé au niveau du poignet, l'utilisateur sera averti en fonction de sa distance d'un objet proche (la vitesse de vibration dépend de cette distance). Le système aide uniquement à la navigation interne, il utilise la technologie Bluetooth, un smartphone muni d'une caméra RGB-D et des balises qui nécessitent un déploiement correct et précis. ce qui rend la tâche difficile pour une navigation uniquement en interne. Le point faible majeur de la solution se situe au niveau de Google tango, en effet cette réalité augmentée qui donne de très bons résultats en termes de localisation fut abandonnée par Google au profit d'ARCore ce qui la rend moins exploitable.

**Matériel utilisé :** Ble-beacon, Smartphone, Capteurs infrarouges, assistance vibro active.

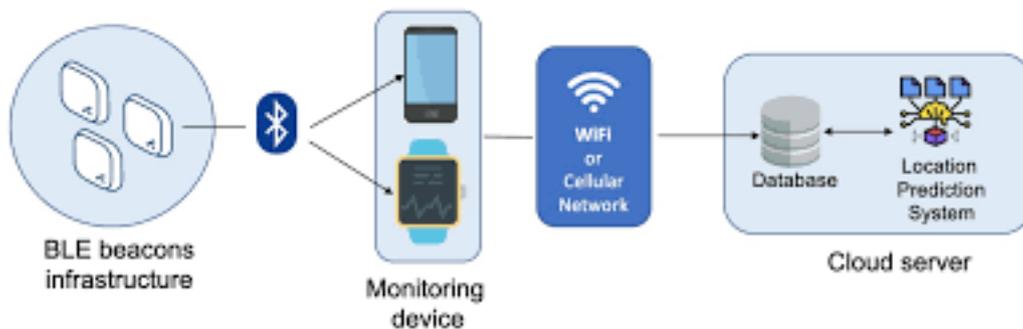


FIGURE 2.8 – BLE Beacons (Adapté de [26])

## 2. IoT Based

Intercommunication de systèmes variés pouvant transférer des données à travers un réseau sans aucune interaction homme-machine. On peut citer l'exemple d'Indriya [15], qui est une smart canne qui aide à la navigation, elle peut détecter des obstacles statiques et dynamiques à partir de 3 mètres, et elle fait une distinction entre l'humain et l'objet précise, 80 pourcent, elle émet des vibrations et alertes vocales avant chaque collision possible. Le prototype a coûté moins de 35 dollars.

**Matériel utilisé :** Dispositif Interactif, feedback vibro-tactile et sonore, smartphone (android), boutons interactifs, module bluetooth, pro mini, buzzer.



FIGURE 2.9 – Indriya (Adapté de [15])

## 3. Blind guide [39]

En utilisant une canne qui aide à la navigation intérieure et extérieure, si le capteur détecte

un obstacle, il envoie un signal à l'unité de calcul, l'identification de l'objet se fait en se connectant au cloud à travers un micro-controller Wifi, l'utilisateur sera informé sur le nom de l'objet et la distance qui le sépare de lui à travers un feedback sonore. Ces systèmes sont dépendants, car ils requièrent une connexion internet permanente. L'autre point négatif que l'on peut citer est l'utilisation de smartphones qui se déchargent rapidement.

**Matériel utilisé :** Canne, microcontrôleur, Wi-fi, Caméra, Raspberry pi, Capteur ultrason.



FIGURE 2.10 – Blind guide (Adapté de [39])

#### 4. Systèmes utilisant les capteurs ultrasons

La Navigation, système basé sur des capteurs ultrasons, peut être considérée comme un choix design commun après la caméra, elle utilise des technologies compatibles avec des cartes Raspberry pi et Arduino [32].

**Matériel utilisé :** Une canne ajustable sensible, capteur ultrason, module gps.

#### 5. NaviGuide [28]

Capable de catégoriser les obstacles et situations environnantes à l'aide de capteur à ultrasons, le système peut fournir des informations à son utilisateur à l'aide d'alertes vibratoires, sonores. Il peut aussi détecter les obstacles au niveau du sol et au niveau du genou, une de ses limites : ne détecte pas les fosses/ descentes / détecte un sol mouillé après que l'utilisateur a déjà marché dessus.

**Matériel utilisé :** Une canne ajustable sensible, capteur ultrason, module gps.

#### 6. Capteurs infrarouges

Ces capteurs offrent une faible consommation d'énergie et un faible coût par rapport aux capteurs ultrasons. Les technologies basées sur l'IR ont été expérimentées dans des systèmes de navigations comme Le projet combinant Google tango et UNITY (exemple : visual and infrared sensor data based obstacle detection for visually impaired using the Google project tango tablet development kit and the unity engine [13]) on peut citer l'exemple du ' Low energy precise navigation system for the blind with infrared sensors ' [23] qui présente une solution en utilisant un capteur infrarouge qui détecte des objets variés, comme des murs, buildings, le dispositif doit être placé sur le bras de l'utilisateur et peut transmettre des signaux de navigation par des vibrations

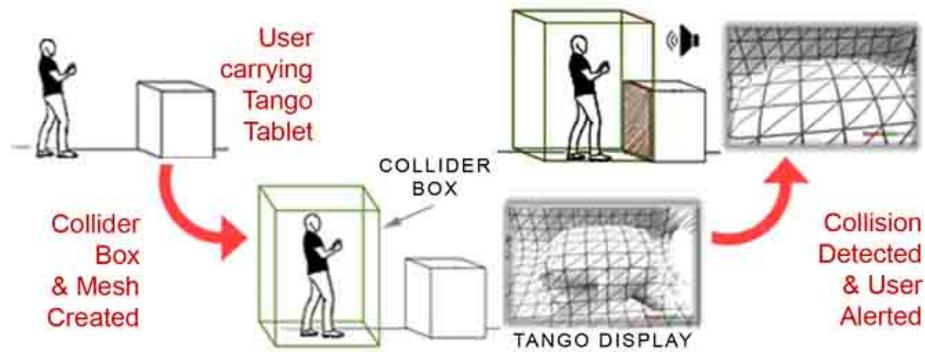


FIGURE 2.11 – Google Tango

## 7. Système avec son 3D

Le but de ces systèmes est d'aider à la navigation des personnes malvoyantes dans un environnement inconnu en interne ou externe, en utilisant des dispositifs portables. On peut citer l'exemple du système The sound of vision [6], un capteur ayant pour objectif de donner une représentation 3D de l'environnement et de renvoyer à l'utilisateur un retour audio et haptique.

**Le système se base sur 4 étapes principales :**

- Acquisition d'informations 3D à partir de l'environnement.(recueil les éléments d'intérêts comme les obstacles).
- Reconstruction 3D de l'environnement détecté et segmentation en objet d'intérêt
- Modélisation Audio et haptique du traitement 3D.(encodage)
- Renvoi de l'information haptique et audio à l'utilisateur.

Le système utilise une caméra RGB stéréo (SC) utilisée pour l'imagerie externe, une caméra avec outil de profondeur pour l'intérieur, Dispositif (unité de mesure) IMU qui récolte les données sur l'orientation de la tête et des caméras.

Perception de l'environnement par l'utilisateur : l'utilisateur reçoit des informations concernant la localisation et la taille des objets, ainsi que leur taille, type et élévation. Le système est coûteux, il a besoin d'amélioration dans sa capacité à être utilisable et ses facteurs de précision.

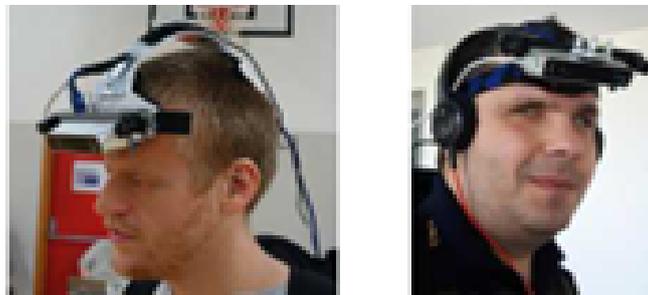


FIGURE 2.12 – The sound of vision (Adapté de [6])

### 2.3.3 Soutuion basée sur les smartphones

#### 1. NavCog3 [31]

Système de navigation intérieure qui fournit des instructions détaillées et un retour immédiat lorsqu'une orientation incorrecte est détectée, il fournit des informations sur les points de repères à proximité, le système fournit un retour audio aux utilisateurs.

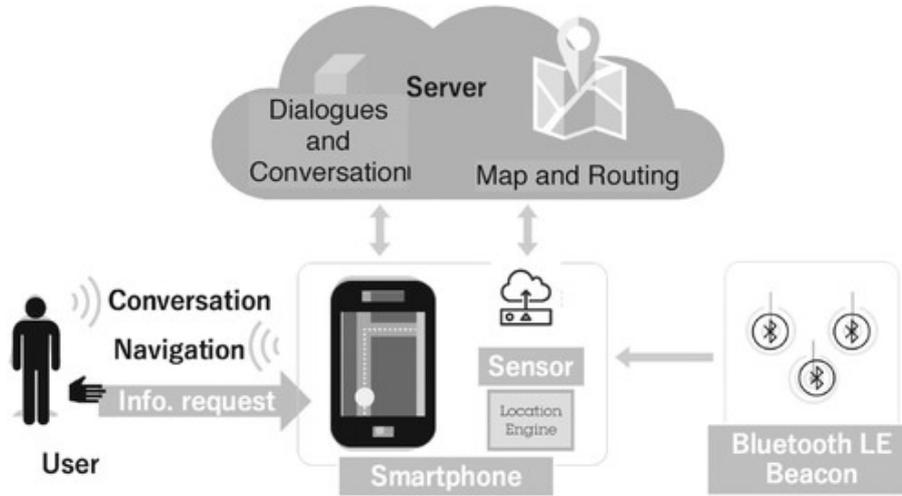


FIGURE 2.13 – NavCog3 (Adapté de [31])

## 2. Percept 2 [11]

L'utilisateur reçoit des instructions de navigation vers la destination choisie, les spots destination ont été marqués avec des balises Near field communication.

Une application pour smartphone pouvant être intégrée à un système de reconnaissance d'images pour former un système de navigation assistée. En fonction de la disponibilité du réseau, le système peut choisir deux modes de fonctionnement : un mode en ligne et un mode hors ligne. Lorsque le système est lancé, le smartphone capture une image puis l'envoie au serveur pour traitement. Le serveur utilise des algorithmes d'apprentissage en profondeur pour reconnaître différents obstacles. Les principales limitations du système incluent une consommation d'énergie élevée et le besoin d'une connectivité réseau à haut débit.

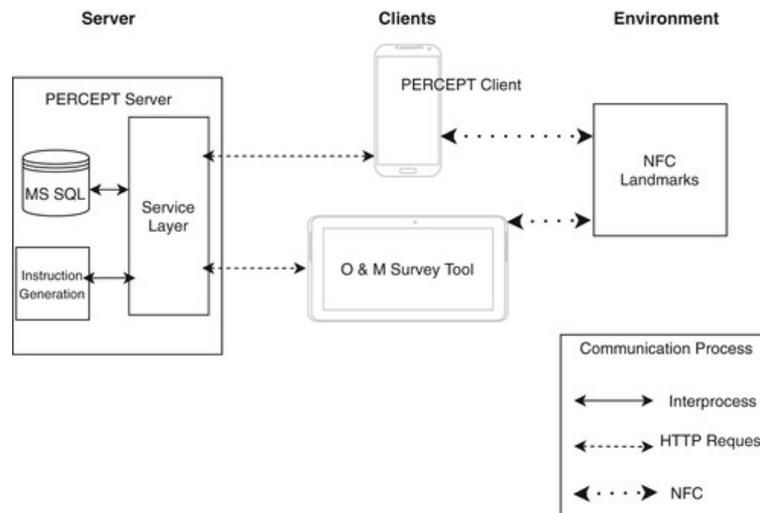


FIGURE 2.14 – Percept 2 (Adapté de [11])

Le tableau ci-dessous résume et compare les systèmes à imagerie non visuelle ainsi que les systèmes basés smartphones :

Nom du système	Matériel utilisé	Avantages	Inconvénients
<b>BLE Beacons system</b>	Ble-balises, Smartphone, Capteurs infrarouges, Assistance vibro active.	Faible utilisation de batterie. Google tango fournit des informations fiables et précises sur la localisation.	Le système aide uniquement à la navigation en interne. Le système utilise la réalité augmentée de Google tango pour la localisation, mais cette dernière fut abandonnée par Google.
<b>Indriya (Basé IoT)</b>	Canne, feedback vibrotactile et sonore, smartphone, boutons interactifs, module Bluetooth, pro mini, buzzer.	Détecte les obstacles statiques et dynamiques à partir de 3m. Fait la distinction entre l'humain et l'objet avec une précision de 80 Utilise une smart canne peu coûteuse.	Le système ne peut pas localiser l'utilisateur en temps réel. Le système est limité, il ne détecte pas les objets localisés à plus de 3 mètres.
<b>Blind guide (Basé IoT)</b>	Canne, Wi-Fi Microcontrôleur, Caméra, Raspberry Pi, Capteur Ultras-son.	Le système aide à la navigation interne et externe. Mesure la distance entre l'utilisateur et l'objet et renvoie une estimation fiable.	Le système requiert une connexion permanente. Le système est encombrant et intrusif.
<b>Navguide (Système qui se base sur les capteurs ultrasons)</b>	Canne ajustable et sensible, capteur ultrason, module gps.	Le système fournit des informations à l'utilisateur en catégorisant les obstacles et situations environnantes à l'aide de retours sonores et vibratoires.	Le système ne détecte pas les pentes et les fosses. Détecte ce qui se trouve au sol uniquement après que l'utilisateur ait marché dessus.
<b>Low energy precise navigation system for the blind with infrared sensors</b>	Capteurs Infrarouges	Les systèmes basés capteurs infrarouges sont abordables et peu coûteux. Le système renvoie des vibrations aux utilisateurs pour l'avertir du danger.	Le dispositif est placé autour du bras ce qui peut être gênant. Le système a des limitations quant à la localisation et la navigation en temps réel.
<b>The sound of vision</b>	Dispositif pour Feedback haptique et sonore, smartphone, Caméra, unité de mesure.	Donne une représentation 3D de l'environnement. L'utilisateur reçoit des informations sur la taille et la localisation des objets.	Le système est coûteux. Le système a besoin d'améliorations dans son utilisation et ses facteurs de précision.
<b>NavCog3</b>	Dispositif pour feedback audio, Smartphone, Canne, balise Bluetooth.	Détecte automatiquement si l'orientation est correcte ou non.	Requiert un Matériel spécifique. Le système marche que dans les lieux qu'il connaît déjà.
<b>Percept 2</b>	Balises, Near field communication, smartphone, Capteurs Bluetooth	Le système peut être utilisé dans des espaces publics tels que l'hôpital, aéroports et autres espaces fermés. Utilise des balises à faible consommation d'énergie.	Le système a besoin d'une connectivité réseau à haut débit. Le système Ne fonctionne qu'en intérieur.

