

Ministère de la Communication et de
l'Économie Numérique

École Supérieure Africaine des Technologies
de l'Information et de la Communication

Année académique : 2022 – 2023



MÉMOIRE DU PROJET INTERNE MASTER 2

ANALYSE ET CONCEPTION D'UN MODULE DE PERCEPTION SECURISE POUR LA NAVIGATION AUTONOME D'UN ROBOT MOBILE : CAS DE ZBOT

Présenté par :

KOUASSI Assoh Sonia Dorcas, Master 2 Sécurité Informatique et Technologies du Web

ZADI Jonathan, Master 2 Système Informatique et Génie Logiciel

DABONE Souleymane, Master 2 Réseaux Télécommunications

Période : Du 01 octobre 2022 au 31 janvier 2023

Encadrant académique

Dr PANDRY Ghislain,
Maître-assistant, ESATIC

Membres du jury

Dr SORO Adama, Maitre de Conférence, l'université de Cocody (Président du jury)
Dr BROU Pacôme, Maître-assistant, ESATIC (Rapporteur)
Dr. DOHO Libaud R.A., Maître-assistant, ESATIC

Ministère de la Communication et de
l'Économie Numérique

École Supérieure Africaine des Technologies
de l'Information et de la Communication

Année académique : 2022 – 2023



MÉMOIRE DU PROJET INTERNE MASTER 2

ANALYSE ET CONCEPTION D'UN MODULE DE PERCEPTION SECURISE POUR LA NAVIGATION AUTONOME D'UN ROBOT MOBILE : CAS DE ZBOT

Présenté par :

KOUASSI Assoh Sonia Dorcas, Master 2 Sécurité Informatique et Technologies du Web

ZADI Jonathan, Master 2 Système Informatique et Génie Logiciel

DABONE Souleymane, Master 2 Réseaux Télécommunications

Période : Du 01 octobre 2022 au 31 janvier 2023

Encadrant académique

Dr PANDRY Ghislain,
Maître-assistant, ESATIC

Membres du jury

Dr SORO Adama, Maitre de Conférence, l'université de Cocody (Président du jury)
Dr BROU Pacôme, Maître-assistant, ESATIC (Rapporteur)
Dr. DOHO Libaud R.A., Maître-assistant, ESATIC

DÉDICACE

Nous dédions ce travail à l'ensemble de nos familles.

RÉMERCIEMENTS

Nous adressons nos remerciements à toutes les personnes qui ont directement ou indirectement apporté leurs contributions à la réalisation de ce travail. Notre profonde gratitude à l'endroit de :

- Prof KONATE Adama, Directeur Général de l'ESATIC, pour la mise en place et la maintenance d'un cadre de formation de qualité et épanouissant.
- Dr PANDRY Ghislain, Chercheur à l'ESATIC, pour sa supervision rigoureuse et le temps accordé pour la réalisation de ce travail.

Tout le personnel de l'ESATIC, du corps enseignant au corps administratif, pour l'encadrement, les conseils et les coachings de développement personnel donnés durant ces trois dernières années.

Nous remercions également notre famille et nos amis pour leurs soutiens, leurs encouragements, leurs conseils et surtout leurs confiances placées en nous.

Pour terminer, nous remercions tous les étudiants de l'ESATIC, particulièrement nos aînés pour leur accompagnement et leurs conseils, pendant ces trois dernières années.

AVANT-PROPOS

Ce document reflète de façon écrite notre travail effectué durant trois (03) mois dans le cadre des projets internes de l'ESATIC qui est une école d'enseignement supérieur spécialisé dans les Technologies de l'Information et de la communication (TIC). Le but de notre travail est d'améliorer les capacités d'un robot mobile dénommé "ZBOT" dont la fonction principale est d'assister les personnes à mobilité limitée. Notre solution d'amélioration a été de concevoir un module de perception pour permettre la navigation autonome et sécurisée du robot. Cependant, la réalisation de ce projet a été parsemé de difficultés dont, on peut citer entre autres :

- l'inaccessibilité du robot ZBOT, du fait de son mauvais état, qui nous constraint à la construction d'un nouveau robot ;
- le manque de temps dû à la continuité cours pendant la période du projet ;
- la non-disponibilité des équipements en local.

En outre, ce document est un mémoire qui permet de mettre en exergue les capacités de réflexion ou les aptitudes d'analyse d'un étudiant devant une problématique spécifique.

SOMMAIRE

INTRODUCTION

PARTIE I : GÉNÉRALITÉS

Chapitre I : État de l'art

Chapitre II : Étude de l'existant

Chapitre III : Cahier de charges

DEUXIÈME PARTIE : ÉTUDE TECHNIQUE

Chapitre IV : Méthodes de Perception

Chapitre V : Analyse conceptuelle

TROISIÈME PARTIE : RÉSULTATS, DISCUSSIONS ET ESTIMATION FINANCIÈRE

Chapitre VI : Environnement de développement

Chapitre VII : Présentation des résultats

Chapitre IX : Discussions et estimation financière

CONCLUSION

LISTE DES SIGLES ET ABRÉVIATIONS

AA	Apprentissage Automatique
AP	Average Precision
CID	Cellule d’Innovation et de Développement
CNN	Convolutional Neural Network
HTML	Hyper Text Markup Language
IA	Intelligence artificielle
LAAS	Laboratoire d’Analyse et d’Architecture des Système
OpenCV	Open Source Computer Vision Library
PID	Proportionnel Intégrale Dérivé
ROS	Robot Operating System
SLAM	Simultaneous Localisation And Mapping
SysML	System Modeling Language
VO	Vision par Ordinateur
YOLO	You only look once

GLOSSAIRE

Authentification

L'authentification est un processus permettant à un système informatique de s'assurer de la légitimité de la demande d'accès faite par une entité afin d'autoriser son accès à des ressources du système conformément au paramétrage du contrôle d'accès.

Apprentissage Automatique

L'Apprentissage Automatique (AA), également appelé apprentissage machine ou apprentissage artificiel et en anglais Machine Learning (ML), est une forme d'IA qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite.

Couches Convolutives

Les couches convolutives sont des couches de réseaux de neurones profonds qui sont couramment utilisés pour l'analyse de données telles que des images, des vidéos ou des signaux audios. Ces couches permettent de capturer des motifs locaux dans les données d'entrée en appliquant une opération de convolution.

Intelligence Artificielle

L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains.

Réseaux de neurones profonds

Les réseaux de neurones profonds (ou deep learning en anglais) sont un groupe de modèles d'apprentissage automatique qui se base sur des architectures de réseau neuronal avec plusieurs couches cachées pour apprendre au travers de données complexes.

WebRTC

WebRTC est une nouvelle technologie ouverte de communication en temps réel pour navigateurs web.

LISTE DES FIGURES

Figure 1 : Robot ZBOT	8
Figure 2 : Diagramme de GANT	12
Figure 3 : Diagramme de cas d'utilisation	20
Figure 4 : Diagramme des exigences	21
Figure 5 : Architecture global du module perception la navigation.....	25
Figure 6 : Jeux de données	27
Figure 7 : Grille de détection d'objets.....	29
Figure 8 : Architecture de Yolo pour la détection des objets [12]	29
Figure 9 : Boxes de détection d'un objet	30
Figure 10 : Équation du filtre de Kalman [9].....	32
Figure 11 : Caméra de profondeur	33
Figure 12 : Caméra Stéréo.....	33
Figure 13 : Triangulation	34
Figure 14: Matrice de confusion	37
Figure 15 : Évaluation de Performance.....	38
Figure 16 : Courbe précision- rappel	39
Figure 17 : Courbe précision-seuil de confiance.....	40
Figure 18 : Courbe rappel-seuil de confiance	40
Figure 19 : Courbe spécificité-seuil de confiance.....	41
Figure 20 : Détection d'une chaise à 35%	51
Figure 21 : Détection d'une souris à 71%	51
Figure 22 : Détection d'une paire de basket à 78%	51
Figure 23 : Détection d'un téléphone de à 45%	51
Figure 24 : Suivi de la bouteille de base	52
Figure 25 : Suivi de la bouteille mouvement vers la droite	52
Figure 26 : Suivi de la bouteille mouvement vers l'arrière	53
Figure 27 : Nouveau robot	54
Figure 28 : Architecture électronique du robot.....	55
Figure 29 : Logiciel Robot	56
Figure 30 : Interface de commande.....	57

Figure 31 : Réseaux de neurones profonds	xiv
Figure 32 : Étiquetage des images avec Roboflow	xiv
Figure 33 : Augmentation et répartition des objets avec roboflow	xv
Figure 34 : Vue latérale du robot	xvi
Figure 35 : Vue Avant du Robot	xvi
Figure 36 : Alerte modèle	xxi
Figure 37 : Alerte Serializer	xxi
Figure 38 : Login.....	xxii

LISTE DES TABLEAUX

Tableau 1 : Étude comparative des méthodes	16
Tableau 2 : Matrice de confusion.....	36
Tableau 3 : Caractéristiques des machines.....	45
Tableau 4 : Liste complète des éléments de la structure du robot	46
Tableau 5 : Liste des composants électronique et mécanique.....	47
Tableau 6 : Système d'exploitation.....	48
Tableau 7 : Langages de programmation	49
Tableau 8 : Estimation financière	60
Tableau 9 : Liste des composants du robot	xvii
Tableau 10 : Tableau des machines et outils utilisés pour la réalisation	xx

INTRODUCTION

Les robots mobiles deviennent de plus en plus importants pour l'homme dans l'accomplissement des tâches difficiles à réaliser par ce dernier. Plusieurs types de modèles ont été développés, notamment les robots d'assistance aux personnes. Ces robots doivent être capables de naviguer, quel que soit l'environnement dans lequel ils se trouvent, au moyen de la perception afin d'être plus utiles à l'homme. En robotique, la perception est comprise comme un système qui confère au robot la capacité de percevoir, de comprendre et de raisonner sur l'environnement qui l'entoure. Elle est liée à plusieurs applications de la robotique où les données sensorielles et les techniques d'intelligence artificielle ainsi que l'apprentissage automatique (IA/ML) sont indispensables. Ce module de perception est crucial pour qu'un robot puisse comprendre l'environnement afin de prendre des décisions, planifier et opérer dans cet environnement. Par ailleurs, Ce module doit également inclure des mécanismes de sécurité pour protéger les données de perception, détecter les menaces potentielles, réagir en temps réel et s'adapter aux changements dans l'environnement. Les attaques informatiques doivent être aussi prises en compte et protéger le système contre les actions malveillantes. C'est dans cette optique que, passionnées des domaines de la robotique, l'IA, l'apprentissage automatique et à la sécurité informatique, nous portons un grand intérêt pour notre projet interne dont le thème est : « Analyse et Conception d'un module de perception sécurisé pour la navigation autonome d'un robot mobile : cas de Zbot ». Alors, il est important de se poser un certain nombre de questions dont les réponses nous permettront d'apporter une solution au problème :

- Comment concevoir un module de perception sécurisé pour la navigation autonome d'un robot mobile ?
- Quelles sont les techniques et technologies adaptées pour la conception d'un tel module ?
- Quels mécanismes de sécurité faut-il intégrer pour assurer la sécurité de robot ?

Afin de mener à bien ce projet, il sera structuré en trois grandes parties : d'abord, une étude théorique qui nous permettra d'avoir une connaissance transversale sur les travaux similaires déjà réalisés et l'étude du projet. Ensuite, une étude technique dans laquelle nous retrouverons les différentes techniques et technologies pouvant être exploité pour la réalisation du projet avant de passer à la mise en œuvre dudit projet. Enfin, nous terminerons par la présentation des résultats obtenus et analyserons ces résultats comparativement à l'étude théorique et aux résultats attendu.