TABLE 2.2 – Tableau comparatif des systèmes données non-visuelles et systèmes basées sur les smartphones.

### 2.3.4 Systèmes utilisant les méthodes d'apprentissage profond

#### 1. ENVISION [18]

Utilise une méthode spécifique pour détecter les obstacles statiques et dynamiques de manière robuste et précise en utilisant le streaming vidéo en temps réel enregistré par un smartphone, il est muni d'une technologie de pointe qui permet la reconnaissance d'image, la reconnaissance de texte et la synthèse vocale. Il utilise l'intelligence artificielle et les méthodes d'apprentissage profond pour analyser les et décrire l'environnement de l'utilisateur.

Les utilisateurs peuvent poser des questions au système et ce dernier peut fournir des réponses spécifiques à ces derniers. Le système utilise les algorithmes tels que YOLO, RetinaNet, SSD (single shot detector) afin d'effectuer une détection d'objets et permettre une navigation dans un environnement extérieur en temps réel.

Envision a remporté plusieurs prix et est reconnue pour son impact positif sur l'amélioration de la qualité de vie des personnes malvoyantes.

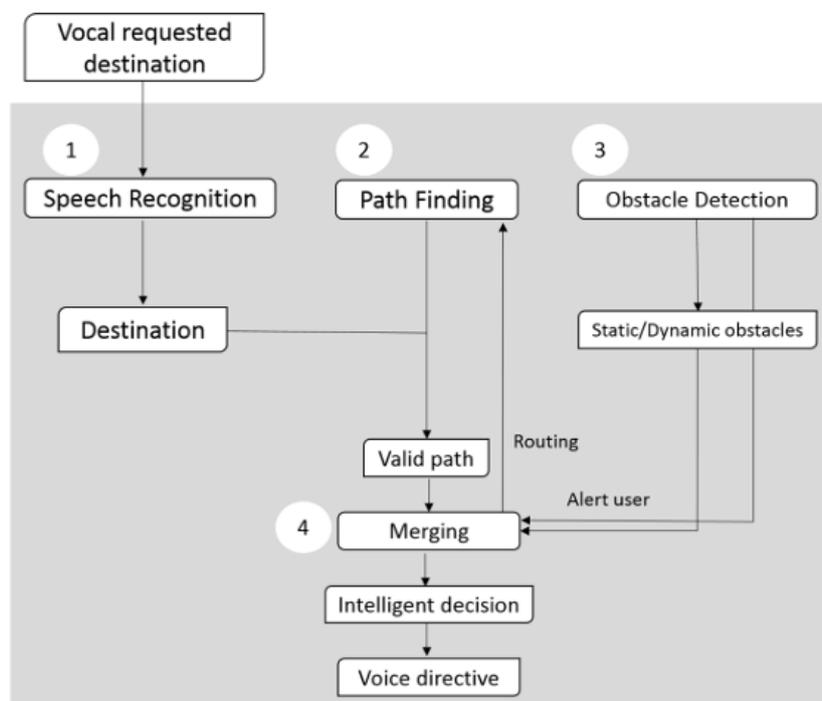


FIGURE 2.15 – Envision (Adapté de [18])

#### 2. Voice Based Navigation System for Blind People Using Raspberry Pi

Les auteurs [29] proposent une solution mobile et portable pour aider les personnes aveugles à naviguer dans leur environnement à l'aide d'un système de navigation vocal, la solution proposée consiste à utiliser une Raspberry Pi combinée avec un capteur infrarouge, un capteur à ultrasons et un module GPS pour fournir des informations de localisation en temps réel et des directions vocales à l'utilisateur. Le système utilise les API de TensorFlow et un système Text-to-Speech (TTS) pour fournir à l'utilisateur des directions vocales. L'utilisateur entre sa destination via une interface web, et le système calcule ensuite l'itinéraire et fournit à l'utilisateur des directions étape par étape. L'utilisateur peut également demander des informations supplémentaires, telles que la position actuelle ou les points de repère à proximité, en parlant dans le microphone. Les auteurs ont mis en place le système et l'ont testé sur des personnes aveugles pour évaluer son efficacité. Les résultats ont montré que le système était facile à utiliser et fournissait des directions précises aux utilisateurs. Le modèle utilisé fut entraîné sur la base de données COCO.

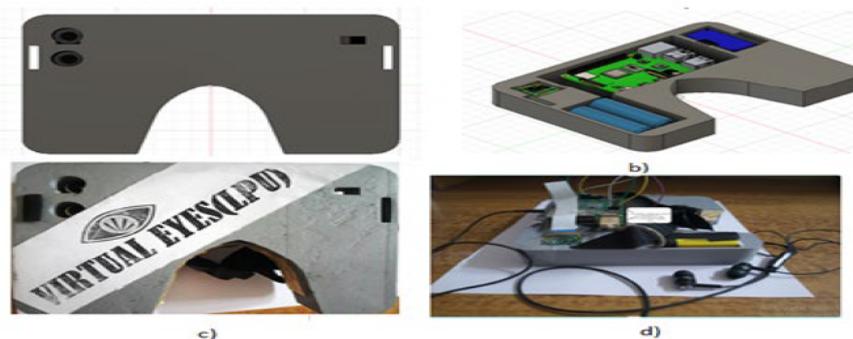


FIGURE 2.16 – A Deep learning based model to assist blind people in their navigation (Adapté de [29])

### 3. Smart Glass System Using Deep Learning for the Blind and Visually Impaired

Les auteurs [32] ont contribué au développement d'un système de lunettes intelligentes qui utilise des algorithmes d'apprentissage profond pour aider les personnes aveugles et malvoyantes dans leur vie quotidienne. Le système se compose d'un dispositif de lunettes intelligentes portables équipé d'une caméra et d'un mini-ordinateur qui traite les images capturées par la caméra. L'algorithme d'apprentissage profond utilisé dans le système utilise un réseau de neurones à convolution (CNN) pour détecter et reconnaître les objets en temps réel. Le système comprend la détection d'objet, la méthode Saliency object extraction, l'extraction et de reconnaissance de texte, utilisant la vision par ordinateur et l'apprentissage en profondeur, il est capable de faire une reconnaissance d'objet dans des endroits à faible luminosité ou sombre.

Les auteurs ont entraîné les modèles tels que Faster RCNN, DETR, ExtremeNet, etc. sur les bases de données ExDart et LOL pour arriver à des résultats satisfaisants permettant une navigation en temps réel.

**Matériel utilisé :** Lunettes intelligentes, capteur GPS, capteur ultrason, smartphone, écouteurs.

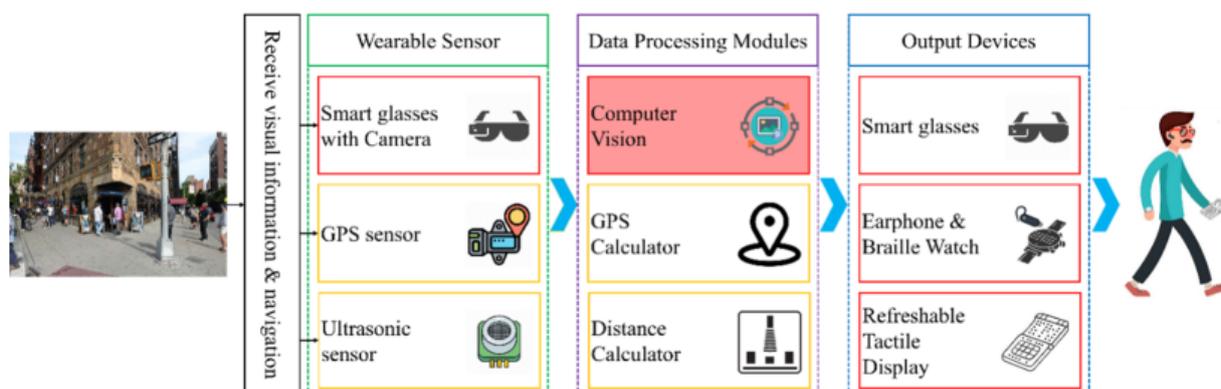


FIGURE 2.17 – Smart Glass System Using Deep Learning for the Blind and Visually Impaired (Asapté de [32])

le tableau ci-dessus résume et compare les systèmes utilisant les méthodes d'apprentissage profond :

Nom du système	Modèle utilisé	Dataset utilisé	Taux de réussite
<b>ENVISION</b>	YOLO, RetinaNetSSD.	COCO.	92.6% pour la détection d'obstacles.
<b>Voice Based Navigation System for Blind People Using Raspberry Pi</b>	SSD model (single-shot detector).	COCO.	74.3%
<b>A deep Learning based model to assist blind people in their navigation</b>	YOLOv5.	COCO.	89.5%
<b>Smart Glass System Using Deep Learning for the Blind and Visually Impaired</b>	Faster RCNN, DETR, ExtremeNet.	ExDart ,LOL.	93.6%

TABLE 2.3 – Tableau comparatif des systèmes utilisant les méthodes d'apprentissages profond

## Conclusion

Dans ce chapitre, nous avons pu voir les différentes approches des chercheurs dans le domaine de l'assistance de navigation. Nous les avons divisés en quatre catégories, puis nous avons comparé chacune des solutions en donnant ses avantages et limites, puis nous nous sommes inspirés de quelques-unes d'entre elles afin d'élaborer notre solution. Dans le chapitre suivant, nous aborderons les besoins matériels et logiciels utilisés pour implémenter notre méthode.

## Chapitre 3

# Conception et prototypage

### 3.1 Introduction

Le développement de prototypes est une activité qui consiste à réaliser un exemplaire unique d'un produit. Notre méthode aspire à utiliser le machine learning et le deep learning en s'appuyant sur des algorithmes d'apprentissage profond afin de créer un système intelligent embarqué pouvant aider une personne aveugle ou malvoyante à naviguer en extérieur.

Dans ce chapitre, nous explorerons en détail la contribution du Machine learning dans la conception de notre projet en commençant par citer les étapes du cycle de vie d'un projet ML, nous aborderons ensuite les besoins matériels et logiciels essentiels à conception de notre prototype. Enfin, nous exposerons l'aspect technique de notre méthode en présentant l'architecture de notre modèle ainsi que les fonctionnalités clés qui lui sont associées.

## 3.2 La contribution du ML dans un système intelligent embarqué

L'intégration des algorithmes de machine learning aux systèmes intelligents embarqués confère une capacité d'apprentissage des données collectées à partir des entrées sensorielles. Le système aura donc la possibilité de prendre des décisions intelligentes afin de réaliser des tâches de manière autonome.

Parmi les contributions des algorithmes de machine learning dans les systèmes intelligents embarqués, nous pouvons citer l'élaboration de systèmes utilisant la prédiction, la reconnaissance, la détection, l'optimisation des ressources, les systèmes de recommandations, et tant d'autres exemples.

Ces avancées permettent d'améliorer les performances, l'efficacité et l'expérience utilisateur de ces systèmes dans divers domaines d'application.

## 3.3 Cycle de vie d'un projet Machine Learning

Voici les étapes du cycle de vie d'un projet Machine Learning :

- **Étape 1 : Définir l'étendu du projet**

Dans cette étape, nous avons à définir les objectifs du projet en termes de résultats attendus ainsi que l'identification des contraintes et limitations du projet. Cela pourrait toucher les ressources disponibles, délais et exigences spécifiques.

- **Étape 2 : Récolte de données**

Au niveau de cette étape, il faut passer à l'identification et collecte en passant par la préparation des données pertinentes pour le projet en effectuant des opérations de nettoyage de données, fusion de plusieurs ensembles de données en finissant par la vérification de la qualité des données et effectuer des ajustements si nécessaires.

- **Étape 3 : Entraînement du modèle**

À ce niveau, nous devons sélectionner le modèle de machine learning approprié en fonction des objectifs du projet. Puis, passer à la division des données en trois ensembles, à savoir : entraînement, validation et test.

Il faut ensuite entraîner le modèle en utilisant les données d'entraînement et ajuster les paramètres pour l'optimisation des performances. Enfin, il faut évaluer le modèle en utilisant les données de validation et effectuer, si nécessaire, des itérations supplémentaires.

- **Étape 4 : Déploiement en production**

Dans cette dernière étape, il faut préparer le modèle en production en le convertissant au format approprié. L'intégrer ensuite dans l'infrastructure souhaitée, que ça soit un système en ligne, application ou service. Faire les tests et validations supplémentaires pour s'assurer de la fiabilité et bon fonctionnement du modèle et enfin surveiller les performances du modèle déployé et le mettre à jour ou l'ajuster selon la nécessité.

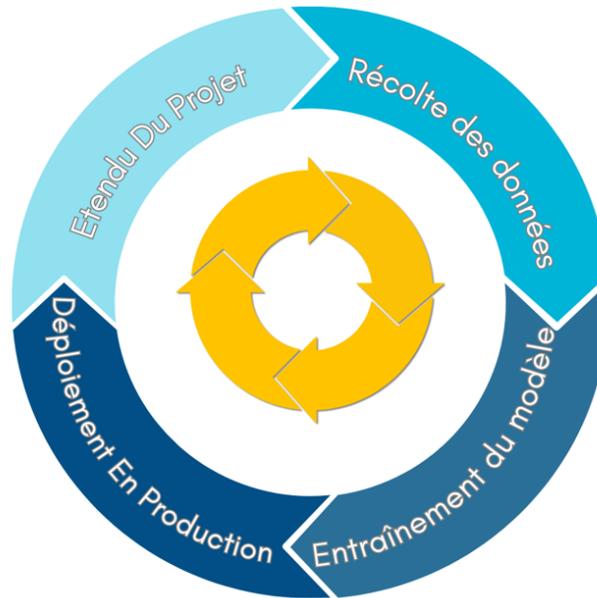


FIGURE 3.1 – Cycle de vie d'un projet machine learning

### 3.4 Comparaison des différents modèles étudiés

#### 3.4.1 Comparaison des modèles de réseaux de neurones utilisés

La recherche et le développement d'un prototype intelligent passent par plusieurs étapes. Nous avons testé plusieurs modèles de réseaux de neurones avant de choisir la solution la plus adéquate.

Voici un tableau comparatif des performances de chaque modèle testé :

Modèle	Backbone (Architecture principale)	Précision mAP <sub>val50-95</sub> <sup>1</sup>	FPS <sup>2</sup>	Avantages	Inconvénients
<b>Yolov5n</b>	CSPDarknet	28.0	1 FPS	Yolov5n est une variante relativement légère de YOLOv5, idéale et efficace pour la Raspberry Pi 4 qui a des ressources matérielles limitées (RAM, pas de carte graphique, etc.). Cette version permet la détection d'objets en temps réel.	L'exécution de Yolov5n sur Pi 4 peut être lente en termes de FPS pour la détection d'objets en temps réel. Consommation d'énergie et accroissement du chauffage de la Raspberry Pi 4.

1. mAP : mean Average Precision value (précision moyenne)

2. FPS : Frames per secondes (Images par secondes)

<b>YoloR</b>	CSPDarknet	55.3	1.6 FPS	L'avantage principal de YoloR réside dans sa précision élevée. YoloR effectue une seule représentation de l'image en utilisant le même modèle, cette représentation peut être utilisée pour servir différentes tâches.	Dans des systèmes embarqués tels que la Raspberry Pi, YoloR offre un nombre de FPS plus petit que d'autres versions de YOLO tels que le v5 cité précédemment. La complexité et les exigences de calcul élevées pour ce modèle, car il combine à la fois des connaissances explicites et implicites à partir de différentes couches.
<b>Faster RCNN</b>	ResNet101	39	1.8 FPS	Précision de détection élevée. Faster RCNN permet de choisir différents backbones tels que ResNet et VGGNet, ce qui offre une certaine flexibilité d'adaptation du modèle sur une Raspberry Pi 4.	Modèle complexe qui nécessite une puissance de calcul et des ressources mémoires significatives. Faible débit d'images par seconde sur Raspberry Pi 4, ce qui peut être un point faible lors du traitement en temps réel.
<b>Mask RCNN</b>	ResNet101	43	2 FPS	Détection et segmentation simultanées, le modèle offre une détection précise des objets. Peut-être utilisé dans plusieurs tâches telles que la détection d'objets, segmentation sémantique et estimation des masques, ce qui offre une polyvalence aux différentes applications sur Raspberry Pi 4.	Le modèle est plus complexe que Faster R-CNN ce qui le rend plus exigeant en termes de puissance de calcul et plus gourmand en ressources, ce qui rend son exécution plus lente sur Raspberry Pi 4. Débit d'images par secondes faible sur Raspberry Pi 4, ce qui limite ses performances en temps réel.
<b>Mobile Net SSD</b> [Modèle adopté]	MobileNet SSD V2	62	[6.92 ; 8] FPS	Efficacité des ressources : le modèle est léger et parfaitement adapté aux systèmes embarqués tels que la Raspberry Pi 4. Grâce à sa légèreté, le modèle fournit plus de FPS sur Raspberry Pi 4, cela permet une détection d'objets plus rapide en temps réel.	Le modèle peut être moins précis en raison de sa légèreté lorsqu'il s'agit de détecter des objets petits. Le modèle est adapté pour la détection d'objets courants, il peut être moins performant lorsqu'il s'agit de détecter des objets moins communs.

TABLE 3.1 – Tableau comparatif des performances des modèles testés avec le modèle adopté.

### 3.4.2 Comparaison des modèles de NLP utilisés

Afin de choisir la solution la plus appropriée et la moins gourmande pour notre unité de calcul, nous avons testé et comparé plusieurs modèles de NLP avant de choisir le modèle à adopter.

Voici un tableau comparatif des architectures des modèles testés, comparées à celle du modèle adopté :

Modèle de NLP	Architecture et mécanisme	Avantages	inconvénients
<b>Bert</b>	Transformer utilisant un mécanisme de décodeur bidirectionnel.	Haute performance due à son entraînement sur de vastes données textuelles. Précis dans la compréhension contextuelle.	Modèle gourmand en utilisation de ressources due à son grand volume. Temps d'inférence lent en raison de sa complexité
<b>TinyBert</b>	Version compressée de Bert, utilise également une architecture transformer.	Modèle compact et léger nécessitant moins de ressources. Bonne conservation des performances malgré sa compression.	Performances légèrement réduites en raison de sa compression. Moins adapté que BERT aux tâches complexes nécessitant une compréhension approfondie du langage.
<b>GPT Neo2</b>	Version optimisée et accélérée de GPT2 utilise une architecture transformer.	Performances optimisées pour le déploiement, il offre des temps d'inférence plus rapides. Permet une utilisation plus efficace des ressources matérielles.	Ne peut pas convenir à toutes les plateformes matérielles, ce qui limite sa portabilité. Une expertise en optimisation matérielle peut être requise, limitant ainsi son accessibilité à certains développeurs/utilisateurs.
<b>GPT2 (Modèle adopté)</b>	Transformer, utilisant un mécanisme de décodeur uniquement.	Haute qualité de génération textuelle. Peut-être utilisé dans une variété de tâches de traitement du langage naturel telles que la génération de texte, la traduction et les résumés, offrant ainsi une grande flexibilité.	Modèle volumineux qui ne peut être utilisé dans certaines ressources informatiques, ce qui limite son utilisabilité. Une attention contextuelle supplémentaire doit être apportée pour filtrer la génération textuelle.

TABLE 3.2 – Tableau comparatif des architectures et performances des modèles NLP testé ainsi que du modèle adopté.

## 3.5 Définition des métriques de performance

Les métriques de performance [3] en apprentissage automatique font référence aux mesures quantitatives utilisées pour évaluer les performances d'un modèle d'apprentissage automatique. Ces métriques sont utilisées pour évaluer la capacité du modèle à prédire les résultats sur un ensemble de données donné. Les métriques de performance courantes en apprentissage automatique comprennent la précision (accuracy), la sensibilité (précision), la spécificité (recall), le score F1, le score AUC-ROC qui sont des métriques de classification et l'erreur quadratique

moyenne (MSE), l'erreur absolue moyenne (MAE), Racine de l'erreur quadratique moyenne (RMSE), R carré (R-squared) qui sont des métriques adaptées aux modèles de régression. entre autres. Le choix des métriques à utiliser dépend du problème spécifique abordé et des objectifs de l'analyse.

### 1. Erreur quadratique moyenne (MSE)

MSE (Mean Squared Error) [3] est une métrique de performance couramment utilisée en apprentissage automatique. Elle mesure la moyenne des différences au carré entre les valeurs prédites et réelles dans un problème de régression. En d'autres termes, elle quantifie la capacité d'un modèle de régression à ajuster les données en comparant les valeurs prédites aux valeurs réelles. Une MSE **1.** plus faible indique un meilleur ajustement du modèle aux données, elle est mathématiquement représentée comme tel :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \dots\dots (1.)$$

- $n$  : est le nombre total d'échantillons.
- $y_i$  : est la valeur réelle de l'échantillon  $i$ .
- $\hat{y}_i$  : est la valeur prédite de l'échantillon  $i$ .
- $(y_i - \hat{y}_i)^2$  : calcule la différence au carré entre la valeur réelle et la valeur prédite de chaque échantillon.

### 2. Erreur absolue moyenne MAE

L'erreur absolue moyenne [3] est la moyenne de la différence entre la vérité terrain et les valeurs prédites. Mathématiquement, il est représenté par :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \dots\dots (2.)$$

- $n$  : est le nombre total d'échantillons.
- $y_i$  : est la valeur réelle de l'échantillon  $i$ .
- $\hat{y}_i$  : est la valeur prédite de l'échantillon  $i$ .
- $|y_i - \hat{y}_i|$  : calcule la différence absolue entre la valeur réelle et la valeur prédite de chaque échantillon.

### 3. L'exactitude

La métrique de performance "accuracy" (exactitude en français) [3] est une mesure courante utilisée pour évaluer la performance d'un modèle de classification en machine learning. Elle représente le pourcentage de prédictions correctes effectuées par le modèle sur l'ensemble des échantillons de données test. Pour la calculer, il suffit de diviser le nombre de prédictions correctes par le nombre total des prédictions puis multiplier par 100.

### 4. Précision

La précision [3] est une métrique de performance couramment pour évaluer la qualité des prédictions positives d'un modèle. Elle mesure le nombre de vrais positifs (observations correctement prédites comme positives) parmi toutes les observations prédites comme positives, La précision est définie comme suit :

$$\text{precision} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}} \dots\dots (4.)$$

où :

- Les vrais positifs sont les observations correctement prédites comme positives.
- Les faux positifs sont les observations incorrectement prédites comme positives.

Une valeur de précision élevée indique que le modèle est précis dans la prédiction des observations positives.

## 5. Sensibilité

La métrique de performance "recall" ou "sensibilité" [3] en français est une mesure utilisée en apprentissage automatique pour évaluer la capacité d'un modèle à identifier tous les exemples positifs d'une classe donnée. En d'autres termes, le recall mesure la proportion d'exemples positifs d'une classe qui ont été correctement identifiés par le modèle. Il est calculé en divisant le nombre de vrais positifs par la somme des vrais positifs et des faux négatifs. Un recall élevé indique que le modèle est capable d'identifier la plupart des exemples positifs, ce qui est particulièrement important pour les problèmes de détection de fraude, de détection de maladies ou de détection d'anomalies.

$$\text{RECALL} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}} \dots\dots(5.)$$

## 6. F1 Score

Le F1 score [3] est une mesure de performance couramment utilisée en classification binaire dans le domaine de l'apprentissage automatique. Il représente la moyenne harmonique de la précision et de recall, qui sont également des mesures de performance importantes. Le F1 score peut être interprété comme une mesure de précision et de la fiabilité du modèle, car il prend en compte à la fois les vrais positifs, les faux positifs et les faux négatifs. Plus le F1 score est proche de 1, plus le modèle est considéré comme précis et fiable.

$$\text{F1-score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \dots\dots(6.)$$

## 3.6 Plan du prototype à mettre en œuvre

Tout système embarqué nécessite la combinaison de matériel et de logiciel lors de sa conception. Dans ce qui suit, nous aborderons les besoins matériels et logiciels, ainsi que les motivations qui nous ont orientés vers ces choix.

### 3.6.1 Besoins matériels

#### 1. Dispositif à porter (Récepteurs de données)

##### (a) Caméra

Motivation : une personne aveugle ou malvoyante a besoin d'une assistance en temps réel, une caméra en mesure de capturer des images ou vidéos dans un environnement dynamique est nécessaire au prototype proposé.

##### (b) Capteur ultrason

Motivation : le capteur ultrason a la capacité de retourner la distance qui sépare le prototype d'un objet, ce qui peut être essentiel pour la prévention d'un danger quelconque envers une personne ayant des problèmes de vision.

#### 2. Dispositif de traitement (Data processor)

##### (a) Raspberry Pi 4B

Motivation : la Raspberry Pi 4B est une unité de calcul abordable et accessible pour de nombreux projets d'IA à petite échelle, elle est suffisamment puissante pour exécuter des tâches d'IA légères à moyenne, elle est flexible, et donne la possibilité de déploiement distribué.

### 3. Dispositif d'entrée audio

#### (a) Microphone

Motivation : ne disposant pas d'une carte son capable de récupérer des données audio de la part de son utilisateur, la Raspberry pi a besoin d'un microphone pour traiter les entrées audios envoyées sous formes de requêtes au système.

### 4. Dispositif de sortie audio

#### (a) Sortie audio du choix de l'utilisateur : Casque, écouteurs.

Motivation : une personne aveugle ou malvoyante doit être prévenue des dangers potentiels qui se trouvent autour d'elle, un feedback audio reste une des meilleures solutions, peu coûteuse, pratique et transportable, elle permet à l'utilisateur d'imaginer ce qui l'entoure.

Dans notre dispositif, nous donnons le choix le plus confortable pour la sortie audio.

## 3.6.2 Besoins Logiciels

### 1. MobileNet SSD V2

Motivation :

- Modèle de détection d'objets peu gourmand en performance pour le déploiement sur Raspberry Pi.
- Le modèle est optimisé, car il est conçu spécialement pour les appareils mobiles et les ressources limitées.
- Architecture légère, ce qui est important pour les limitations de stockage de la Raspberry Pi.
- Vitesse d'inférence rapides sur les appareils mobiles.

### 2. Algorithme de profondeur

Motivation :

- Une personne aveugle ou malvoyante devrait être prévenue de sa proximité avec un objet ou un obstacle afin de garantir sa sécurité, l'utilisation d'un capteur de distance combiné à un algorithme de profondeur permet d'assurer une meilleure expérience utilisateur à ce niveau.

### 3. GPT-2

Motivation : Afin d'offrir une description de scène aux personnes aveugles ou malvoyantes, nous utilisons GPT-2 pour ces raisons :

- NLP peu gourmand pour la description de scène.
- Capacité à générer du texte de haute qualité.
- Compréhension sémantique avancée.
- Évolution et amélioration continue des modèles GPT.

### 4. Google Tesseract

Motivation :

- Une solution flexible, précise et adaptable sur plusieurs langues lui permettant d'être déployé au niveau international.
- OCR léger et facilement déployable pour la lecture de texte récupéré par la caméra sur Raspberry Pi.
- Reconnaissance de texte précise.
- Google Tesseract prend en charge un large éventail de langues et scripts, ce qui assure son adaptabilité.

## 3.7 Méthode et implémentation de l'ultrason

### 3.7.1 Les capteurs ultrason

La définition d'un son est une vibration mécanique d'un fluide se propageant sous forme d'onde (par exemple dans l'air). L'oreille humaine n'est capable de percevoir qu'une seule partie de ces fréquences, à savoir le spectre sonore. Les sons à basse fréquence sont appelés "infrasons" et ceux à haute fréquence "ultrasons".

Les capteurs à ultrasons sont conçus pour la détection des objets à proximité en réfléchissant les ultrasons comme un radar. Le principe est simple, il suffit de calculer la distance de l'obstacle à partir du temps mis par l'onde sonore pour se déplacer dans l'environnement, passer par une réflexion de l'onde sur l'obstacle et retourner jusqu'au capteur.

Les capteurs à ultrasons sont conçus pour la détection des objets à proximité en réfléchissant les ultrasons comme un radar. Le principe est simple, il suffit de calculer la distance de l'obstacle à partir du temps mis par l'onde sonore pour se déplacer dans l'environnement, passer par une réflexion de l'onde sur l'obstacle et retourner jusqu'au capteur.