PRÉMIÈRE PARTIE : GÉNÉRALITÉS

CHAPITRE I : ETAT DE L'ART

Ce chapitre faire l'objet des notions de base et d'une étude sur les travaux antérieurs réalisé par rapport à notre thème

I. DEFINITIONS

1. Perception d'un robot mobile

La perception d'un robot mobile consiste à collecter et analyser des données sensorielles pour créer une représentation de l'environnement du robot. Cela inclut la détection d'objets, la reconnaissance de forme, la mesure de la distance et de la position et la détection de mouvement.

2. La navigation sémantique

La navigation sémantique d'un robot mobile se concentre sur la compréhension de l'environnement dans lequel le robot se déplace. Cela implique la capacité du robot à reconnaître les objets, les personnes et les lieux dans son environnement, ainsi qu'à comprendre les relations spatiales entre ces éléments. La navigation utilise des données de perception pour créer une carte de l'environnement et planifier des itinéraires pour atteindre des objectifs spécifiques.

3. Robot mobile

Un robot mobile est un type de robot capable de se déplacer dans un environnement physique en utilisant des méthodes telles que la locomotion à roues, à pattes, à chenilles, à hélices, à réaction ou à tout autre mécanisme de mouvement. Les robots mobiles peuvent être contrôlés manuellement ou programmés pour fonctionner de manière autonome en utilisant des capteurs pour percevoir leur environnement et des algorithmes pour prendre des décisions en fonction de ces perceptions. Les robots mobiles sont utilisés dans de nombreuses applications, telles que l'exploration spatiale, la recherche et sauvetage, la surveillance, la logistique, la fabrication et le divertissement.

II. QUELQUES ROBOT DOTÉ DE LA NAVIGATION AUTONOME ET CRITIQUE

1. Les robots LAAS-CNRS

Pour naviguer de façon autonome, les robots du Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS) intègrent plusieurs fonctions sensorimotrices réalisant les actions de planification de trajectoire, localisation et contrôle de mouvement. Ainsi, il contrôle ses mouvements grâce à différentes stratégies ou méthodes durant une tâche de navigation : méthode réactive par potentiels attractif et répulsif, méthode de déformation de bande élastique le long de la trajectoire pré planifiée. La localisation quant à elle s'appuie sur deux capteurs : les capteurs proprioceptifs qui mesurent des quantités reliées à des dérivées premières ou secondes de la position et les capteurs extéroceptifs qui mesurent l'état d'une relation entre l'entité en mouvement et l'environnement extérieur [10].

2. Les robots mobiles de Meanwhile

Les robots mobiles Meanwhile sont dotés d'une intelligence artificielle spécialisée dans la navigation en intérieur (SLAM). Le SLAM, Simultaneous Localisation And Mapping, permet au robot de construire son environnement et de modifier son comportement en fonction des obstacles non cartographiés tout en se localisant en temps réel. Afin de se déplacer en toute autonomie et sans trajectoire pré définie, le robot va combiner les informations qui lui sont propres avec les informations de son environnement (renvoyées par ses lasers et capteurs). En cas d'obstacle (opérateurs, palettes, chariots élévateurs, etc.) obstruant son passage, le robot mobile le contourne en toute sécurité, en déterminant une trajectoire alternative. La productivité s'en trouve optimisée, car les délais d'approvisionnement sont plus facilement respectés [13].

3. Le robot de surveillance de Knightscope

Le robot de surveillance mobile de Knightscope est un robot de sécurité autonome conçu pour patrouiller les zones de sécurité telles que les campus, les centres commerciaux, les immeubles de bureaux et les hôpitaux conçus par la société Knightscope, Inc., une entreprise américaine spécialisée dans la fabrication de robots de sécurité autonomes. Il utilise une variété de capteurs pour surveiller son environnement, y compris des caméras haute définition, des microphones, des détecteurs de gaz, des capteurs de température et des capteurs de mouvement. Les données recueillies par ces capteurs sont analysées en temps réel par des algorithmes d'apprentissage automatique pour détecter les menaces potentielles et les activités suspectes. Le robot est également équipé de capacités de reconnaissance faciale et de reconnaissance de plaques d'immatriculation, ce qui permet aux opérateurs de sécurité de suivre les individus suspects ou de rechercher des véhicules spécifiques.

4. Critique

Les robots mobiles de Meanwhile, les robots LAAS-CNRS et le robot de surveillance de Knightscope sont tous des exemples de robots mobiles avancés qui utilisent des technologies de navigation autonomes. Cependant, comme toute technologie, ces robots ont des limites qui doivent être prises en compte. Voici quelques limites pour chacun de ces types de robots :

4.1. Les robots mobiles de Meanwhile

Les robots de Meanwhile présentent deux limites principales, à savoir :

- Limites de la perception : les capteurs et caméra utilisés par ce robot peuvent ne pas être suffisamment précis pour détecter tous les obstacles ou identifier les zones de danger potentiel.

- Limites de l'autonomie : Ils nécessitent toujours une intervention humaine pour certaines tâches, telles que le changement de batterie ou le réglage des capteurs.

4.2. Les robots LAAS-CNRS

Ces robots aussi ne sont pas des exceptions quant aux limites, nous avons entre autres :

- Limites de la navigation : les algorithmes utilisés ne sont pas suffisamment robustes pour naviguer dans des environnements complexes ou changer de comportement en fonction des conditions.
- Limites de la perception : Les capteurs qu'ils utilisent peuvent être limités par les conditions de lumière ou les obstacles.

4.3. Le robot de surveillance de Knightscope

Au niveau du robot de surveillance de Knightscope nous notons comme limites :

- L'interaction humaine : ils ne sont pas toujours capables d'interagir efficacement avec les personnes dans ces zones, en particulier en cas d'urgence ou de crise.
- La surveillance : Ils ne sont souvent pas en mesure de détecter tous les types de menaces ou de suivre tous les individus suspects.

Conclusion partielle

Les robots présentés sont des exemples impressionnantes de la technologie de la navigation autonome, mais il est important de reconnaître leurs limites pour une utilisation efficace et sûre dans diverses applications. Notre contribution sera d'apporter cette fonctionnalité à un robot mobile existant tout essayant de pallier les limites relevées par rapport à ces robots mobiles avancés

CHAPITRE II : ÉTUDE DE L'EXISTANT

Ce chapitre est consacré à présenter l'existant de notre projet et en dégager les limites afin de trouver la solution appropriée.

I. PRÉSENTATION DE L'EXISTANT

Notre existant dans ce projet est un robot dénommé “ZBOT”. ZBOT est un projet qui allie santé et technologie. Il est parti du constat que les personnes souffrant d'un handicap moteur déploient la plupart du temps beaucoup d'efforts pour effectuer des tâches qui paraissent anodines pour le commun des mortels. Il représente donc une extension qui comble le handicap moteur. Ce projet a été développé au sein du département d'ingénierie de la Société Ivoirienne d'Intelligence Numérique (SIIN) en collaboration avec la Cellule d'Innovation et de Développement (CID) de l'ESATIC.



Figure 1 : Robot ZBOT

Le Zbot est doté d'une suspension rocker bogie et d'un bras mécanique pour la récupération des objets. Il dispose aussi d'une interface de commande qui peut être déployée sur un ordinateur, un smartphone ou une tablette. Il est commandé manuellement grâce à l'interface de commande.

Ce projet a permis à l'équipe de participer à de nombreuses compétitions et remporté plusieurs prix dont l'Hackathon 5G d'Orange et bien d'autres.

II. LIMITES DU ROBOT ZBOT

Bien que le ZBOT soit une réelle révolution dans le cadre de l'assistance aux personnes à capacité réduite, il demande cependant beaucoup d'effort ces personnes, et cela s'explique à plusieurs niveaux :

- La navigation et le fonctionnement du bras mécanique dépendent entièrement d'une présence humaine, en occurrence celle des personnes à capacité réduite.
- Il n'est pas doté d'une intelligence lui permettant reconnaître les objets.
- Incapacité à cartographier l'environnement dans lequel il se trouve.

Notre contribution dans le cadre du projet interne est de combler ces limites énumérées par la conception d'un module permettant au robot de s'émouvoir de façon autonome.

Conclusion partielle

L'étude de l'existant sur le robot ZBOT nous a permis de relever les limites et de mieux orienter notre solution.

CHAPITRE III : CAHIER DES CHARGES

Ce chapitre est consacré à la présentation générale de notre projet au moyen de l'analyse du contexte et du cahier des charges.

I. PRESENTATION GENERALE DU PROJET

1. Contexte du projet

Dans le souci de mieux assister les personnes à capacité réduite pour l'accomplissement des tâches dont la réalisation demande assez d'effort, un robot mobile a été mis en place, le robot “ZBOT”. Cependant, il ne peut assurer l'assistance complète et efficace qui sera plus utile pour ces personnes. C'est dans l'optique de combler ces limites que nous avons décidé de travailler sur la navigation autonome et sécurisée du robot en concevant un module de perception, dans le cadre des projets internes de l'ESATIC.

2. Objectifs du projet

2.1. Objectif général

L'objectif principal de ce projet est d'assurer la navigation autonome et sécurisée du robot ZBOT.

1.2. Objectifs spécifiques

Dans l'atteinte de notre objectif principale, nous l'avons spécifié en sous objectifs qui sont entre autres :

- Assurer une bonne capacité de visualisation de l'environnement.
- Contrôler la navigation au travers de la planification de trajectoire.
- Déetecter et reconnaître les objets.

- Calcul de distance entre la position du robot et celle de l'objet.
- Commander le robot à distance.

II. BESOINS ET CONTRAINTES DE RÉALISATION ET DIAGRAMME DE PROJET

1. Besoins et contraintes de réalisation

Les besoins fonctionnels auxquels notre robot mobile devra répondre pour assurer une navigation tout à fait autonome et sécurisée sont les suivants :

- L'apprentissage automatique pour la reconnaissance des objets.
- Une interface de commande prenant en compte l'accessibilité, la facilité d'utilisation, la personnalisation et le feedback, permettant de prendre le contrôle manuel du robot et le pilotage à distance.
- Une interaction entre l'application de commande et robot.
- Affichage des données perçue par le robot sur l'interface de commande.
- Recherche d'un objet bien défini avec précision tout évitant les obstacles.
- Une grande précision dans la cartographie de l'environnement.
- Assurer la protection contre les interférences électromagnétiques et les injections de données.

2. Calendrier de réalisation du projet

Nous nous sommes servis du diagramme de Gantt visible sur la figure 2 pour représenter l'état d'avancement de notre projet. En effet, le projet a été réalisé sur 3 mois du vingt (20) novembre au dix-huit (18) février, et a été subdivisé en plusieurs tâches avec différents responsables dans le but de travailler de façon structurée.