#### 1. Pont diviseur de tension

Un pont diviseur de tension se constitue de deux résistances R1 et R2 montées en série et connectées à une tension d'entrée  $V_{in}$ . La tension de sortie  $V_{out}$  est abaissée en fonction de la valeur des résistances utilisées.

Pour notre circuit, on a une tension d'entrée de 5 V ( $V_{in}$ ) générée par la broche Echo et devra être abaissée jusqu'à 3,3 V ( $V_{out}$ ).

Pour cela, nous avons utilisé une résistance de 1 k $\Omega$  et une résistance de 2 k $\Omega$ .

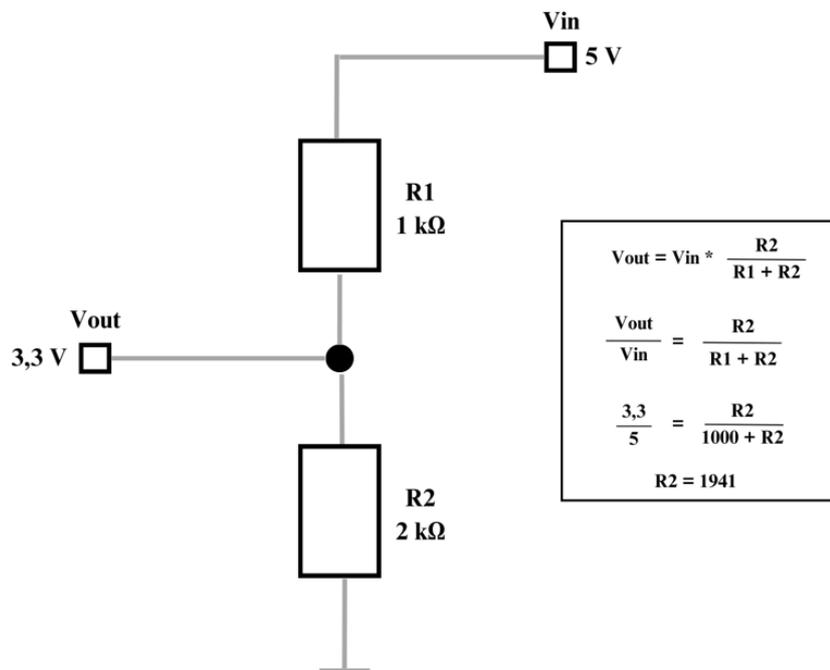


FIGURE 3.2 – Pont diviseur de tension.

#### 2. Matériel nécessaire

- Un module ultrasons HC-SR04.
- Une résistance d'un (1) k $\Omega$ .
- Une résistance de deux (2) k $\Omega$ .
- Des câbles de connexion.

Le module à ultrasons HC-SR04 utilisé possède quatre broches :

- Une broche (Gnd) pour mettre le module à la masse (0 V).
- Une broche de sortie (Echo) pour informer de la fin de l'émission du train d'ultrasons et de son retour après réflexion sur l'obstacle.
- Une broche d'entrée (Trig pour Trigger) pour déclencher l'émission du train d'ultrasons.
- Une broche (Vcc) pour alimenter le capteur en 5 V.

### 3. Plan du montage de l'ultrason

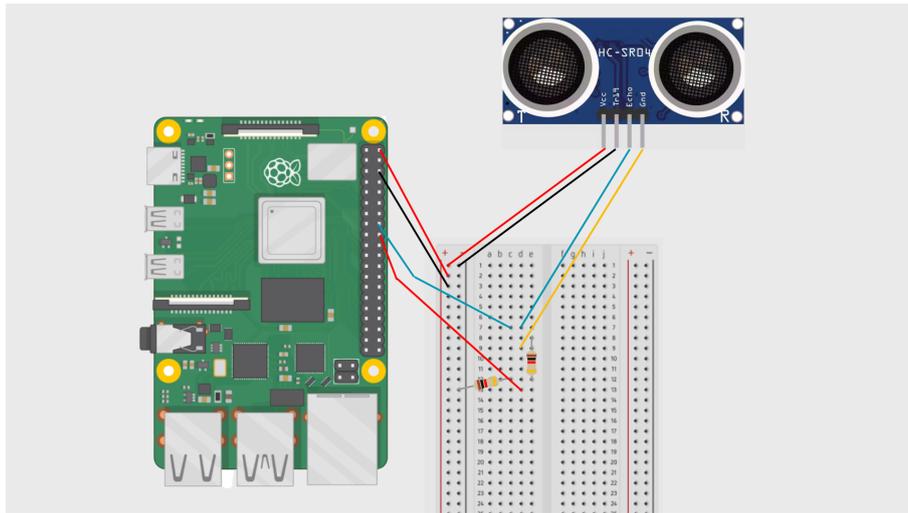


FIGURE 3.3 – Plan du montage du capteur ultrason

### 4. Algorithme de calcul de distance de l'ultrason

Voici un aperçu de l'algorithme utilisé pour récupérer la distance des objets en face de l'ultrason en langage Python :

```

1 import RPi.GPIO as GPIO
2
3 import time
4
5
6 def measure_distance(trigger_pin, echo_pin):
7
8     # Set GPIO Pins
9
10    GPIO_TRIGGER = 23
11
12    GPIO_ECHO = 24
13
14
15
16    # GPIO Mode (BOARD / BCM)
17
18    GPIO.setmode(GPIO.BCM)
19
20
21
22    # Set GPIO direction (IN / OUT)
23
24    GPIO.setup(trigger_pin, GPIO.OUT)
25
26    GPIO.setup(echo_pin, GPIO.IN)
27
28
29
30    def distance():
31
32        # Set Trigger to HIGH

```

```
33
34     GPIO.output(trigger_pin, True)
35
36
37
38     # Set Trigger after 0.01ms to LOW
39
40     time.sleep(0.00001)
41
42     GPIO.output(trigger_pin, False)
43
44
45
46     StartTime = time.time()
47
48     StopTime = time.time()
49
50
51
52     # Save StartTime
53
54     while GPIO.input(echo_pin) == 0:
55
56         StartTime = time.time()
57
58
59
60     # Save time of arrival
61
62     while GPIO.input(echo_pin) == 1:
63
64         StopTime = time.time()
65
66
67
68     # Time difference between start and arrival
69
70     TimeElapsed = StopTime - StartTime
71
72     # Multiply with the sonic speed (34300 cm/s)
73
74     # and divide by 2, because there and back
75
76     distance = (TimeElapsed * 34300) / 2
77
78
79
80     return distance
81
82
83
84     try:
85
86         # Calculate distance once
87
88         dist = distance()
89
90         return(dist)
91
92         #print("Measured Distance = %.1f cm" % dist)
93
94
95     except KeyboardInterrupt:
96
97         print("Measurement stopped by User")
98
99
100     finally:
101
102         GPIO.cleanup()
```

## 3.8 Manipulation des modèles

### 3.8.1 MobileNet SSD V2

#### 1. Architecture

##### (a) Architecture principale de MobileNet V2 [40]

MobileNet V2 utilise des convolutions séparables en profondeur, son bloc principal est schématisé ci-dessous dans la figure 3.3, le bloc contient trois couches de convolution ; une convolution en profondeur qui filtre les entrées, suivie d'une de convolution 1x1, cette couche réduit le nombre de canaux, elle est appelée aussi couche de projection, car elle projette les données ayant un grand nombre de dimensions (canaux) dans un tensor avec un nombre beaucoup plus faible d dimensions.

Par exemple, la couche en profondeur peut travailler sur un tensor avec 144 canaux, que la couche de projection réduira ensuite à seulement 24 canaux. Cette couche est aussi appelée couche d'étranglement, car elle réduit la quantité de données dans le réseau.

La première couche est aussi une convolution 1x1, son rôle est d'augmenter le nombre de canaux des données avant qu'elles n'entrent dans la couche convolution en profondeur, ce qui fait que cette couche d'expansion a toujours plus de canaux en sortie qu'en entrée.

L'expansion de données se fait grâce à un hyperparamètre qui permet d'expérimenter différentes options d'architecture, il est paramétré par défaut à 6.

L'entrée et la sortie du bloc sont des tensors de basse dimension, tandis que l'étape de filtrage qui se produit à l'intérieur du bloc est effectuée en tensor de haute dimension.

Parmi les nouveautés qui distinguent MobileNet V2 de la version 1, on peut citer la connexion résiduelle. Cela fonctionne comme dans ResNet, elle permet de faciliter le flux de gradients à travers le réseau.

Chaque couche est normalisée en lots et la fonction d'activation est ReLU6. Cependant, la couche de projection n'a pas de fonction d'activation.

L'architecture complète de MobileNet V2 est donc composée de 17 de ces blocs de construction à la suite. Cela est suivi d'une convolution  $1 \times 1$  classique, d'une couche de pooling moyenne globale et d'une couche de classification. (Petit détail : le tout premier bloc est légèrement différent, il utilise une convolution  $3 \times 3$  classique avec 32 canaux au lieu de la couche d'expansion).

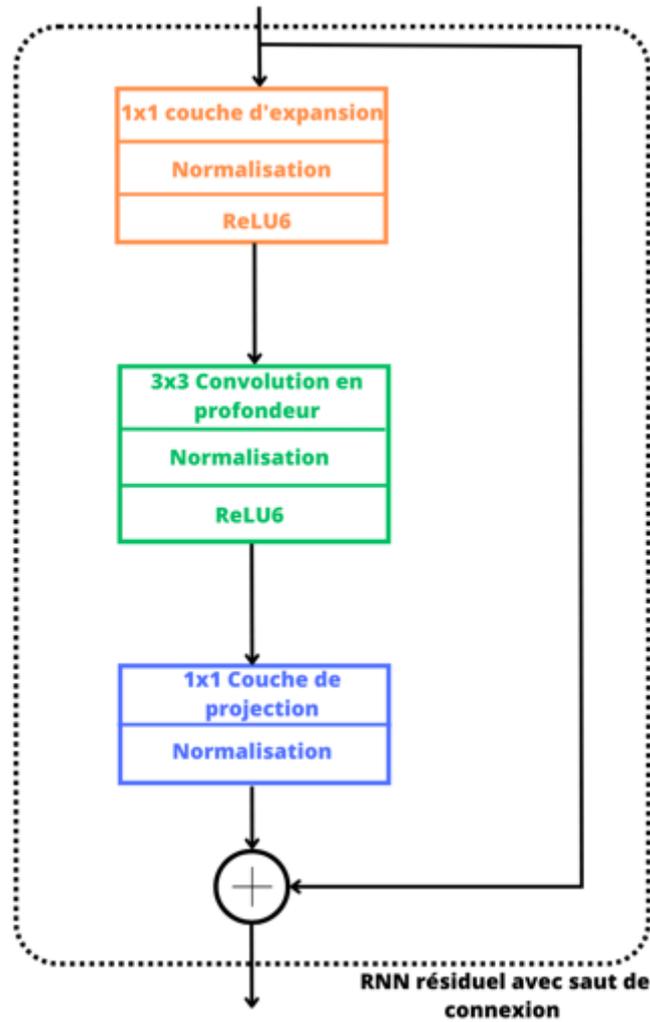


FIGURE 3.4 – Architecture principale de MobileNet V2.

## (b) Entrée des données dans le réseau

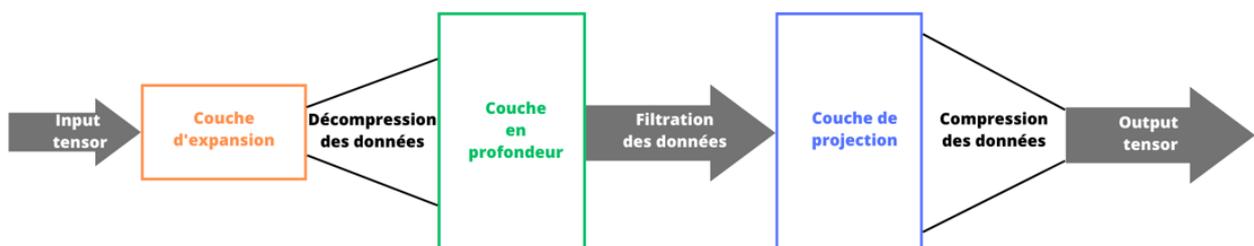


FIGURE 3.5 – Entrée des tensors dans le réseau

Les données de basse dimension qui circulent entre les blocs sont une version compressée de données réelle afin d'appliquer des filtres à ces données, c'est pourquoi la couche d'expansion agit comme un décompresseur (unzipper) qui rétablit les données dans leur forme initiale, la couche en profondeur s'occupe ensuite de réaliser les filtrages nécessaires à cette étape du réseau, enfin, la couche de projection compresse à nouveau les données pour les rendre plus petites. Tout cela est possible grâce au fait que

les couches expansions et projection soient réalisées à l'aide de couches convolutionnelles avec des paramètres ajustables de sorte que le modèle soit capable d'apprendre comment décompresser au mieux les données à chaque étape du réseau.

(c) **Architecture du classificateur de MobileNet**

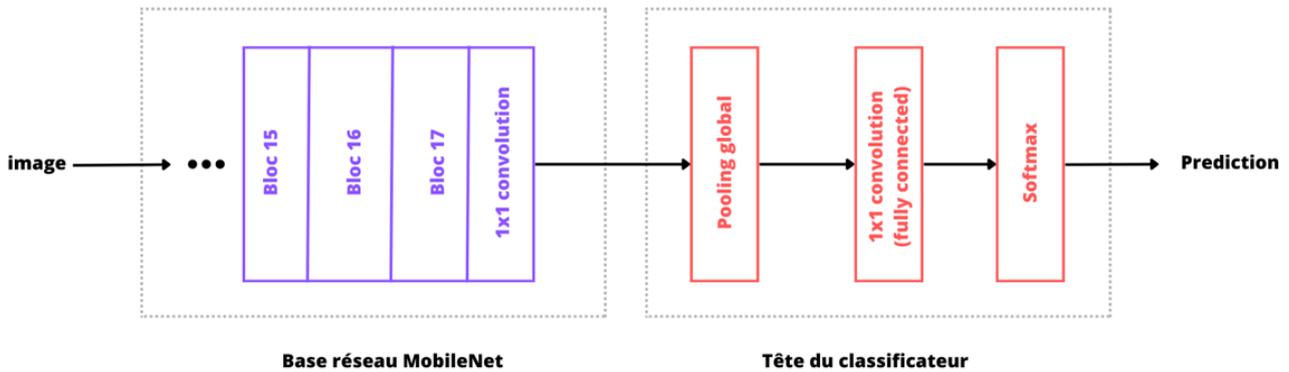


FIGURE 3.6 – Architecture du classificateur de MobileNet.