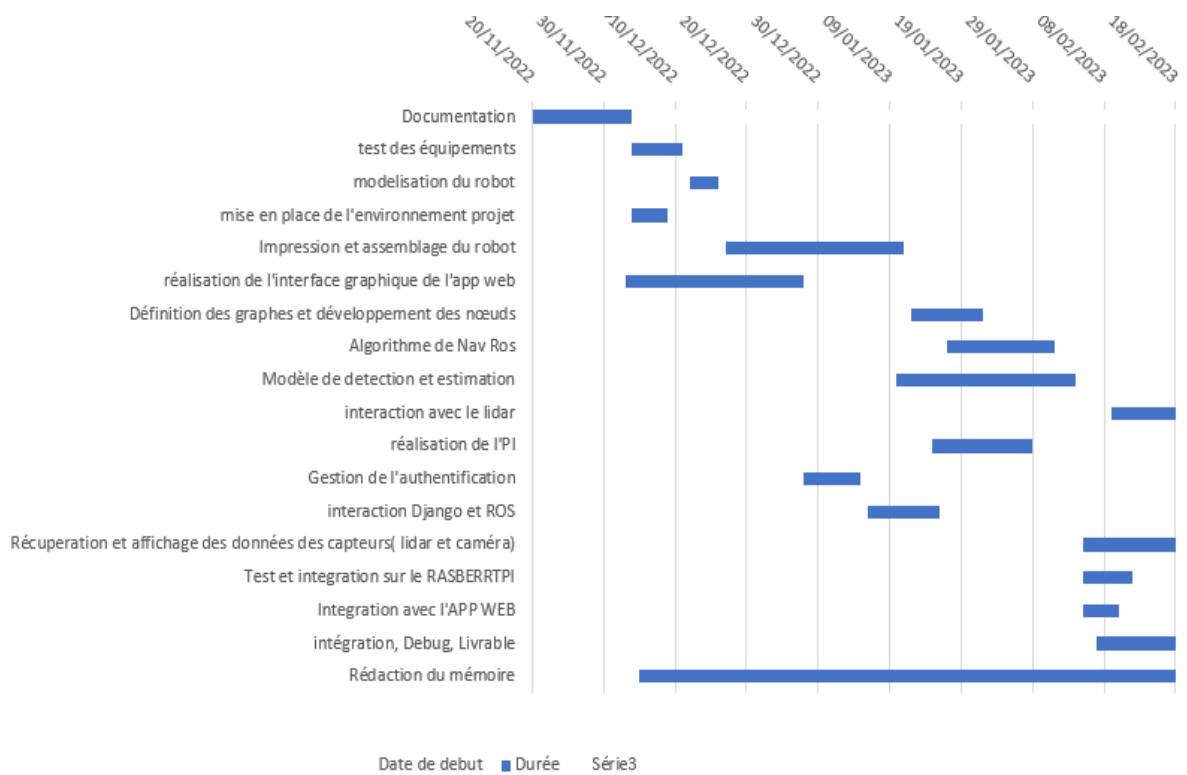


Figure 2 : Diagramme de GANT

Conclusion partielle

Le cahier de charge du projet étant bien défini, au travers des Objectifs, et des besoins fonctionnels, nous passons maintenant l'étude de l'étude conceptuelle de la solution.

DEUXIÈME PARTIE : ÉTUDE TECHNIQUE

CHAPITRE V : MÉTHODES ET OUTILS UTILISÉS

I. PRÉSENTATION DES MÉTHODES DE PERCEPTION

1. Méthode de perception basée sur l'apprentissage automatique

La perception par l'apprentissage automatique est une méthode d'analyse de données qui permet à une machine d'apprendre à interpréter les informations sensorielles, telles que des images, des sons ou des données textuelles. L'apprentissage automatique est une méthode spécifique pour développer des modèles d'IA en utilisant des algorithmes d'apprentissage à partir de données d'entraînement. Elle est largement utilisée dans des applications telles que la reconnaissance d'images, la reconnaissance vocale, la traduction automatique et la prédiction de séries chronologiques. Cette méthode se fait en plusieurs étapes :

- Collecte de données : Un ensemble de données d'entraînement doit être collecté et étiqueté pour l'apprentissage. Les données peuvent être des images, des vidéos, du texte, des signaux audios, etc.
- Prétraitement de données : Les données doivent être nettoyées, normalisées et préparées pour l'analyse
- Création d'un modèle de réseau de neurones : Le modèle est créé en spécifiant le nombre de couches de neurones, le type de fonctions d'activation et d'autres paramètres.
- Entraînement du modèle : Le modèle est entraîné à partir des données d'entraînement en ajustant les poids des connexions neuronales pour minimiser l'erreur de prédiction.
- Validation du modèle : Le modèle est validé en utilisant un ensemble de données de validation pour mesurer ses performances et ajuster les hyperparamètres si nécessaire.
- Test du modèle : Le modèle est testé avec un ensemble de données de test pour évaluer sa capacité à généraliser les connaissances apprises.
- Utilisation du modèle : Le modèle peut être utilisé pour prédire des sorties pour de nouvelles données non vues auparavant.

2. Méthode de perception basée sur les capteurs

La méthode de perception basée sur les capteurs est une approche courante pour permettre à des machines ou des robots de percevoir leur environnement en utilisant des capteurs. Cette méthode permet aux machines de collecter des informations sur leur environnement à partir de sources telles que des caméras, des microphones, des capteurs de pression, des capteurs de température, des capteurs de proximité, etc. Les données collectées par les capteurs sont ensuite traitées à l'aide d'algorithmes de traitement de signal et de techniques de traitement d'image pour extraire des informations pertinentes sur l'environnement. Les informations collectées peuvent inclure la détection d'objets, la reconnaissance de formes, la localisation, la cartographie, etc. Les avancées dans les technologies de capteurs et les algorithmes de traitement de signal ont permis une amélioration significative de la précision et de la fiabilité des systèmes de perception basés sur les capteurs.

3. Méthode de perception basée sur la Vision par ordinateur

La vision par ordinateur est un domaine de l'intelligence artificielle (IA) qui permet aux ordinateurs et aux systèmes de dériver des informations notables à partir d'images numériques, de vidéos et d'autres entrées visuelles, et de prendre des mesures ou de faire des recommandations sur la base de ces informations. La méthode de perception basée sur la vision par ordinateur est une approche pour permettre aux machines de percevoir leur environnement à l'aide de caméras et de techniques de traitement d'image. Cette méthode est couramment utilisée dans les applications de reconnaissance d'images, de navigation autonome, de surveillance et de robotique. Dans cette méthode, des images en 2D ou en 3D de l'environnement sont capturées à l'aide de caméras ou de capteurs d'image. Les images sont ensuite traitées à l'aide d'algorithmes de traitement d'image pour extraire des informations pertinentes sur l'environnement. Les informations collectées peuvent inclure la détection d'objets, la reconnaissance de formes, la localisation, la cartographie, etc. Les algorithmes de traitement d'image utilisent des techniques telles que la segmentation d'image, la détection de contours, la reconnaissance de formes et la correspondance d'image pour extraire des caractéristiques clés des images.

Ces caractéristiques sont ensuite utilisées pour identifier et localiser les objets dans l'image.

II. ETUDE COMPARATIVE DES MÉTHODES ET CHOIX DE LA MÉTHODE

1. Étude comparative

Le tableau suivant illustre l'étude comparative.

Tableau 1 : Étude comparative des méthodes

Méthodes de perception	Apprentissage automatique	Capteurs	Capteurs
Avantages	<p>Capacité d'apprentissage continu et d'adaptation aux nouvelles situations</p> <p>Capacité à détecter des modèles complexes et à reconnaître des objets inconnus</p> <p>Réduction du besoin de programmation manuelle pour la reconnaissance d'objets</p> <p>Possibilité de prendre en compte plusieurs sources de donnée</p>	<p>Réponse rapide aux changements environnementaux</p> <p>Possibilité de collecter des données en temps réel</p> <p>Faible coût par rapport aux autres méthodes</p> <p>Faible consommation d'énergie</p>	<p>Capacité à collecter des données visuelles riches</p> <p>Précision élevée dans la reconnaissance d'objets et de formes</p> <p>Faible coût par rapport à la méthode de perception basée sur l'apprentissage automatique</p>

			Possibilité de prendre en compte des données spatiales
Inconvénients	<p>Besoin d'un grand volume de données pour entraîner le modèle de reconnaissance</p> <p>Risque de biais dans les données d'entraînement, qui peut affecter la performance du modèle</p> <p>Temps de traitement plus long que les autres méthodes</p>	<p>Limitations dans la précision des capteurs</p> <p>Nécessite des connaissances approfondies en ingénierie pour la conception et la mise en place des capteurs</p> <p>Incapacité à collecter des informations visuelles ou sonores dans certains environnements</p>	<p>Limitations dans les conditions d'éclairage et les obstacles physiques pouvant bloquer les images</p> <p>Nécessite des connaissances approfondies en traitement d'image pour extraire des caractéristiques pertinentes</p> <p>Peut être sensible aux biais et aux erreurs dans les données d'entraînement</p>

2. Choix de la méthode

Au regard des avantages de chaque méthode, nous choisissons de faire une combinaison des trois (3) méthodes pour plusieurs raisons. En combinant ces trois (3) méthodes, le robot mobile peut bénéficier de la robustesse et de la flexibilité de l'apprentissage automatique, de la précision et de la fiabilité des capteurs, ainsi que de l'analyse détaillée des images fournies par la vision par ordinateur.

Conclusion partielle

En somme, la combinaison de ces méthodes de perception permet de fournir des informations complètes et précises sur l'environnement, et est donc une solution idéale pour la navigation autonome.

CHAPITRES VI : ANALYSE CONCEPTUELLE

I. MODÉLISATION DU SYSTÈME

1. Présentation de la méthode de modélisation SysML

Pour modéliser le système de navigation d'un robot mobile, SysML (System Modeling Language) est un langage de modélisation adapté. Il permet de représenter visuellement la structure, le comportement et les interactions d'un système complexe comme celui-ci.

En effet, grâce à ses diagrammes standardisés, SysML permet de modéliser de manière complète et cohérente les différents composants du système, ainsi que leurs interactions et leurs processus. Par exemple, en utilisant des diagrammes de blocs, d'exigences, d'états, de séquence et d'activité, vous pouvez décrire la plateforme, les moteurs, les capteurs, les exigences de navigation, les états de fonctionnement du robot, les interactions entre les différents composants, les étapes de navigation, etc. En outre, SysML permet également de vérifier et de valider la conception du système, en effectuant des simulations et des analyses de performance, et en détectant d'éventuelles erreurs ou incohérences dans la modélisation.

En résumé, SysML est un langage de modélisation adapté pour la conception de systèmes complexes comme le système de navigation d'un robot mobile, car il permet de représenter visuellement et de manière rigoureuse l'ensemble des composants, des interactions et des processus du système, tout en offrant des outils d'analyse et de validation performants.

2. Diagramme de cas d'utilisation

La figure ci-contre présente notre diagramme de cas d'utilisation.

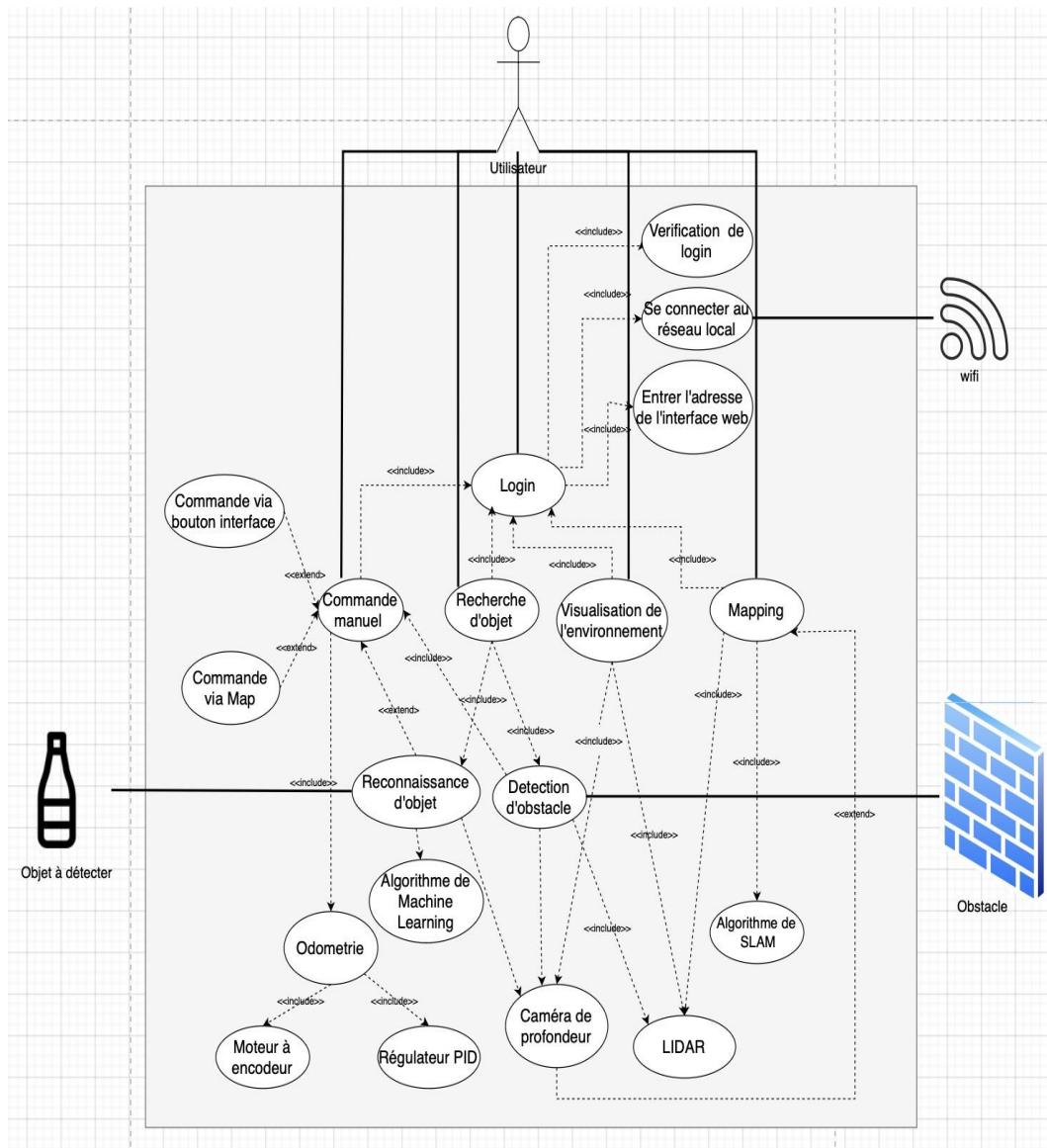


Figure 3 : Diagramme de cas d'utilisation

3. Diagramme des exigences

Ce diagramme permet de spécifier les exigences du système, c'est-à-dire les fonctionnalités et les contraintes qu'il doit respecter. Il nous permet ainsi de définir les exigences de navigation du robot, telles que la précision, la vitesse, la sécurité, etc.

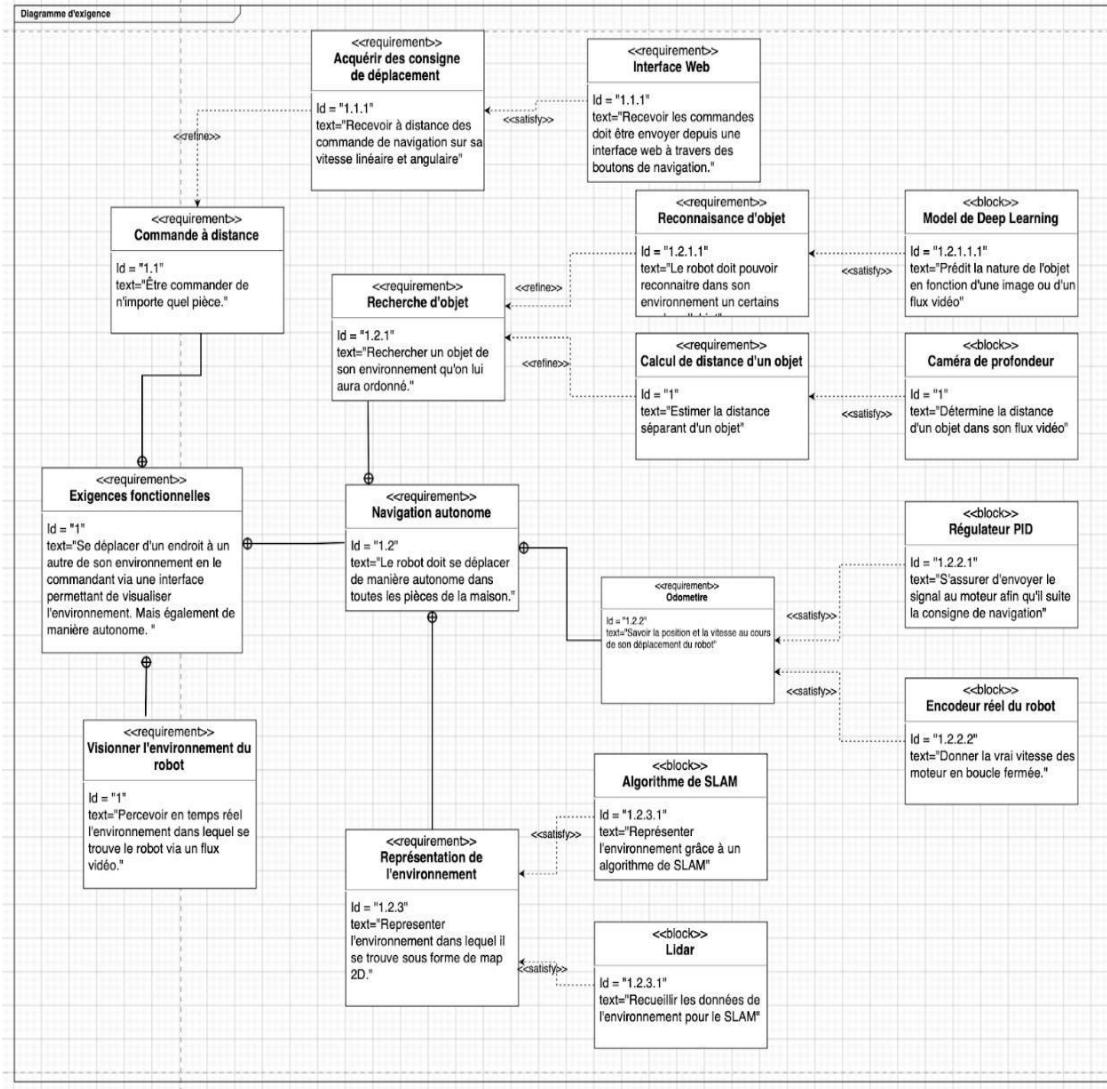


Figure 4 : Diagramme des exigences

4. Diagramme de séquence

Le diagramme de séquence est un diagramme qui permet de représenter les interactions entre les différents composants du système, en montrant les messages échangés entre eux. Vous pouvez ainsi représenter les interactions entre les capteurs, le système de contrôle, les moteurs, etc.

II. CONCEPTION DU SYSTÈME DE PERCEPTION PAR LES CAPTEURS

1. Description

La conception d'un système de perception efficace est essentielle pour la navigation autonome d'un robot mobile. Dans cette partie, nous allons décrire comment nous avons conçu le système de perception de notre robot mobile à partir des capteurs suivants : un lidar, une caméra de profondeur et des encodeurs moteurs. Nous allons également décrire comment nous avons intégré ces capteurs dans la plateforme robotique, qui est un robot à différentiel drive de 35 cm de haut et 48 cm de diamètre, équipé d'un Raspberry Pi comme ordinateur embarqué et d'un Arduino pour les moteurs et les encodeurs.

2. Spécification des besoins de perception

Avant de concevoir le système de perception, il est important de déterminer les besoins de perception. Pour ce faire, nous avons identifié les exigences de perception pour la navigation autonome de notre robot, telles que la détection d'obstacles, la localisation précise du robot, la cartographie de l'environnement et la détection de la profondeur. Nous avons également défini les contraintes et les limites du système de perception, telles que la taille, le poids, la consommation d'énergie et le coût. Enfin, nous avons déterminé les caractéristiques de performance requises pour chaque capteur, telles que la portée, la résolution spatiale et temporelle, la précision et la fiabilité.

3. Sélection des capteurs pour la perception

Pour répondre aux besoins de perception identifiés, nous avons choisi trois capteurs : un lidar, une caméra de profondeur et des encodeurs moteurs. Le lidar est un capteur laser qui mesure la distance et la position des objets environnants.

La caméra de profondeur est une caméra qui utilise une lumière structurée pour mesurer la profondeur des objets environnants. Les encodeurs moteurs sont des capteurs qui mesurent la vitesse de rotation des moteurs et la distance parcourue par le robot. Nous avons justifié ces choix en fonction des caractéristiques techniques des capteurs et de leur capacité à répondre aux exigences de perception.

4. Conception des interfaces capteurs

4.1. Conception du système de perception par le LIDAR

Le premier aspect à considérer dans la conception du système de perception est l'utilisation d'un LIDAR. Le LIDAR est un capteur de télémétrie laser qui mesure la distance entre le robot et les objets environnants. Il permettra de cartographier l'environnement du robot et de détecter les obstacles sur son chemin. Pour intégrer le LIDAR dans le système, il faudra d'abord définir sa position sur le robot. Ensuite, il faudra mettre en place un algorithme pour traiter les données collectées par le LIDAR et créer une carte de l'environnement.

4.2. Conception du système de perception par la caméra de profondeur

La caméra de profondeur est un autre élément clé du système de perception. Elle permet de mesurer la profondeur des objets et de créer une représentation en trois dimensions de l'environnement. Comme pour le LIDAR, il faudra déterminer la position de la caméra sur le robot. Ensuite, il faudra développer un algorithme pour traiter les données de profondeur collectées par la caméra. Cet algorithme permettra de détecter les obstacles et de créer une carte de l'environnement en 3D.

4.3. Conception du système de perception par les encodeurs moteur

Les encodeurs moteurs sont des capteurs qui mesurent la vitesse et la position des moteurs du robot. Ils permettent de calculer la distance parcourue par le robot et de connaître sa position exacte. Pour intégrer les encodeurs moteurs dans le système, il faudra les connecter à l'Arduino et à le Raspberry Pi, et développer un algorithme pour traiter les données collectées. Cet algorithme permettra de calculer la position et l'orientation du robot en temps réel.

5. Intégration des différents composants

Une fois que les différents systèmes de perception sont conçus et testés, il faudra les intégrer dans le robot. Le LIDAR, la caméra de profondeur, les encodeurs moteurs, l'Arduino et le Raspberry Pi doivent être connectés et communiquer. Il faudra également développer un algorithme de contrôle pour le robot, qui utilise les données de perception pour se déplacer en toute sécurité dans son environnement. Cela permet la mesure de la vitesse de rotation des moteurs et la distance parcourue par le robot. La figure suivante présente l'architecture globale du module perception la navigation.