Lorsqu'on fait une classification, la sortie du réseau de base est généralement une image de  $7 \times 7$  pixels. Le classificateur utilise d'abord une couche de pooling global pour réduire la taille de  $7 \times 7$  à  $1 \times 1$  pixel - en prenant essentiellement un ensemble de 49 prédicteurs différents - suivi d'une couche de classification et d'un softmax.

(d) **Architecture de la partie SSD de MobileNet**

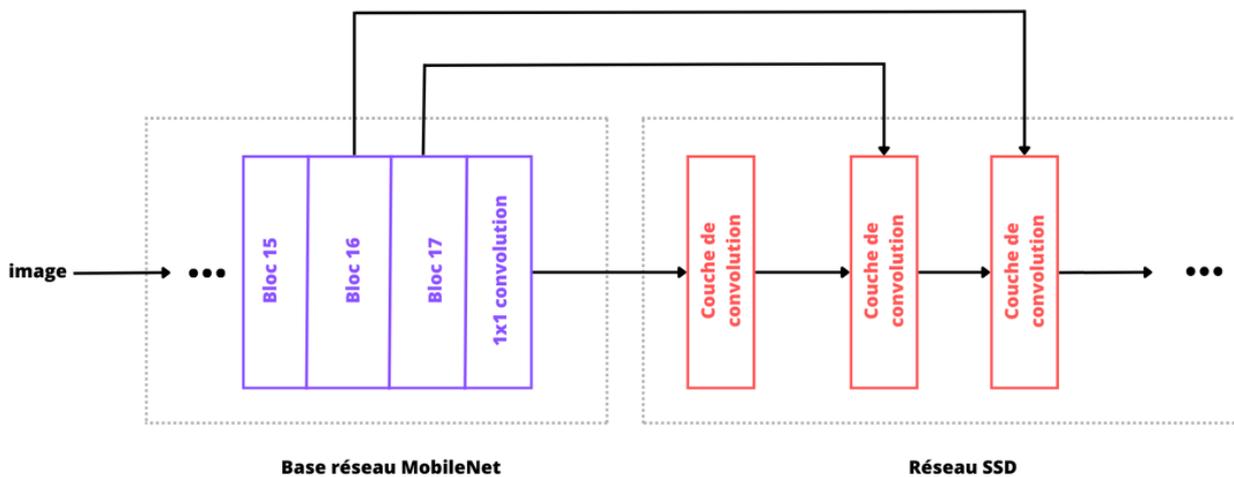


FIGURE 3.7 – Architecture de la partie SSD de MobileNet

Pour utiliser quelque chose comme SSDLite avec MobileNet, la sortie de la dernière couche du réseau de base et les sorties de plusieurs couches précédentes sont prises pour alimenter les couches SSD. Le rôle des couches MobileNet est de convertir les pixels de l'image passée en entrée en caractéristiques décrivant le contenu de l'image, puis de transmettre aux autres couches. Ainsi, MobileNet est utilisé ici comme extracteur de caractéristiques pour un deuxième réseau de neurones.

Dans le cas de la détection d'objets avec SSD, nous voulons connaître les caractéristiques de haut niveau et les caractéristiques de niveau inférieur, c'est pourquoi nous lisons également à partir des couches précédentes. SSD ajoute de nombreuses couches

de convolution supplémentaires au-dessus du réseau de base. Il est donc important d'avoir un extracteur de caractéristiques rapide et c'est exactement ce que propose MobileNet V2.

## 2. Détection d'objet

### • Qu'est-ce qu'une détection d'objet ?

La détection d'objet (ou Object detection) est une technique de vision par ordinateur qui vise la localisation et l'identification d'un objet spécifique dans une image ou dans une vidéo, cette méthode utilise des algorithmes et des modèles d'apprentissage automatique tels que les CNN ou les R-CNN afin d'extraire les caractéristiques des images, une prédiction est ensuite retournée sur l'objet identifié avec des délimitations rectangulaires appelées boîtes englobantes, des étiquettes ou des classes d'objets sont ensuite attribuées au résultat.

## 3. Bounding box (Boîte englobante)

La technique 'bounding box' (ou boîte englobante) utilisée dans MobileNet SSD v2 est une méthode de détection d'objet qui délimite les objets présents dans une image ou vidéo, la technique consiste à :

- Une image est passée en entrée au réseau de neurone MobileNet afin d'extraire les caractéristiques et les informations importantes telles que les formes, les textures et les structures de l'objet.
- Le module de détection SSD ( Single Shot MultiBox Detector) intégré au réseau MobileNet utilise les caractéristiques extraites afin d'analyser les différentes échelles et positions de l'image pour détecter l'objet présent, par la suite les boîtes englobantes entrent en jeu pour délimiter les régions où se trouvent les objets présents dans l'image.
- Chaque boîte englobante est définie par quatre coordonnées : les coordonnées du coin supérieur gauche et du coin inférieur droit de la boîte.
- Après avoir défini les boîtes englobantes, MobileNet SSD v2 prédit les classes d'objets correspondants en estimant une probabilité sur chaque objet présent dans l'image.

### 3.8.2 Google's Tesseract

Tesseract est un moteur de reconnaissance Optique de caractères (OCR) open-source développé par Google. Il utilise des algorithmes pour analyser des images afin d'en extraire des données textuelles potentiellement présentes.

#### 1. Architecture

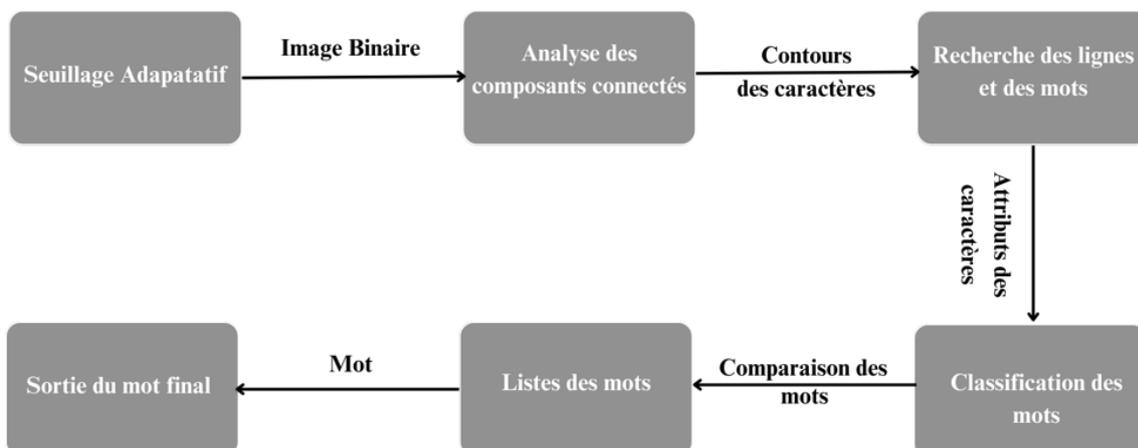


FIGURE 3.8 – Architecture de Google's Tesseract.

## 2. Modes de moteur OCR

Nous retrouvons quatre (04) moteurs OCR ( ou OCR Engine Modes OEM) au niveau de Tesseract :

- 0- Legacy engine only
- 1- Neural nets LSTM engine only
- 2- Legacy + LSTM engines
- 3- Default

## 3. Modes de segmentation de page

Pour la segmentation de page (ou Page Segmentation Modes PSM), Tesseract a quatorze (14) modes :

- 0- Détection de script et d'orientation (Orientation and script detection ou OSD)
- 1- Segmentation automatique de page avec OSD
- 2- Segmentation automatique de la page sans OSD ou OCR
- 3- Segmentation complète automatique de la page sans OSD (par défaut)
- 4- Prendre une seule colonne de texte de tailles variables.
- 5- Prendre un seul bloc uniforme de texte aligné verticalement.
- 6- Prendre un seul bloc de texte uniforme
- 7- Traiter l'image comme une seule ligne de texte
- 8- Traiter l'image comme un seul mot
- 9- Traiter l'image comme un seul mot en cercle
- 10- Traiter l'image comme un seul caractère
- 11- Texte éparpillé. Trouver autant de texte que possible dans aucun ordre possible
- 12- Texte éparpillé avec OSD
- 13- Ligne brute. Traiter l'image comme une seule ligne de texte, ignorant les hacks (ou aides) spécifiques à Tesseract.

La combinaison de chacun des OEM et PSM nous permettra d'avoir des résultats différents, chacune plus adaptés les uns des autres aux situations auxquelles il faille faire face.

### 3.8.3 GPT-2

#### i. Architecture [24]

GPT-2 utilise une architecture Transformer qui est un réseau de neurones basé sur l'attention. Il se compose de plusieurs couches d'encodeurs Transformer qui permettent de capturer les relations à longue distance entre les mots dans le texte.

L'attention est une composante clé du Transformer. Elle permet au modèle de tenir compte des dépendances entre tous les mots d'une phrase lors du calcul de représentations contextuelles. L'attention multi-têtes permet d'explorer différentes relations entre les mots simultanément.

GPT-2 empile plusieurs couches d'encodeurs Transformer, généralement 12 ou 24 couches. Chaque couche d'encodeurs comprend une sous-couche d'attention multi-têtes, une sous-couche de feed-forward, et des connexions résiduelles avec des couches de normalisation.

GPT-2 est un modèle "sans état", ce qui signifie qu'il ne garde pas de mémoire ou d'information contextuelle à long terme. Il génère des prédictions uniquement en fonction du

contexte précédent dans la séquence de texte.

GPT-2 est pré-entraîné sur de grandes quantités de données textuelles non supervisées. Il utilise un objectif d'apprentissage de langage où il doit prédire le mot suivant dans une séquence de texte donnée. Cela permet au modèle de capturer les structures et les règles de la langue.

Après l'entraînement préalable, GPT-2 peut être adapté à des tâches spécifiques en effectuant un fine-tuning sur des données supervisées. Cela permet de spécialiser le modèle pour des tâches telles que la génération de texte, la traduction ou la réponse à des questions spécifiques.

GPT-2 est connu pour sa capacité à générer des textes de manière cohérente et créative. Cependant, il peut également présenter des limitations, notamment en termes de biais et de compréhension du contexte.

### 3.9 Bibliothèques utilisées

#### 1. **Tensorflow**

Bibliothèque d'apprentissage automatique (machine learning) développée par Google. Elle propose une API de haut niveau pour la définition et l'entraînement de modèles de réseaux de neurones. Cette bibliothèque prend en compte des langages de programmation tels que python, C++ et Java. C'est l'une des bibliothèques d'apprentissage les plus utilisées et les plus populaires du moment.

#### 2. **Tensorflow Lite**

Version légère de la bibliothèque Tensorflow, spécialement conçue pour les ressources mobiles et systèmes embarqués, elle permet l'exécution de modèles peu gourmands en énergie tels que les smartphones, tablettes et objets connectés.

#### 3. **PyTorch**

Bibliothèque populaire d'apprentissage approfondi open source, elle propose un framework de calcul dynamique pour les réseaux de neurones. Largement utilisée dans le domaine de la recherche, nous avons utilisé PyTorch pour les modèles MobileNet SSD V2 ainsi que GPT-2.

#### 4. **Keras**

C'est une interface haut niveau pour les réseaux de neurones en Python. Elle s'exécute au-dessus de bibliothèques d'apprentissage profond telles que TensorFlow et fournit une interface simple et ergonomique pour la construction et l'entraînement de réseau de neurones.

#### 5. **PyTesseract (Python)**

Module permettant d'utiliser Tesseract, un moteur de reconnaissance optique de caractères (OCR) développé par Google. Facilitant l'extraction de texte à partir d'images en utilisant les capacités de Tesseract, il est couramment utilisé pour l'OCR en Python.

#### 6. **Hugging Face Transformers**

Bibliothèque qui propose une interface simple et facile à utiliser pour travailler avec des modèles basés sur les transformers. Construite sur les frameworks de TensorFlow et PyTorch, Hugging Face facilite le chargement, fine-tuning et utilisation de modèles NLP pré-entraînés.

#### 7. **OpenCV**

Open Source Computer Vision Library est une bibliothèque influente dans le domaine de l'imagerie et de la vision par ordinateur, elle est utilisée dans le traitement et la manipulation de données visuelles. Sa flexibilité, sa performance et sa particularité à être multi-plateforme justifient sa popularité.

### 3.10 Plan du montage du projet



FIGURE 3.9 – Montage du prototype.

Dans cette figure, nous présentons le prototype monté, il est composé d'un capteur ultrason relié à une Raspberry Pi 4B avec des câbles pins.

Ces composants sont les acteurs principaux du côté traitement.

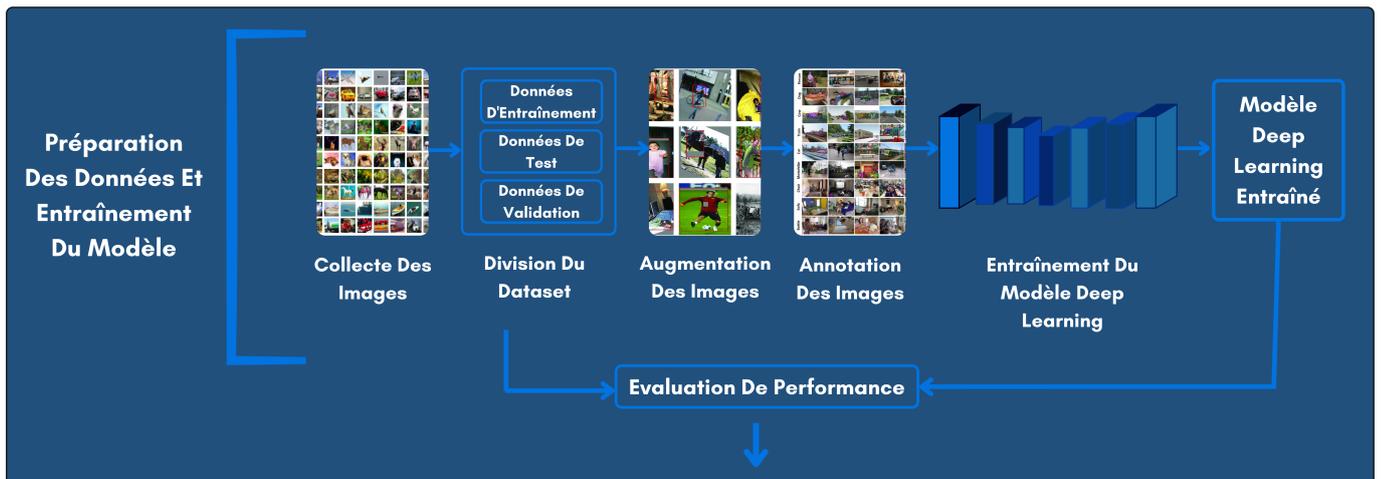


FIGURE 3.10 – Entraînement du modèle.