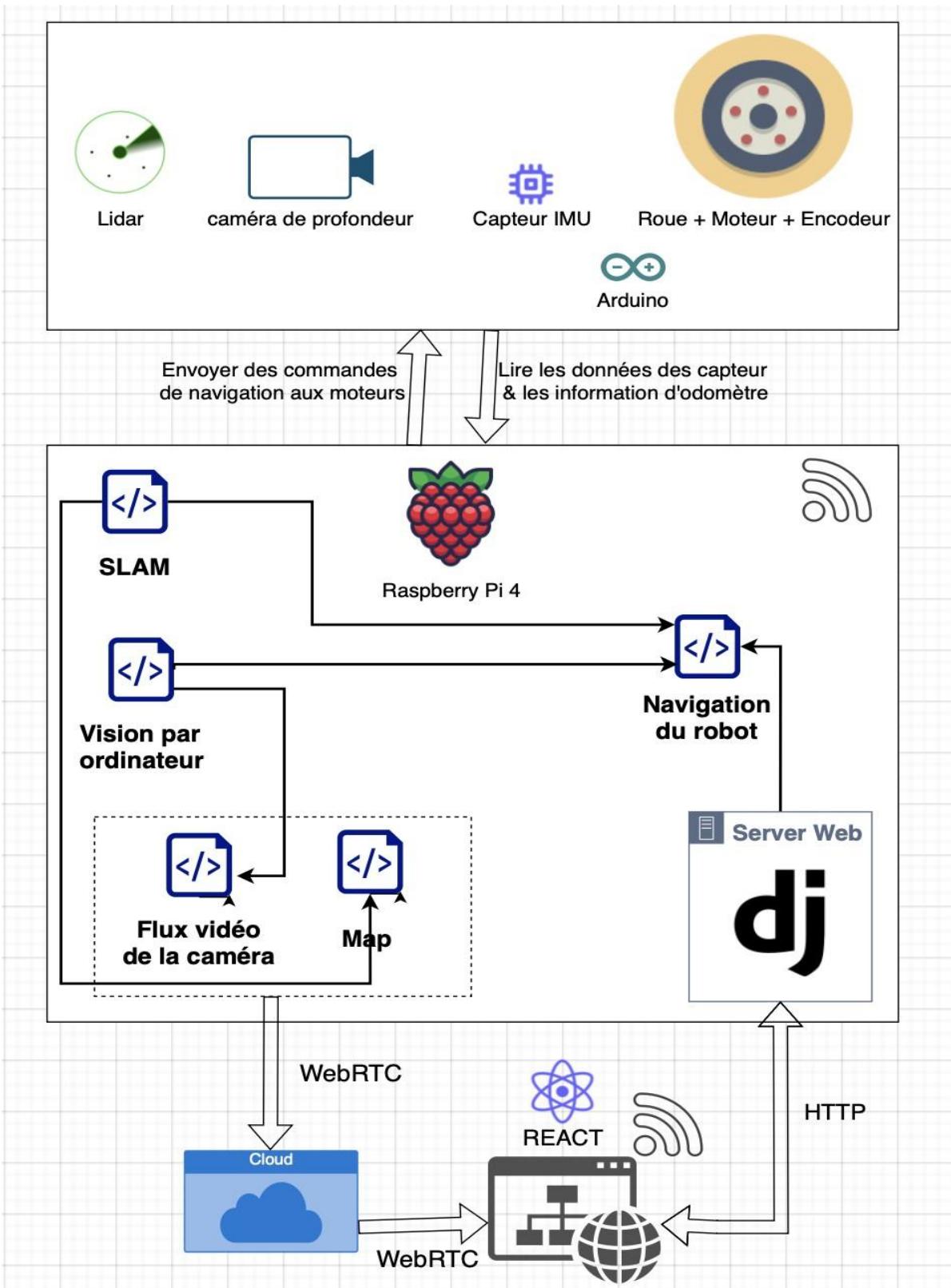


Figure 5 : Architecture global du module perception la navigation

III. CONCEPTION DU MODULE DE VISION PAR ORDINATEUR

1. Collecte des données

Pour réaliser un modèle de VO, la première étape est la collecte de données. L'efficacité et la crédibilité de notre modèle seront basées sur ces données. Pour ce projet, nous avons utilisé les données provenant de ce lien [11]. Sur cette page Web l'on retrouve un fichier zip contenant les images objets qu'on retrouve fréquemment dans une maison. Ce fichier zip contient quatre catégories d'images qui sont : des images réelles, de produits, artistiques et des cliparts. Ces quatre d'images permettront à notre modèle de s'entraîner sur une distribution variée et de mieux cerner certaines caractéristiques de différenciations des objets de la maison au sein des images.

2. Traitement des données

Les données recueillies sur ce site web ne sont pas en mesure d'être interprétées par notre modèle de VO. C'est dans le but de les rendre interprétables que l'on effectue leurs traitements. Le traitement a consisté à :

- télécharger les données de la page web ;
- supprimer les images inexploitables ;
- étiqueter les objets au sein de nos images à l'aide de l'outil roboflow, vous pouvez voir un exemple d'étiquetage via la figure 35 à l'annexe 1 ;
- augmenter nos données en appliquant des filtres (luminosité, contraste, rotation, zoom, netteté, etc) pour passer de 2 694 images à 6 510 Ces filtres permettent à nos modèles de mieux fonctionner sur tous types de flux d'images qu'il reçoit (flux d'images floue, sombre et autre) ;
- diviser le jeu de données en ensemble d'entraînement (train), de validation (val) et de test (test). Cette division permet de déterminer nos images sont utilisées pour entraîner, valider et évaluer nos algorithmes de traitement

d'images. A la suite de cette division, nous avons obtenu 5 700 pour le train soit 87%, 537 pour le val soit 8% et 273 pour le test, soit 4%. Voir la figure 36 à l'annexe 1 Pour plus de détails [4].

Une fois toutes ces étapes terminées, nous avons téléchargé nos données pour les fournir à nos algorithmes de VO.

3. Visualisation des données

La figure 6 ci-dessous nous présente l'aperçu de nos données obtenues réparties en 16 classes d'objets.

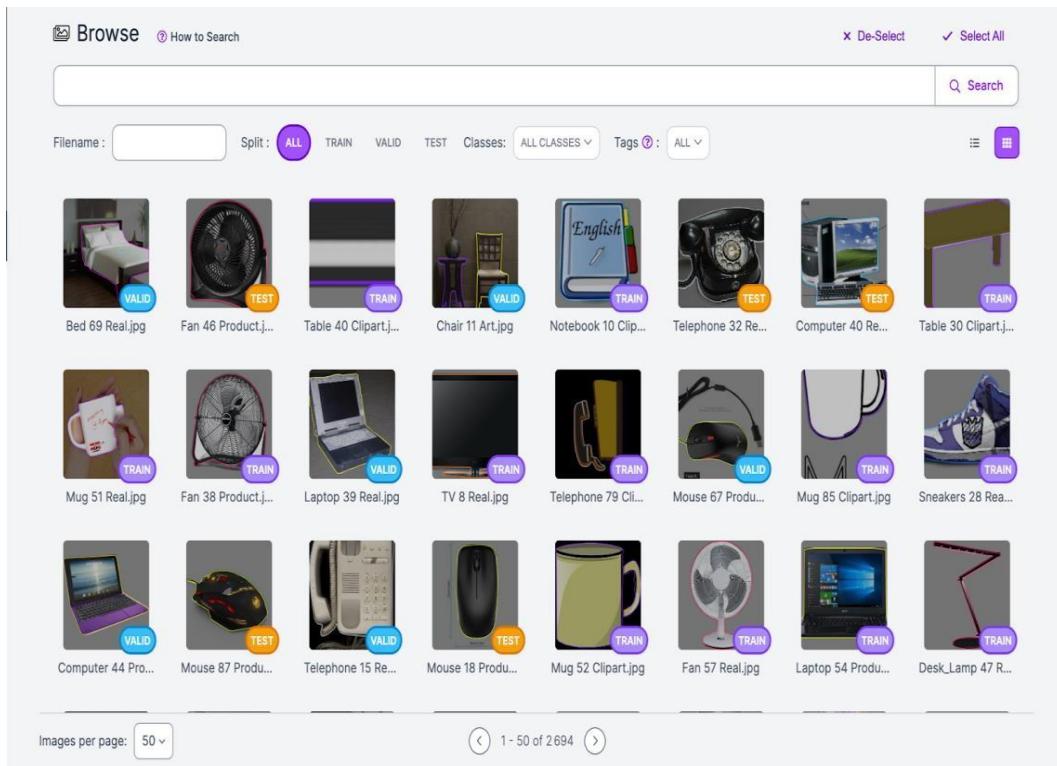


Figure 6 : Jeux de données

4. Réalisation du modèle de vision par ordinateur

Pour la réalisation de nos modèles de VO permettant à notre caméra de profondeur installée sur le rover de détecter et suivre les objets. Nous sommes parties sur l'implémentation des algorithmes de détection, de suivi d'objets se basant sur la bibliothèque YOLO et la mesure de la profondeur de la scène.

4.1. Choix de YOLO

YOLO est un algorithme de détection d'objets rapide et efficace qui a été développé pour la reconnaissance d'objets en temps réel. Il a été introduit pour la première fois en 2015 par Joseph Redmon et Ali Farhadi [7]. YOLO utilise un réseau de neurones convolutifs (CNN) pour effectuer la reconnaissance d'objets en une seule passe. Contrairement à d'autres approches de détection d'objets qui nécessitent plusieurs passes pour détecter les objets, YOLO utilise une architecture de réseau de neurones en une seule passe pour localiser et classer les objets dans une image. L'algorithme YOLO utilise également une approche de la boîte englobante pour localiser les objets dans une image. Comparativement à son voisin OpenCV (OpenCV est une bibliothèque logicielle open-source populaire de vision par ordinateur et d'apprentissage automatique utilisée pour développer des applications de vision par ordinateur en temps réel [8]), YOLO est le plus adapté pour la solution que nous essayons de mettre en place car il nous permet de :

- détecter et suivre plus rapidement nos objets en temps réel au sein de la maison ;
- compatible avec la caméra de profondeur Intel® RealSense que nous utilisons customiser notre modèle assez aisément sur nos propres données.

4.2. Algorithmes de détection d'objet

Le fonctionnement de l'algorithme de détection d'objets YOLO se déroule généralement en plusieurs étapes :

- Prétraitement de l'image : L'image d'entrée est redimensionnée pour s'adapter à la taille d'entrée du modèle YOLO.
- Diviser l'image en grille : L'image est divisée en une grille de $n \times n$ cellules, où chaque cellule est responsable de détecter un objet potentiel.

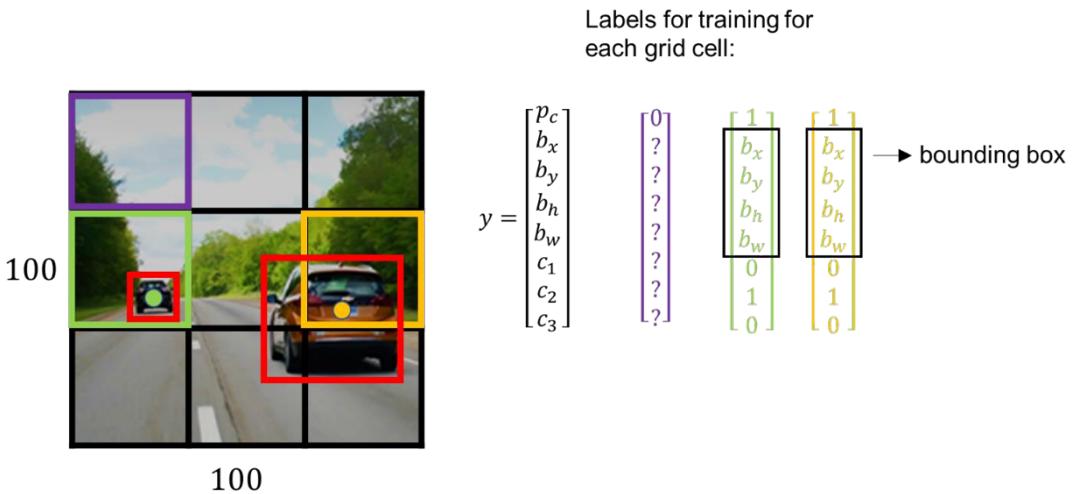


Figure 7 : Grille de détection d'objets

- Prédiction de la probabilité et des boxes : Pour chaque cellule, YOLO utilise un réseau neuronal pour prédire la probabilité d'existence d'un objet dans cette cellule, ainsi que les coordonnées de la Bbox correspondante voir la figure 8.

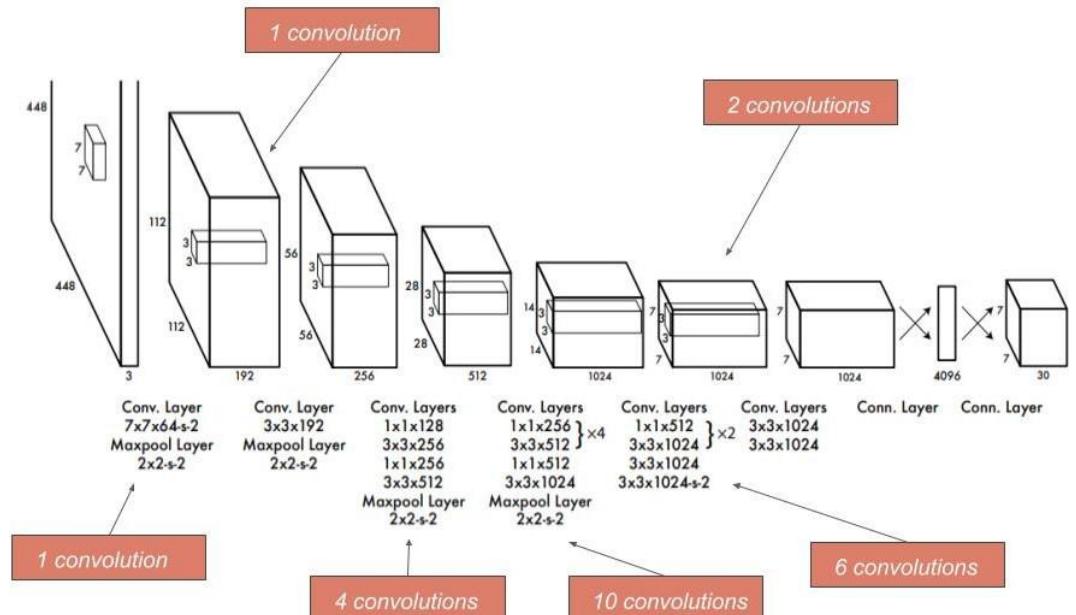


Figure 8 : Architecture de Yolo pour la détection des objets [12]

L'architecture du modèle se compose de 24 couches convolutives pour extraire les caractéristiques suivies de 2 couches denses entièrement connectées pour réaliser la détection d'objets.

- Fusion des prédictions : Les prédictions de chaque cellule sont ensuite fusionnées pour produire une image annotée qui montre les objets détectés avec leurs boxes respectives.

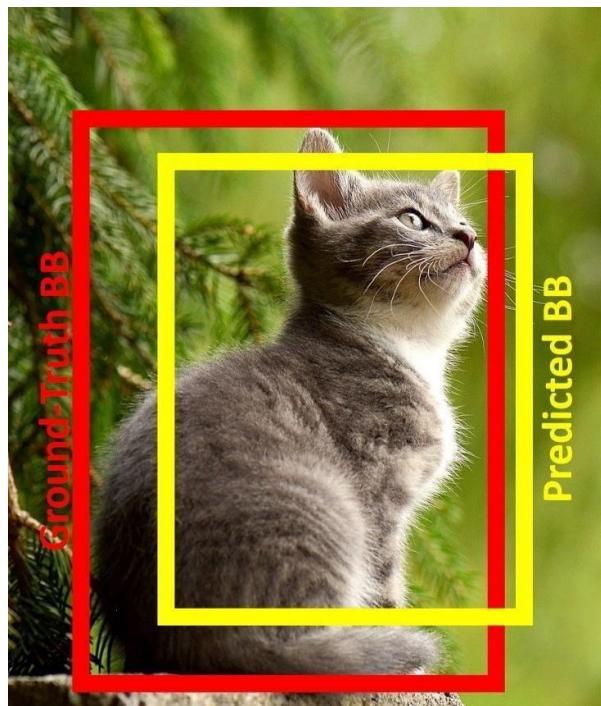


Figure 9 : Boxes de détection d'un objet

- Suppression des doublons et des fausses détections : Enfin, YOLO utilise des techniques de post-traitement pour éliminer les doublons et les fausses détections, en se basant sur des critères tels que la confiance de la détection et la superficie de la boîte englobante.
- Classification des objets : Enfin, les objets détectés sont classés en utilisant des étiquettes prédéfinies, telles que les noms d'objets de la maison (véhicules, animaux, bâtiments, etc.).

4.3. Les Algorithmes de suivi d'objet

Le suivi d'objets est un processus qui consiste à suivre la position et la forme d'un objet au fil du temps dans un flux vidéo. L'algorithme de suivi d'objets YOLO utilise le modèle de détection d'objets YOLO pour détecter les objets dans chaque trame de la vidéo, puis utilise des techniques de suivi pour suivre l'emplacement et la

forme de ces objets d'un frame à l'autre. L'algorithme de suivi d'objets YOLO peut être implémenté en utilisant diverses techniques de suivi, telles que le suivi de Kalman, le suivi par caméra apparente, le suivi par modèle d'objet, etc.

4.3.1. Filtre de Kalman

Dans notre solution, nous avons implémenté l'algorithme du filtre de Kalman, par exemple, une estimation de l'état de l'objet à la trame précédente est utilisée pour prédire sa position à la trame actuelle. Cette prédition est ensuite corrigée à l'aide des données de détection de l'objet à la trame actuelle pour produire une estimation plus précise de sa position. Ce processus est répété pour chaque trame de la vidéo.

4.3.2. Fonctionnement du filtre de Kalman

Les équations fondamentales du filtre de Kalman sont les suivantes :

- État précédent : $x(1)$ est l'estimation de l'état de l'objet à la trame précédente.
- Prédition de l'état $x^p = Ax_k + Ba_k$ (I) est la prédition de l'état de l'objet à la trame actuelle. A est la matrice de transition, qui modèle la dynamique de l'objet et B a(k) est la commande de contrôle qui modèle les influences externes sur l'objet.
- Matrice de covariance précédente : P(k) est la matrice de covariance de l'état précédent, qui décrit l'incertitude sur l'état de l'objet à la trame précédente.
- Prédition de la matrice de covariance : $P_{K+1}^p = AP_kA^t + C_k^{rs}$ (II) est la prédition de la matrice de covariance de l'état à la trame actuelle. Q est la matrice de covariance du bruit d'observation.
- Correction : La mesure de détection de l'objet à la trame actuelle est utilisée pour corriger la prédition. La correction est effectuée en utilisant les équations de mise à jour de Kalman.

- Kalman gain : $K_k = P_k^p + H^T(HP_k^p - C_k^{rm})(-1)$ (III) est le gain de Kalman, qui mesure la confiance dans la mesure. H est la matrice d'observation, qui modélise la relation entre l'état de l'objet et la mesure, et R est la matrice de covariance du bruit de mesure.
- Mise à jour de l'état : $x_k = x_k^p + K_k(z_k + HP^p)$ (IV) est la mise à jour de l'état, où $z(k)$ est la mesure de détection de l'objet à la trame actuelle.
- Mise à jour de la matrice de covariance : $P_k = (I - K_k H)P_k^p$ (V) est la mise à jour de la matrice de covariance, où I est la matrice identité.

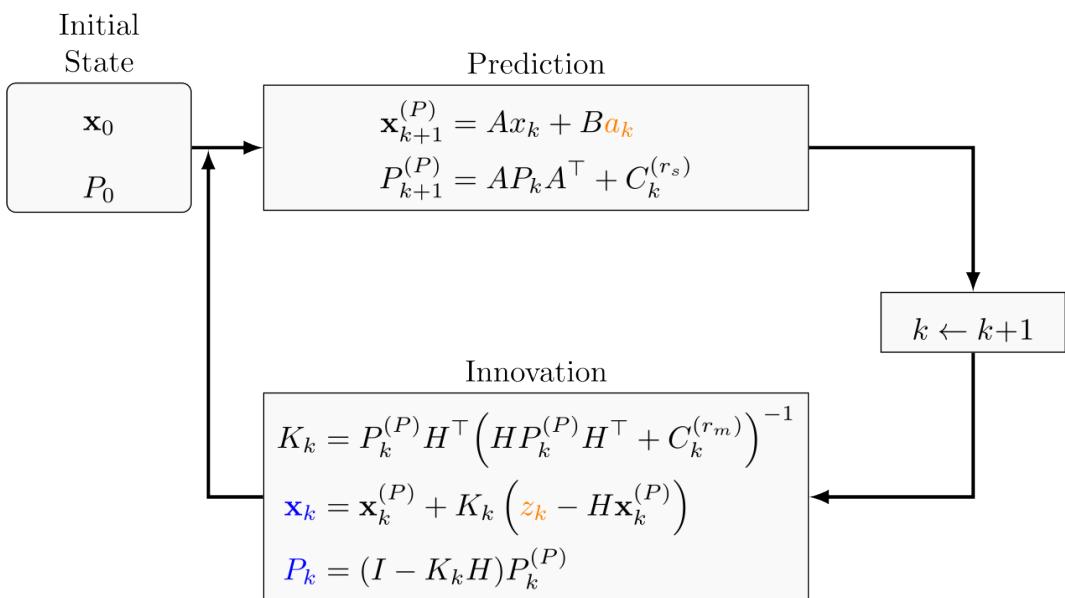


Figure 10 : Équation du filtre de Kalman [9]

Le processus décrit ci-dessus est répété pour chaque trame de la vidéo pour suivre la position et la forme de l'objet au fil du temps comme le montre la figure 11 ci-dessus.

4.4. Mesure de la distance entre la caméra et l'objet

Pour la mesure de la distance entre un objet et la caméra, on avait le choix entre la caméra de profondeur et la caméra stéréo, voir les figures 11 et 12.



Figure 12 : Caméra Stéréo



Figure 11 : Caméra de profondeur

4.4.1. La caméra stéréo

Une caméra stéréo est un type de caméra qui utilise deux objectifs pour créer des images stéréoscopiques, qui sont ensuite utilisées pour mesurer la profondeur. Les deux objectifs sont placés de manière à simuler la vision binoculaire humaine, où chaque œil voit une image légèrement différente.

- La distance entre les deux objectifs est appelée la base stéréo et elle est importante, car elle détermine la précision de la mesure de profondeur. Plus la base stéréo est grande, plus la mesure de profondeur est précise, mais elle peut également être plus sensible aux erreurs de calibrage.
- Pour mesurer la profondeur à partir des images stéréoscopiques, il existe plusieurs techniques, telles que la corrélation stéréo et la triangulation.
- La corrélation stéréo implique la correspondance de chaque pixel dans les deux images stéréoscopiques pour calculer la disparité, qui est une mesure de la différence de position entre les deux images. Puis la triangulation implique la mesure de la distance à partir des angles et des longueurs connus, qui sont déterminés à partir des images stéréoscopiques, voir la figure 13 ci-dessous.