Dans cette figure, nous avons illustré la partie 'Préparation et Entraînement du modèle MobileNet V2'.

Les images collectées sont divisé 3 parties : Données d'entraînement, données de validation et données test.

Le modèle effectue ensuite une augmentation puis une labellisation des images.

Le modèle est entraîné et à l'issue de cet entraînement, nous évaluons ses performances.

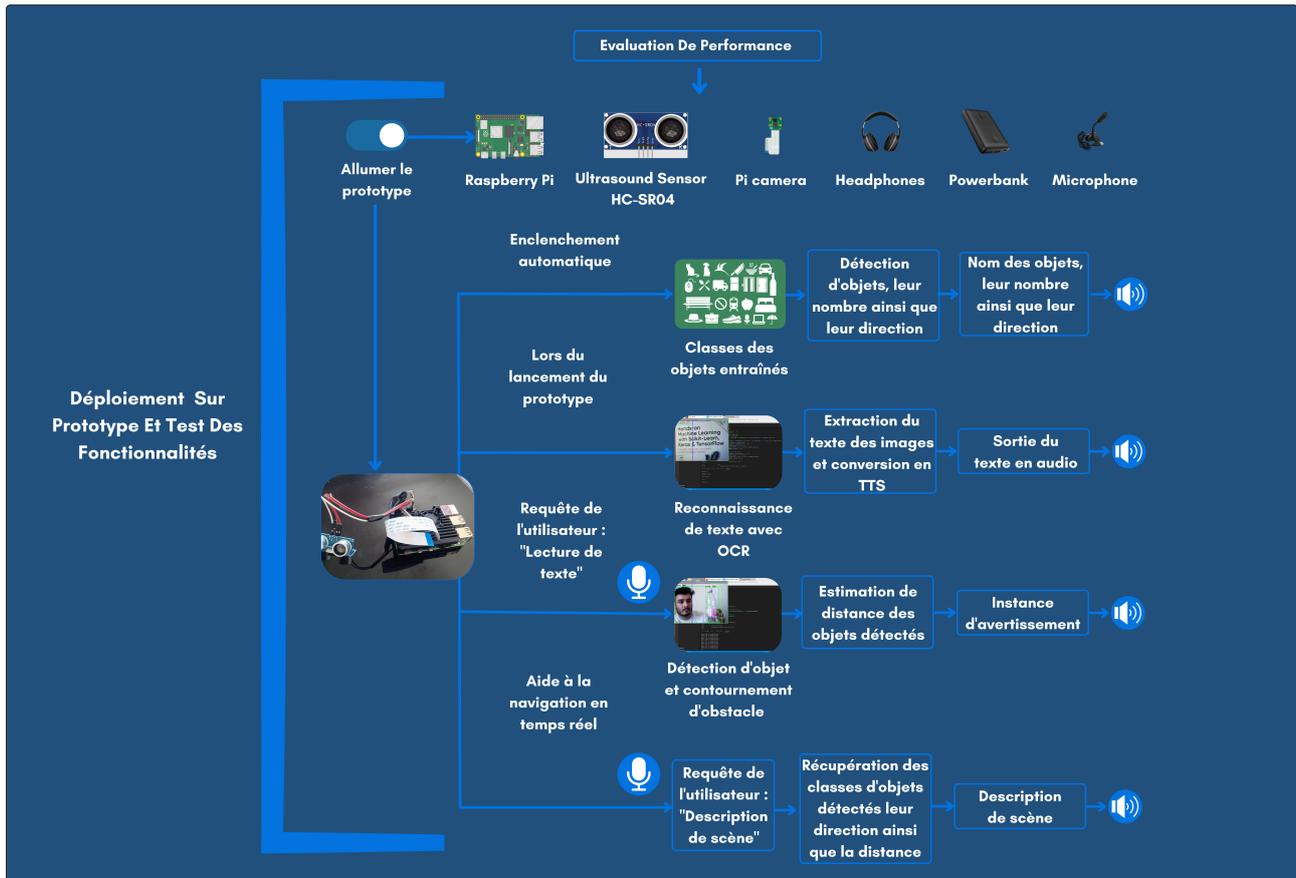


FIGURE 3.11 – Plan du fonctionnement du prototype

Dans cette figure, nous illustrons le déploiement du modèle sur le prototype.

Les différentes fonctionnalités du système sont lancées après sa mise en marche. Nous en parlons dans ce qui suit.

### 3.11 Fonctionnalités du prototype

- **Détection d'objets**

Le prototype développé et entraîné sur le dataset COCO est capable de détecter différents objets communs, ce qui peut être bénéfique à l'utilisateur dans sa familiarisation avec l'environnement dans lequel il se trouve.

- **Définition de la direction de l'objet détecté**

Le champ de vision de la caméra est divisé en trois (3) parties ; Gauche, droite et milieu. L'utilisateur aura donc un retour sur la position exacte de l'objet.

Par exemple, si l'objet détecté se situe à droite, le système renvoie : Classe d'objet, droite, distance. (même chose pour la gauche). Si l'objet se trouve au milieu, le système renvoie : Classe d'objet, en face, distance.

- **Assertion de la distance de l'objet en face de l'utilisateur**

Le prototype développé est muni d'un capteur ultrason, ce dernier est capable de renvoyer à l'utilisateur les objets se trouvant à une distance de moins de 5 mètres.

- **Description dynamique de la scène à la demande de l'utilisateur**

Le prototype a la faculté de faire une description de scène sous requête de l'utilisateur, lorsqu'il envoie une requête audio grâce au microphone intégré, le système lui renvoie tout ce qui apparaît dans la caméra.

- **Avertissement de l'utilisateur et aide à la navigation**

Le prototype détecte les obstacles et danger potentiels présents autour de lui puis renvoie des avertissements à l'utilisateur.

- **Détection et lecture des données textuelles à la demande de l'utilisateur**

Le prototype a la faculté de lire une donnée textuelle sous requête de l'utilisateur. En effet, il est possible d'envoyer des requêtes audio via le microphone, ainsi le système déclenche l'exécution d'une tâche speech to text. Ensuite, segmentation et extraction du text de l'image prise au moment de la requête utilisateur avant de renvoyer le tout à travers une tâche de text to speech.

À noter que toutes les fonctionnalités s'exécutent en temps réel dans un intervalle de [4.2, 5.2[ FPS, sur une résolution de 640x480.

### 3.12 Diagramme de cas d'utilisation

La figure ci-dessous résume les différentes fonctionnalités citées précédemment :

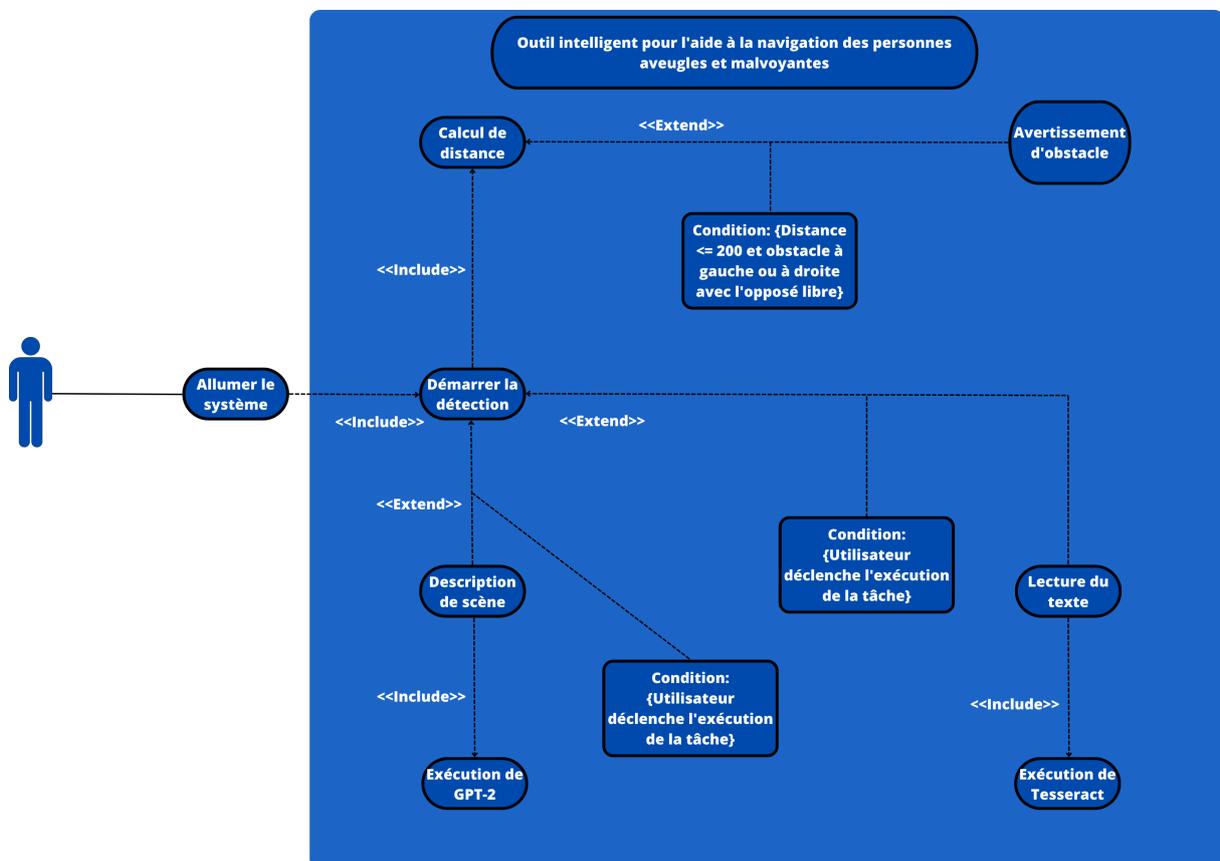


FIGURE 3.12 – Diagramme de cas d'utilisation

Lorsque l'utilisateur allume le système la détection se lance en premier, lorsque la distance est inférieure ou égale à 200 cm, le système renvoie la position de l'objet détecté ainsi que la distance qui le sépare de l'utilisateur.

L'utilisateur peut déclencher l'exécution d'autres tâches telles que la description de scène (grâce à GPT) et la lecture d'une donnée textuelle (grâce à Tesseract) en lançant une requête audio (Speech to text).

## Conclusion

Dans ce chapitre, nous avons proposé une approche motivée par les objectifs du projet en prenant en compte les ressources matérielles disponibles et utilisables, nous avons ainsi exposé des comparaisons de la solution proposée avec d'autres travaux réalisés préalablement en justifiant notre choix. Dans le chapitre suivant, nous parlerons de l'évaluation et discuterons les résultats obtenus.

# Chapitre 4

## Évaluation et tests

### 4.1 Introduction

Dans ce chapitre, nous allons discuter de l'usage du dataset choisi, des versions des logiciels que l'on a choisi d'utiliser ainsi que des résultats obtenus. Nous passerons par un retour sur l'expérience utilisateur et également la gestion budgétaire du projet. Pour finir, nous ouvrirons un débat sur les avantages, inconvénients et futures améliorations que nous envisageons sur ce projet.

## 4.2 Motivation sur l'usage de COCO

Lors de l'entraînement du modèle que nous avons choisi pour notre projet, le dataset COCO 2017 qui est une version récente ayant été utilisé pour mener à bien la détection d'objets, fût choisi pour les motivations suivantes :

- Coco contient plus de 330 000 images annotées, et 80 catégories d'objets différents ce qui donne accès à une large variété d'objets communs.
- Annotation (étiquetage) de haute qualité.
- Le dataset Coco permet également la réalisation d'autres tâches telles que la segmentation et la prédiction d'objets, ce qui offre la possibilité de développer des modèles plus avancés et complexes.
- Utilisation académique et industrielle.
- Coco est utilisé par une large communauté et c'est aussi un dataset populaire, ce qui permet de bénéficier de l'expérience acquise par d'autres utilisateurs.

## 4.3 Versions des logiciels

Software	Version	Avantages	Limites
MobileNet SSD	ssd-mobilenet-v2-320x320	Vitesse de traitement relativement bonne par rapport à la légèreté de son architecture.	Précision inférieure en comparaison avec des modèles plus grands, les objets moins communs peuvent être difficiles à détecter avec précision.
OCR (Optical Character recognition)	Tesseract 5.0.0	La version 5.0.0 est plus stable, elle a permis d'améliorer la précision de la reconnaissance du texte.	Cette version peut nécessiter des prétraitements supplémentaires sur les images contenant du bruit ou des polices de caractères moins courantes.
GPT (Generative Pre-trained Transformer)	GPT 2.0	Génération de texte de haute qualité.	Nécessite une quantité de données massive ainsi qu'une ressource de calcul assez puissante.
Python (Programming Language)	Python 3	Python 3 offre une Disposition d'une large bibliothèque étendue qui offre de nombreuses fonctionnalités prêtes à l'emploi, ce qui offre un gain en temps considérable.	Pour les tâches demandant un calcul intense, il peut être moins performant que les langages compilés.
Operating system	Kernel version : 6.1. Debian version : 11 (bullseye).	Support plusieurs types de codes et prend en charge plusieurs langages de programmation.	L'utilisation des formats .exe n'est pas prise en charge par les OS de la Raspberry pi, ce qui peut être une limite d'utilisation pour certains développeurs.

TABLE 4.1 – Tableau comparatif des différents logiciels utilisés

## 4.4 Résultats Obtenus

### 4.4.1 Détection

Dans cette section, nous présentons des captures d'écran illustrant le bon fonctionnement de notre programme de détection. Les figures suivantes démontrent les différentes étapes du processus de détection et mettent en évidence les résultats obtenus.