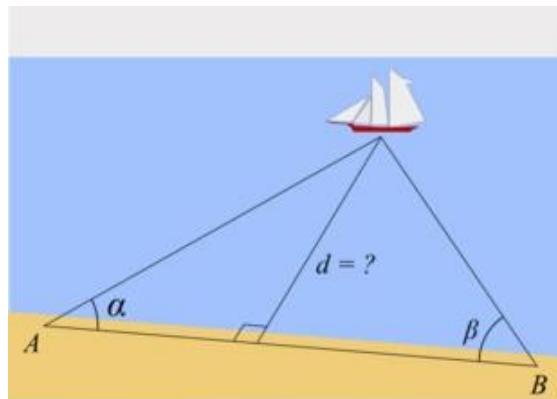


Figure 13 : Triangulation

On a la formule mathématique de suivante :

$$\frac{\sin(\beta)}{PA} = \frac{\sin(\alpha)}{AB} \quad (VI)$$

$$d = \sin(\beta)PB \quad (VII)$$

4.4.2. La caméra de profondeur

Une caméra de profondeur est un type de caméra qui peut fournir des mesures de profondeur pour chaque point d'une image. Elle utilise des techniques telles que la mesure de temps de vol, la lumière structurée ou la stéréo vision (basée sur le principe triangulation) pour déterminer la distance entre l'objet et la caméra [11]. La mesure de temps de vol consiste à émettre un faisceau de lumière et à mesurer le temps qu'il faut pour que la lumière réfléchie revienne à la caméra. Cette technique est rapide et peut fournir des mesures de profondeur précises pour chaque point de l'image. Elle est représentée par la formule :

$$Distance = \frac{\text{Vitesse}}{\text{Temps}} \quad (VIII)$$

La lumière structurée utilise une projection de motifs de lumière sur l'objet, puis mesure la déformation de ces motifs pour calculer la distance entre l'objet et la caméra. Cette technique peut également fournir des mesures de profondeur précises et une résolution élevée, mais elle peut être sensible aux conditions d'éclairage.

La caméra de profondeur que nous avons à notre disposition se base sur le principe de triangulation et la mesure de temps de vol pour évaluer la distance entre objets et la caméra. Le principe de triangulation de cette caméra est similaire à celui de la caméra stéréo.

4.4.3. Choix de la caméra

Dans le cas du suivi et détection d'objets par notre rover, nous choisissons la caméra de profondeur qui est la plus adaptée pour plusieurs raisons :

- elle nous fournit des mesures de profondeur précises pour chaque point de l'image, ce qui nous permet de détecter facilement les objets qui sont à une certaine distance du rover et de les suivre ;
- nous pouvons utiliser la caméra de profondeur pour mesurer la taille des objets, ce qui peut aider le rover à planifier son mouvement et à éviter les collisions ;
- la caméra de profondeur peut nous donner des informations sur la topographie du terrain et les obstacles, ce qui est particulièrement utile pour la navigation de notre rover dans des environnements difficiles ;
- elle peut également nous aider à détecter les changements de hauteur du terrain, ce qui peut être important pour éviter les chutes ou pour planifier le déplacement du rover sur des pentes ;

Nous pouvons utiliser la caméra de profondeur aisément avec des bibliothèques YOLO.

5. Évaluation des performances du modèle de vision par ordinateur

Une fois que nous avons customisé notre modèle de détection d'objet sur la base des images recueillies, l'on doit vérifier les performances lorsqu'il effectue cette tâche. Pour cela, l'on a recouru à une métrique nommée la matrice de confusion. Une matrice de confusion (Confusion Matrix en anglais) ou tableau de contingence est un résumé des résultats de prédictions sur un problème de classification.

Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles. Ainsi, cette matrice permet de comprendre de quelle façon le modèle de détection est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreurs de prédictions commises. Dans notre cas, elle fait une confrontation des prédictions des objets détectés positivement et négativement. Tous les paramètres d'estimation de la matrice de confusion sont basés sur 4 entrées, à savoir :

- Les vrais positifs (VP) : le nombre ou pourcentage des objets détectés véritablement positifs.
- Les vrais négatifs (VN) : le nombre ou pourcentage des objets détectés véritablement négatifs.
- Les faux positifs (FP) : le nombre ou pourcentage des objets détectés positifs alors qu'ils sont négatifs.
- Les faux négatifs (FN) : le nombre ou pourcentage des objets détectés classés négatifs alors qu'ils sont positifs.

Ci-dessous, le tableau suivant présente la matrice de confusion de la détection d'un sac (Bag).

Tableau 2 : Matrice de confusion

Prédictions / Réelles (%)	Positifs	Négatifs
Positifs	VP : 82	FP : 18
Négatifs	FN : 18	VN : 5,5

À la vue de notre échantillon de données précisément la classe pour la détection des sacs chaque classe, l'on constate que le nombre de FP et FN est égal à 18%. Puis le nombre de VP et VN sont respectivement de 82% et 5,5%.

Le nombre de FP et FN nous indique qu'un mauvais étiquetage est fait au niveau d'un nombre de maladies. Le nombre de VP étant de 82% montre que notre modèle classe les maladies.

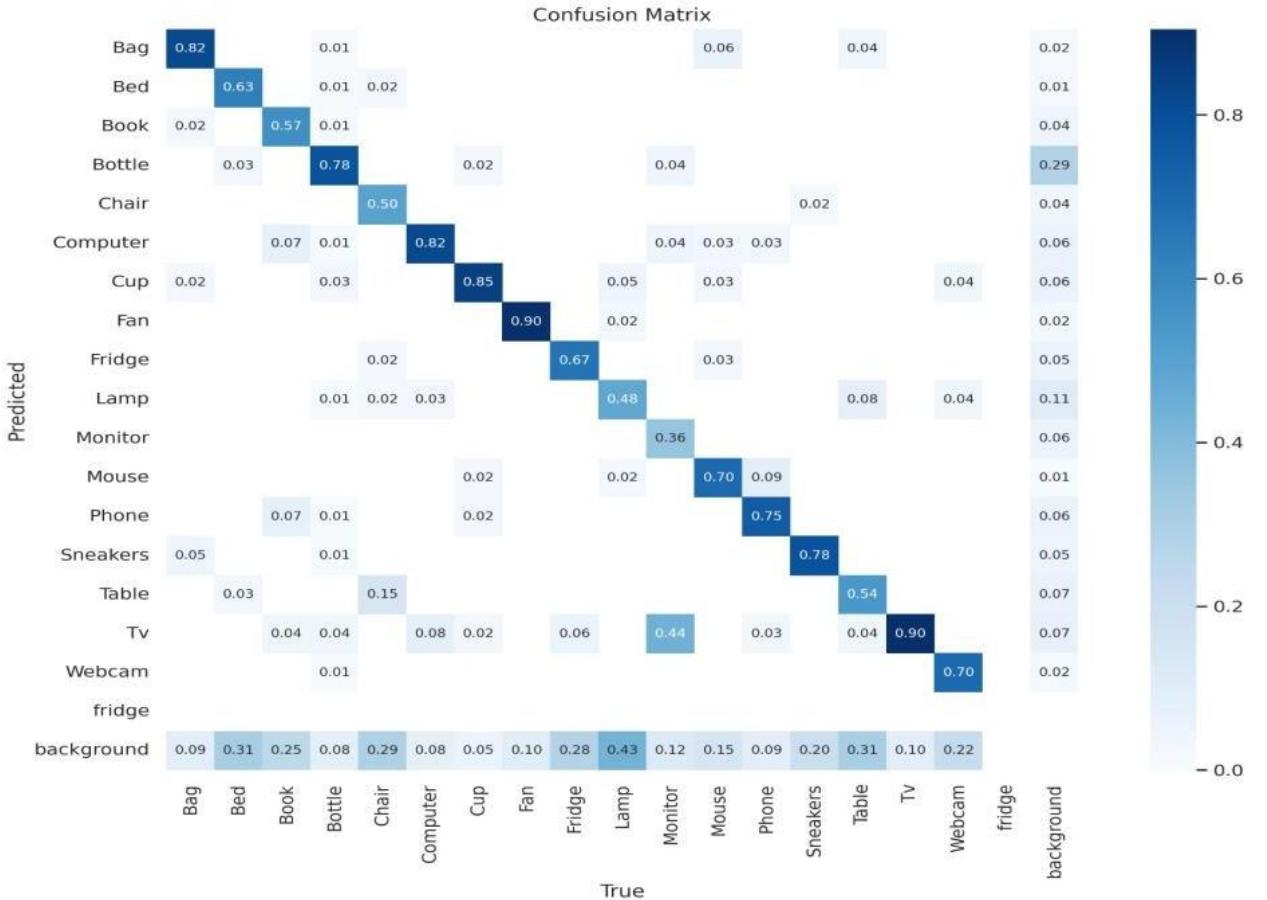


Figure 14: Matrice de confusion

De plus, les différentes courbes présentées ci-dessous permettent de savoir comment notre modèle se comporte au travers de la réduction du coût des boîtes de détections et la précision et le rappel. La première ligne présente le comportement du modèle sur l'ensemble d'entraînement et la seconde sur l'ensemble de validation. L'ensemble de validation permet d'obtenir les meilleures performances pour notre modèle de détection d'objet.

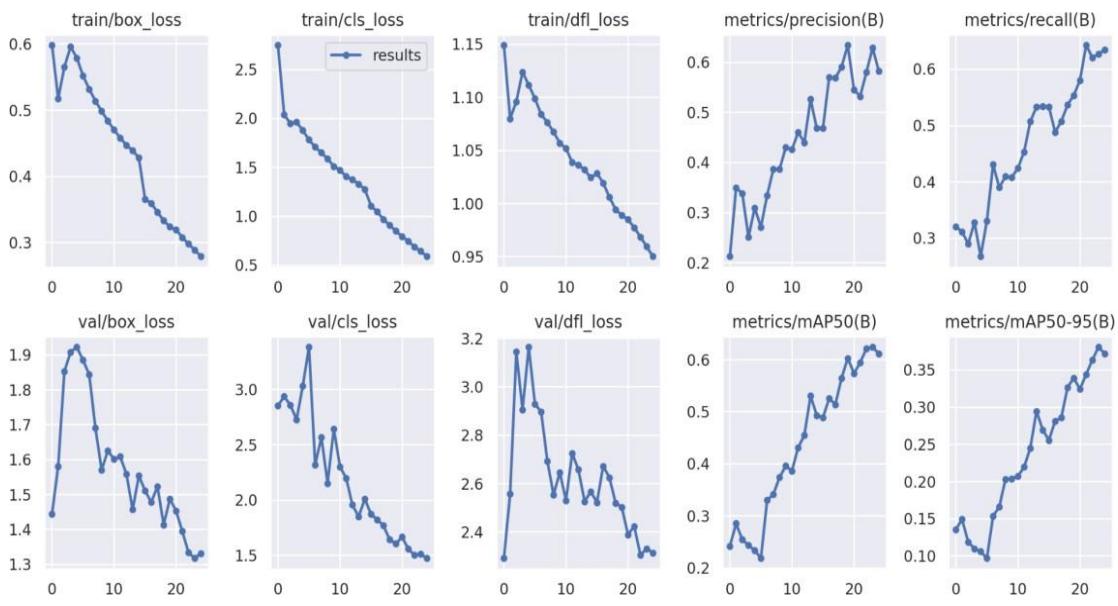


Figure 15 : Évaluation de Performance

Pour mesurer les métriques du rapport de classification qui sont le rappel, la précision, la spécificité, le taux de réussite, nous utilisons les différents paramètres de la matrice de confusion.

5.1. Rappel

Le rappel (sensibilité) est défini comme le rapport du nombre total de vrais positifs par le nombre total de vrais positifs plus le nombre total de faux négatifs au sein de l'ensemble de données.

$$Rappel = \frac{VP}{VP + FN} \quad (IX)$$

5.2. Précision

La précision est définie comme le rapport du nombre total de vrais positifs classé correctement par le nombre total de prédictions positives.

$$Précision = \frac{VP}{VP + FP} \quad (X)$$

5.3. Spécificité

La spécificité définit le taux de vrais négatifs dans l'ensemble de données contenant les prédictions négatives.

$$Spécificité = \frac{VN}{VN + FP} \quad (XI)$$

5.4. Taux de réussite

Le taux de réussite définit le nombre total de vraies prédictions dans l'ensemble de données totales.

$$Taux\ de\ réussite = \frac{VP + VN}{VP + VN + FP + FN} \quad (XXII)$$

5.5. Courbes

Les courbes que nous allons interpréter maintenant permettent d'avoir une aperçue du comportement des métriques annoncées ci-dessus.