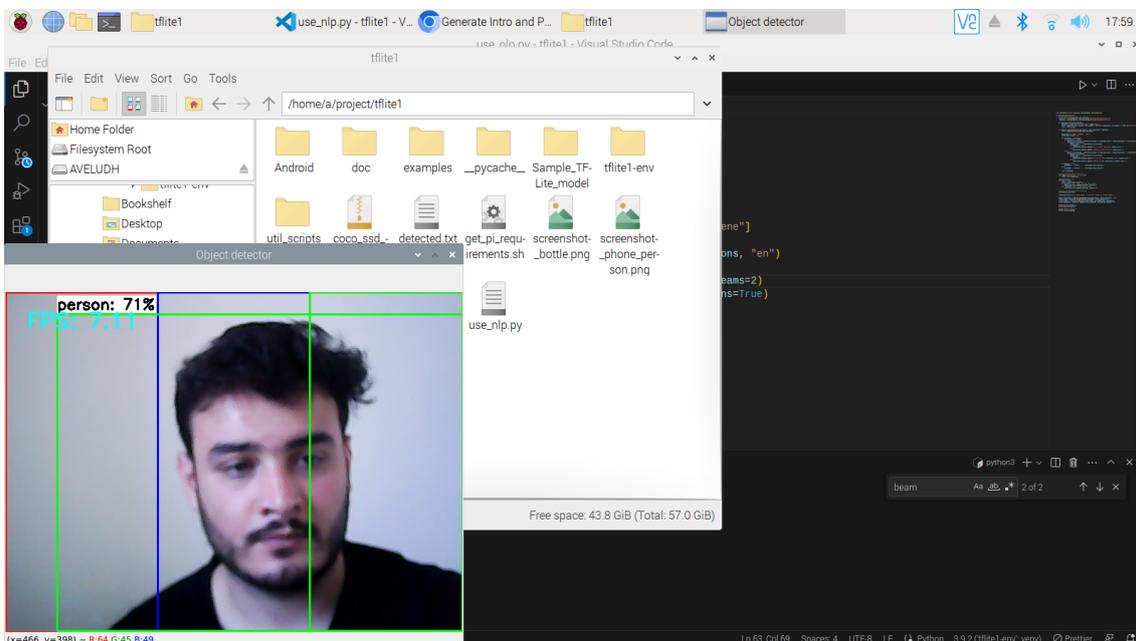


FIGURE 4.1 – Capture d'écran de l'interface utilisateur illustrant les résultats de la détection d'une classe

Dans la figure 4.1, nous avons un retour de 7.11 FPS, une classe personne détectée à 71%, nous pouvons également voir que l'écran est divisé en 3 parties afin de connaître la position de la classe détectée (Gauche, droite, centre).

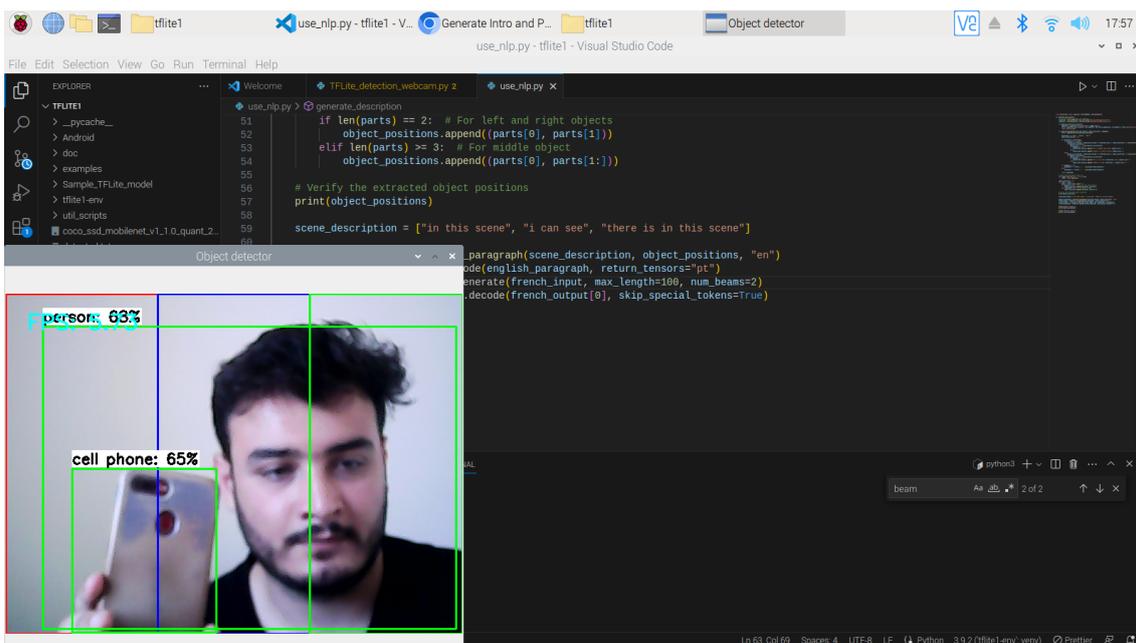


FIGURE 4.2 – Capture d'écran de l'interface utilisateur illustrant les résultats de la détection de plus d'une classe

Dans la figure 4.2, nous avons un retour de 5.73 FPS, une classe personne détectée à 63% et une autre classe cell phone détectée à 65%, le programme est donc capable de détecter plus d'une classe d'objets présents dans une image ou vidéo.

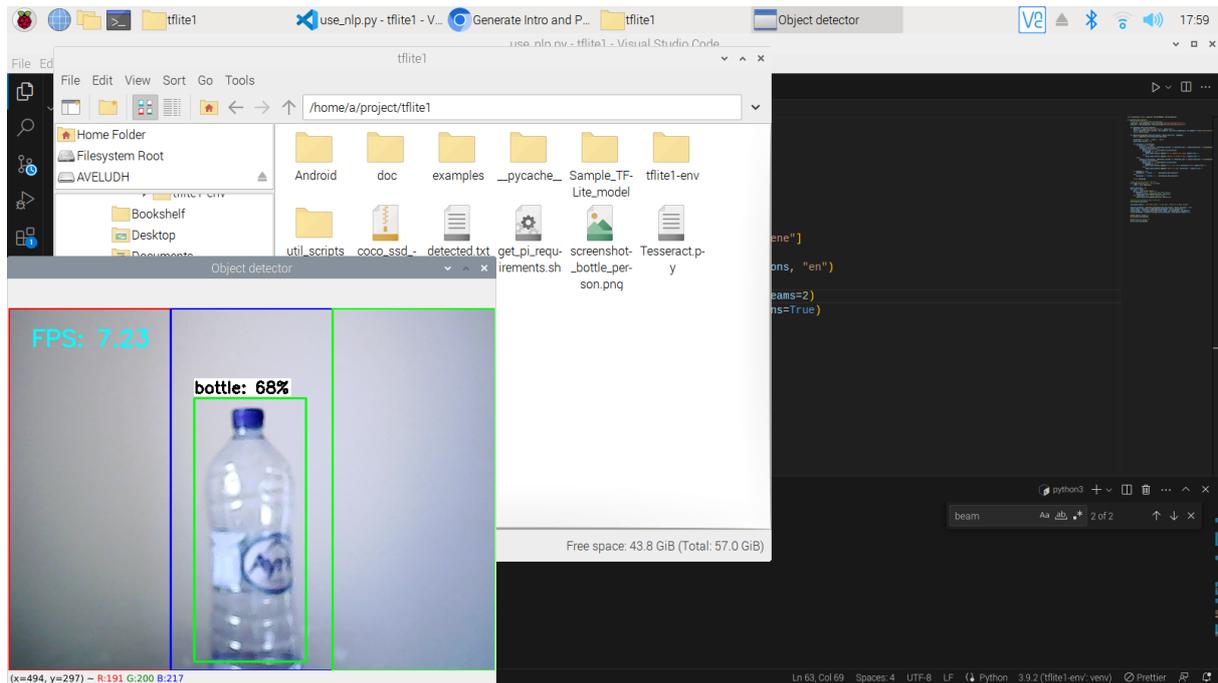


FIGURE 4.3 – Autre exemple illustrant les résultats de la détection d'une classe

La figure 4.3 illustre un autre exemple fournissant un aperçu visuel du bon fonctionnement de notre programme de détection et mets en évidence sa capacité à identifier avec précision les objets d'intérêt.

Les résultats présentés renforcent la fiabilité de notre solution et témoignent de son potentiel dans les systèmes intelligents embarqués.

#### 4.4.2 Temps de réponse du système

En général, le temps de réponse d'un système embarqué intelligent est un indicateur de sa capacité à traiter rapidement les données d'entrée et à générer une sortie pertinente. Pour de nombreux systèmes embarqués, en particulier ceux utilisés dans des applications en temps réel, un temps de réponse rapide est souvent critique pour assurer des performances optimales.

Nous listons ci-dessous le temps de réponse de chaque processus de notre système après son lancement et avant son optimisation.

Processus	Temps de réponse en seconde (s)
Lancement du système	2s
NLP	42s
OCR	3s
Calcul de distance	En temps réel
Détection d'objets	En temps réel
Avertissement	En temps réel

TABLE 4.2 – Tableau résumant le temps de réponse de chaque processus au lancement du système avant optimisation de l'algorithme

Le temps de réponse d'un système embarqué intelligent peut varier en fonction de plusieurs facteurs, tels que la complexité de la tâche à effectuer, la puissance de calcul du système, la qualité de l'algorithme utilisé, la quantité de données à traiter, etc.

Lors de notre premier essai, nous avons utilisé la génération textuelle à partir du NLP en deux langues (français et anglais). Dans la version la plus récente, nous avons choisi d'inclure l'option de traduction tout en conservant le texte initialement généré en anglais. Cette décision nous a permis de gagner 30 secondes de temps d'attente.

Voici les résultats que nous avons obtenus après avoir amélioré la qualité de l'algorithme en l'optimisant.

Processus	Temps de réponse en seconde (s)
Lancement du système	13.31
NLP	9.83s
OCR	1s
Calcul de distance	En temps réel
Détection d'objets	En temps réel
Avertissement	En temps réel

TABLE 4.3 – Tableau résumant le temps de réponse de chaque processus au lancement du système après optimisation de l'algorithme

Dans cette version, le temps de démarrage du système est supérieur à celui enregistré précédemment, car il faut maintenant prendre en compte le temps nécessaire à l'installation des packages. Néanmoins, le temps de réponse du NLP a considérablement diminué, et nous avons gagné deux secondes sur le temps de réponse de l'OCR.

Ces résultats donnent un aperçu du bon fonctionnement de notre prototype en temps réel et renforcent l'idée de sa mobilité.

Les composants choisis pour la réalisation de notre projet sont des ressources accessibles et permettent de se familiariser avec les concepts de l'IA sans se ruiner.

#### 4.4.3 Coût matériel

- **Raspberry PI 4B**

Le modèle utilisé est une Raspberry pi 4B 4GB, son coût est estimé à 26, 000.00 DA.

- **Capteur Ultrason**

Un capteur ultrason HC-SR04, son coût est estimé à 550.00 DA.

- **Résistances**

Les deux résistances utilisées coûtent environs de 10.00 DA l'unité.

- **Câblages**

Le câblage utilisé est de type pins 4, il est estimé à 50.00 DA.

- **Sortie audio**

Des écouteurs ou casque audio peuvent être utilisés, le coût de ces accessoires varie selon les modèles.

- **Batterie externe (Power Bank)**

Le prix de cet accessoire varie selon le modèle utilisé, ils sont estimés aux environs de 4000.00 DA voir plus.

- **Microphone**

Le prix de cet accessoire varie selon les modèles proposés dans le marché, leur prix de base est de 850.00 DA et peut être plus cher pour certains modèles.

- **Total**

Matériel	Prix
Raspberry Pi 4B	26,000.00 DA
Capteur Ultrason	550.00 DA
Résistances	10.00 DA.
Câblages	50.00 DA
Power Bank	4000.00 DA
Microphone	850.00 DA
<b>Total</b>	<b>31,460.00 DA</b>

TABLE 4.4 – Tableau récapitulatif du coût matériel

Pour ce prototype, il faudra compter un total d'environ 31,500.00 DA.

Il est à noter que ce prix est extrêmement bas comparé à de nombreux modèles déjà sur le marché. Un exemple de référence de prix simple serait la paire de lunettes développée par Envision qui se trouve aux environs de 3 500 \$, ce qui est inaccessible pour la bonne majorité de la population.

#### 4.4.4 Coût Logiciel

Tous les logiciels utilisés lors du développement de ce prototype sont sous licence gratuite et open source.

## 4.5 Discussion

### 4.5.1 Avantages et limites de la méthode développée

La méthode que nous exposons dans ce mémoire est avantageuse pour de nombreuses raisons. En effet, une personne malvoyante ou aveugle peut utiliser le système en toute sécurité, ce dernier peut lui fournir de nombreuses informations sur l'environnement dans lequel elle se trouve et cela est possible grâce à l'efficacité de la détection d'objets. La Raspberry Pi 4B est capable d'effectuer une détection de manière efficace et fiable, en additionnant cette unité de calcul à un capteur ultrason, le prototype est capable de renvoyer à quelle distance se trouve un objet ou un obstacle d'une personne ayant besoin d'une assistance lui assurant ainsi une navigation plus sûre.

Nous tenons également à souligner l'accessibilité du prototype en termes de portabilité, mais également en coût, l'unité de calcul choisie est une plateforme qui favorise l'autonomie et l'indépendance, elle est également fiable et précise dans ses calculs ce qui renforce la confiance des utilisateurs du système.

Il est à noter que le prototype développé possède une optimisation suffisante et un besoin réduit en matière de performances. Cela vient justifier une facilité de déploiement sur ETA, ce qui le rend parfait pour la mise en production sur micro-ordinateurs typiquement utilisés dans les ETA.

Enfin, nous tenons à mettre l'accent sur l'évolutivité du prototype, notre projet est ouvert à la possibilité d'ajouter de nouvelles fonctionnalités et améliorations, il est adapté pour répondre

aux futurs besoins des personnes aveugles et malvoyantes et peut également s'adapter aux futures avancées technologiques.

Comme tout prototype, celui présenté ici possède des limitations fonctionnelles. La première est le nombre de FPS capables d'être produits en temps réel et cela reste le challenge principal auquel nous devons faire face dans le futur.

Un autre inconvénient serait la difficulté d'assurer une précision élevée à tout moment de l'utilisation et cela revient à des facteurs matériels et environnementaux, résolution, lumière, nombre d'objets à détecter, objets non recensés.

Le dernier inconvénient se trouve au niveau de la portabilité. En effet, le prototype développé reste assez lourd et peu pratique pour le transport.

#### 4.5.2 Discussion des futures améliorations

Étant donné que nous ne sommes qu'en phase de prototypage, il est possible d'effectuer de nombreuses améliorations.

La première étape réside au niveau du développement d'un système de guidage en temps réel avec trajectoire (Path Finding) pour les aveugles qui leur permettraient de se déplacer plus aisément sans avoir à être en alerte durant leur déplacement.

En deuxième lieu, nous envisageons d'entraîner le modèle de détection sur des jeux de données additionnels afin d'élargir le nombre de classes disponibles et être capables de donner une meilleure description de la scène.

Une autre amélioration envisagée est l'amélioration du système de question réponse (Q et A). Cela se fera en entraînant le modèle sur un jeu de données approprié pour les aveugles avec une liste beaucoup plus extensive des objets ainsi qu'une description de chacun de ces derniers.

Une amélioration importante avant production sera bien-sûr de diminuer la taille du prototype avec une version plus petite et plus légère pouvant être portée par toute personne souhaitant utiliser notre produit.

## Conclusion

Dans ce chapitre, nous avons abordé la partie software du projet, en introduisant le dataset utilisé, les logiciels ainsi que le langage de programmation choisi, nous avons également discuté de ce qui nous a motivé à nous orienter vers ces choix, nous avons également abordé les prochaines étapes qui contribueront à l'amélioration du projet et enfin, nous avons parlé des avantages et des limites de notre méthode.

# Conclusion générale et perspectives

La création d'un système intelligent embarqué visant à aider les personnes malvoyantes à naviguer en toute sécurité représente une avancée significative dans le domaine de l'assistance technologique. Ce projet met en évidence le potentiel de l'intelligence artificielle et des technologies embarquées pour améliorer la qualité de vie des personnes atteintes de déficiences visuelles.

Grâce à l'utilisation du capteur ultrason, des techniques de vision par ordinateur et d'algorithmes d'apprentissage automatique, ce système intelligent est capable de détecter et d'interpréter l'environnement grâce à la description de scène intégrée pour fournir des informations précieuses aux utilisateurs malvoyants. Il peut identifier les obstacles, les dangers potentiels et renvoyer un avertissement à l'utilisateur du système, le tout en utilisant une Raspberry Pi 4B comme unité de calcul.

Ce système intelligent embarqué offre une assistance en temps réel, combinant détection d'obstacles, extraction et lecture de texte, description de scène et alertes de navigation. Cela permet aux utilisateurs malvoyants de prendre des décisions éclairées lors de leurs déplacements. Grâce à sa portabilité, nous aspirons à apporter plusieurs améliorations en commençant par son intégration dans des dispositifs tels que les smartphones ou les lunettes intelligentes, afin d'offrir une solution pratique et accessible pour les utilisateurs.

Une des améliorations prévues, consiste également à entraîner le modèle sur des datasets plus larges et plus élaborés au domaine de l'aide à la navigation afin de détecter plus d'objets. Nous aimerions également inclure le 'Question answering' afin que l'utilisateur puisse poser diverses questions au système pour avoir des informations sur le monde qui l'entour.

Il est important de souligner que ce système ne remplace pas l'importance de l'orientation et de la mobilité traditionnelles, ni le soutien humain dans les situations complexes. Cependant, il offre une meilleure inclusion sociale aux utilisateurs.

En conclusion, la création d'un système intelligent embarqué pour aider les personnes malvoyantes à naviguer en toute sécurité représente une avancée prometteuse. Il est donc essentiel de continuer à développer et à améliorer de telles technologies pour répondre aux besoins spécifiques des utilisateurs et leur permettre ainsi une meilleure inclusion sociale au sein de notre société.

# Bibliographie

- [1] *MobileNet, MobileNet V2 and MobileNet V3.* – Keras , Disponible sur : <https://keras.io/api/applications/mobilenet/#:~:text=MobileNetV2>, Consulté le : 24/06/2023
- [2] ALAM; MAMTAZ : *Capteur infrarouge.* PNG, 2022. – Disponible sur : <https://how2electronics.com/iot-bidirectional-visitor-counter-using-espp8266-mqtt/>, Consulté le 22/05/2023
- [3] BAJAJ, Aayush : *Performance metrics in Machine learning [complete guide].* 2023. – Neptune Ai, Disponible sur : <https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide> Consulté le : 22/06/2023
- [4] BHOWMICK; PRAKASH, S.; BHAGAT, R.; PRASAD, V.; HAZARIKA, S. M. : Intellinavi : navigation for blind based on kinect and machine learning. In : *International Workshop on Multi-disciplinary Trends in Artificial Intelligence* Springer, 2014, S. 172–183
- [5] BOURBAKIS, N. G. ; KAVRAKI, D. : An intelligent assistant for navigation of visually impaired people. In : *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)* IEEE, 2001, S. 230–235
- [6] CARAIMAN, S. ; MORAR, A. ; OWCZAREK, M. ; BURLACU, A. ; RZESZOTARSKI, D. ; BOTEZATU, N. ; HERGHELEGIU, P. ; MOLDOVEANU, F. ; STRUMILLO, P. ; MOLDOVEANU, A. : Computer vision for the visually impaired : The sound of vision system. In : *Proceedings-2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018, S. 1480–1489
- [7] CASTILLO, C. ; M. ; HUARANGA-JUNCO, Edgar ; MONDRAGÓN-RUIZ, Giovanni ; SALAZAR, Andree ; BARBOSA, Luis O. ; ANTÚNEZ, Enrique A. : Ray : Smart indoor/outdoor routes for the blind using Bluetooth 4.0 BLE. In : *Procedia Computer Science* 83 (2016), S. 690–694
- [8] CLOUD, Google : *Qu'es-ce que l'intelligence artificielle ?.* – Google Cloud, Disponible sur : <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=fr> Consulté le : 23/06/2023
- [9] COCODATASET : *Coco dataset Exemple.* JPG, 2023. – Disponible sur : <https://cocodataset.org/#home>, Consulté le 23/05/2023
- [10] FAROQ, Muhammad S. ; KHAN, Harris ; ACHRAF, Imran ; SHAFI, Imran : *Smart Stick.* JPG, 2022. – Disponible sur : [https://www.researchgate.net/figure/Ultrasonic-sensors-placement-on-the-smart-stick\\_fig5\\_365580953](https://www.researchgate.net/figure/Ultrasonic-sensors-placement-on-the-smart-stick_fig5_365580953), Consulté le 23/05/2023
- [11] GANZ, A. ; SCHAFER, J. M. ; TAO, Y. ; WILSON, C. ; ROBERTSON, M. : Perceptii : smartphone based indoor navigation system for the blind. In : *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, S. 3662–3665
- [12] GUILLERM, Mikaël : *Raspberry Pi.* PNG, 2019. – Disponible sur : <https://www.tech2tech.fr/le-raspberry-pi-4-est-arrive/>, Consulté le 22/05/2023
- [13] JAFRI, R. ; CAMPOS, R. L. ; ALI, S. A. ; ARABNIA, H. R. : Visual and infrared sensor data-based obstacle detection for the visually impaired using the google project tango tablet development kit and the unity engine. In : *IEEE. Access.* 6 (2017), S. 443–454

- [14] JOSEPH, L. ; XIAO, J. ; ZHANG, X. ; CHAWDA, B. ; NARANG, K. ; RAJPUT, N.-t. ; MEHTA, S. ; VENKATA SUBRAMANIAM, L. : Being aware of the world : Toward using social media to support the blind with navigation. In : *IEEE Trans. Human-Mach. Syst.* 45 (2015), Nr. 3, S. 399–405
- [15] KALLARA, S. B. ; RAJ, M. ; RAJU, R. ; MATHEW, N.-h. J. ; PADMAPRABHA, V. R. ; DIVYA, D. S. : Indriya—a smart guidance system for the visually impaired. In : *2017 International Conference on Inventive computing and Informatics (ICICI)* IEEE, 2017, S. 26–29
- [16] KARLSSON, N. ; DI BERNARDO, E. ; OSTROWSKI, J. ; GONCALVES, L. ; PIRJANIAN, P. ; MUNICH, M. E. : The vslam algorithm for robust localization and mapping. In : *Proceedings of the 2005 IEEE international conference on robotics and automation* IEEE, 2005, S. 24–29
- [17] K.CHACCOUR ; G.BADR : Novel indoor navigation system for visually impaired and blind people. In : *2015 International Conference on Applied Research in Computer Science and Engineering (ICAR)* IEEE, 2015, S. 1–5
- [18] KHENKAR, S. ; ALSULAIMAN, H. ; ISMAIL, S. ; FAIRAQ, A. ; JARRAYA, S. K. ; BEN-ABDALLAH, H. : ENVISION : Assisted navigation of visually impaired smartphone users. In : *Procedia. Comput. Sci.* 100 (2016), S. 128–135
- [19] KUMAR, Nitin ; JAIN, Anuj : A Deep Learning Based Model to assist Blind People in their Navigation. In : *Journal of Information Technology Education : Innovation in Practice* 21 (2022)
- [20] LAST MINUTE ENGINEERS : *GPS Module*. PNG, 2023. – Disponible sur : <https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/>, Consulté le 23/05/2023
- [21] LI, B. ; MUNOZ, J. P. ; RONG, X. ; CHEN, Q. ; XIAO, J. ; TIAN, Y. ; ARDITI, A. ; YOUSUF, M. : Vision-Based mobile indoor assistive navigation Aid for blind people. In : *IEEE Trans. Mob. Comput.* 18 (2019), Nr. 3, S. 702–714
- [22] MARDER-EPPSTEIN, E. : Project tango. In : *ACM SIGGRAPH 2016 Real-Time Live!, SIG-GRAPH '16* Association for Computing Machinery, 2016, S. 25
- [23] MARZEC, P. ; KOS, A. : Low energy precise navigation system for the blind with infrared sensors. In : *2019 MIXDES-26th International Conference " Mixed Design of Integrated Circuits and Systems"* IEEE, 2019, S. 394–397
- [24] MONTANTES, J. : *Examining the transformer architecture*. 2019. – Medium, Disponible sur : <https://towardsdatascience.com/examining-the-transformer-architecture-part-1-the-openai-gpt-2-controversy-feceda43> Consulté le : 24/06/2023
- [25] MUKHIDDINOV, Mukhriddin ; CHO, Jinsoo : Smart Glass System Using Deep Learning for the Blind and Visually Impaired. In : *Electronics* (2021). <http://dx.doi.org/10.3390>. – DOI 10.3390
- [26] NAIR, Vishnu ; TSANGOURI, Christina ; XIAO, Bowen ; OLMSCHENK, G ; SEIPLE, WH ; ZHU, Z : A hybrid indoor positioning system for blind and visually impaired using Bluetooth and Google tango. In : *Journal on technology and persons with disabilities* 6 (2018)
- [27] PATHAK, A. : *Introduction aux réseaux de neurones*. 2023. – Geekflare, Disponible sur : <https://geekflare.com/fr/neural-networks/> Consulté le : 23/06/2023
- [28] PATIL, K. ; JAWADWALA, Q. ; SHU, F.-l. C. : Design and construction of electronic Aid for visually impaired people. In : *IEEE Trans. Human-Mach. Syst* 48 (2018), Nr. 2, S. 172–182
- [29] PAWAR, Sujay ; PAWAR, Bhavana ; JADHAV, Amruta ; THROAT, Ajay ; ZINJURKE, Rushikesh : Voice Based Navigation System for Blind People Using Raspberry. In : *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* 12 (2021), Nr. 1

- [30] REQUEA : *Qu'est ce qu'un système embarqué dans l'IoT ?*. – REQUEA, Disponible sur : <https://www.requea.com/systeme-embarque-iot.html> Consulté le : 22/06/2023
- [31] SATO, D. ; OH, U. ; NAITO, K. ; TAKAGI, H.-r. ; KITANI, K. ; ASAKAWA, C. : Navcog3. In : *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17*, 2017, S. 270–279
- [32] SEN, A. ; SEN, K. ; DAS, J. : Ultrasonic blind stick for completely blind people to avoid any kind of obstacles. In : *2018 IEEE SENSORS* IEEE, 2018, S. 1–4
- [33] SHAH, D. : *COCO Dataset all you need to know to get started*. 2023. – V7, Disponible sur : <https://www.v7labs.com/blog/coco-dataset-guide#:~:text=V720Dataset20Management-,What20is20the20COCO20dataset3F,520captions20describing20the20scene.>, Consulté le : 23/06/2023
- [34] STAFF : *Envision Smart Glasses*. JPG, 2022. – Disponible sur : <https://www.visionmonday.com/launchpad/systems-equipment/article/envision-releases-next-gen-ai-powered-smart-glasses-for-the-blind-and-visually-impaired> Consulté le 23/05/2023
- [35] THEBAULT : *Capteur ultrason*. JPG, 2020. – Disponible sur : <http://gilles.thebault.free.fr/spip.php?article62>, Consulté le 22/05/2023
- [36] THEBAULT : *Ultrasonic sensor*. JPG, 2020. – Disponible sur : <http://gilles.thebault.free.fr/spip.php?article62>, Consulté le 22/05/2023
- [37] TON, C. ; OMAR, A. ; SZEDENKO, V. ; TRAN, V. H. ; AFTAB, A. ; PERLA, F. ; BERNSTEIN, M. J. ; YANG, Y. : Lidar assist spatial sensing for the visually impaired and performance analysis. In : *IEEE Trans. Neural Syst. Rehabil. Eng.* 26 (2018), Nr. 9, S. 1727–1734
- [38] T.SCHWARZE ; LAUER, M. ; SCHWAAB, M. ; ROMANOVAS, M. ; BÖHM, S. ; JÜRGENSOHN, T. : A camera-based mobility aid for visually impaired people. In : *KI-Künstliche Intelligenz* 30 (2016), Nr. 1, S. 29–36
- [39] VERA, D. ; MARCILLO, D. ; PEREIRA, A. : Blind guide : Anytime, anywhere solution for guiding blind people. In : *World Conference on Information Systems and Technologies* Springer, 2017, S. 353–363
- [40] V.MEHTA : *Object detection using MobileNet V2*. 2021. – Medium, Disponible sur : <https://vidishmehta204.medium.com/object-detection-using-ssd-mobilenet-v2-7ff3543d738d>, Consulté le : 23/06/2023
- [41] (WHO), World Health O. : *Blindness And Visual Impairment*. WHO Fact Sheet, 2022. – Tiré de <https://www.who.int/fr/news-room/fact-sheets/detail/blindness-and-visual-impairment>