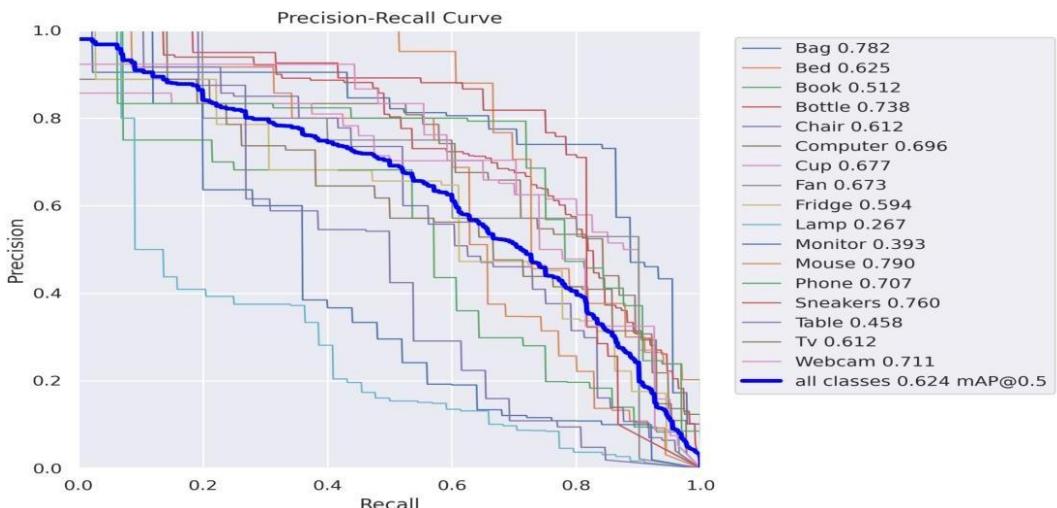


Figure 16 : Courbe précision- rappel

Sur la courbe ci-dessous, on a le mAP qui est la moyenne des scores AP (Average Precision) pour chaque classe de l'ensemble de données.

AP mesure la capacité du modèle à détecter les objets d'une classe particulière dans l'image. Ici, nous pouvons constater que le mAP de notre modèle est de 62,4%.

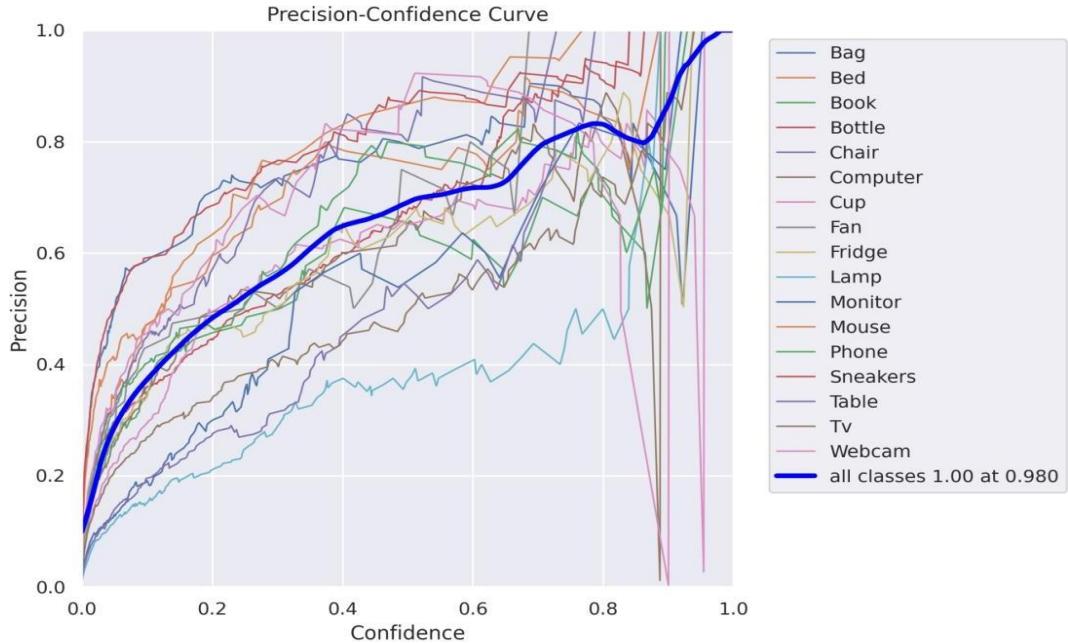


Figure 17 : Courbe précision-seuil de confiance

Sur figure 17, l'on remarque par la ligne en bleu que plus l'intervalle de confiance croît, plus la précision augmente. Pour un seuil de confiance de 100%, nous avons une précision à 98%.

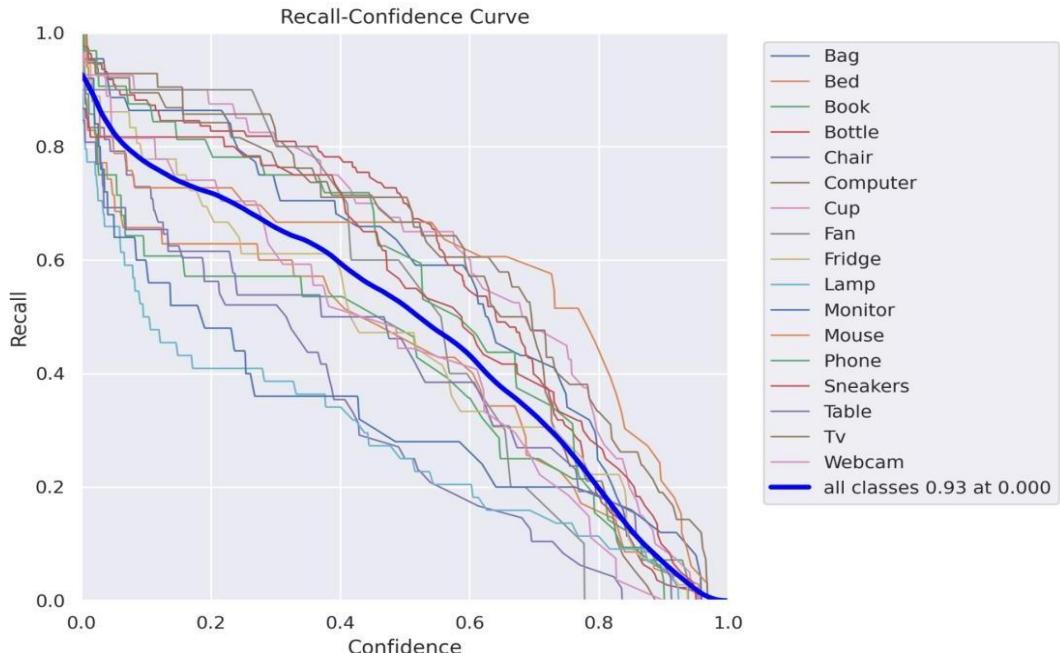


Figure 18 : Courbe rappel-seuil de confiance

Sur figure 18, c'est le contraire : plus l'intervalle de confiance croît, plus le rappel diminue. Pour un seuil de confiance de 93%, nous avons un rappel à 0%.

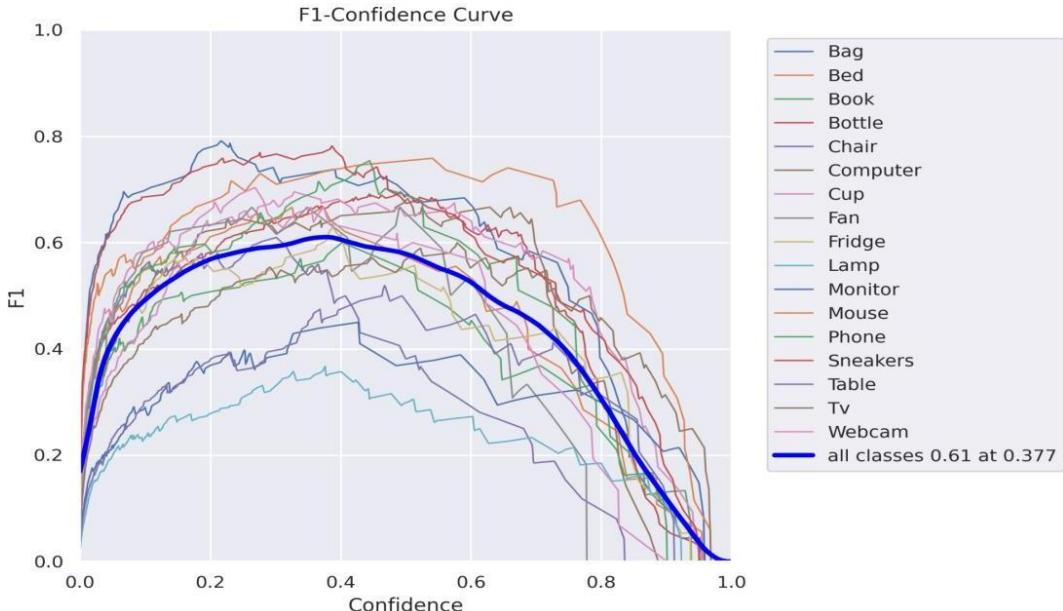


Figure 19 : Courbe spécificité-seuil de confiance

Dans cette courbe (figure 19), pour un seuil de confiance à 61,5%, nous avons la plus haute spécificité 37,7%.

IV. SÉCURISATION DU SYSTÈME

1. Vulnérabilités du système

Le système a été soumis à une évaluation approfondie de sa sécurité, mettant en évidence certaines vulnérabilités :

- Une analyse des ports et services avec Nmap révèle que le port 22 est ouvert pour SSH.
- Toutes les machines appartenant au même réseau que le robot peut se connecter à celui-ci via SSH.
- L'interface web présente une vulnérabilité potentielle face à une attaque par force brute.

2. Sécurisation de la connexion par SSH

2.1. Spécifier une seule adresse MAC à se connecter au robot

Dans le but de renforcer la sécurité du robot, il a été envisagé de spécifier une seule adresse MAC autorisée à se connecter :

- Pour établir une connexion SSH avec le robot, l'utilisateur doit fournir ses identifiants.
- La première machine à se connecter pour effectuer les configurations peut éditer le fichier de configuration SSH du robot, situé à l'emplacement /etc/ssh/sshd_config, et associer l'adresse MAC à la commande "AllowUsers".
- Par exemple, si l'adresse MAC est "00:11:22:33:44:55", l'administrateur doit ajouter "AllowUsers username@00:11:22:33:44:55".
- Enregistrer ensuite le fichier de configuration et relancer le service SSH pour appliquer les modifications. Cette mesure permet de restreindre l'accès au robot à une seule adresse MAC autorisée, renforçant ainsi sa sécurité.

2.2. Authentification par SSH

Il y a deux types d'authentification :

- l'authentification par mot de passe ;
- l'authentification par clés.

Dans notre cas, nous optons pour l'authentification par clés basé sur la cryptographie asymétrique, car elle est plus sûre. La clé privée est stockée localement chez l'utilisateur, ce qui la rend difficile d'accès. Aussi, l'on peut protéger cette clé au cas où un hacker réussit à y accéder.

Mode de fonctionnement :

- Génération des clés : Un utilisateur génère une paire de clés publique et privée, et la clé publique est partagée avec l'autre utilisateur tandis que la clé privée est conservée.

- Le deuxième utilisateur quant à lui ajoute la clé publique reçue dans un fichier d'autorisation (fichier qui spécifie les clés publiques autorisées à se connecter).
- A l'authentification du premier utilisateur, il utilise sa clé privée pour signer la demande de connexion, et si le deuxième utilisateur réussi à faire une correspondance avec la clé publique qu'il a enregistrée, alors la connexion est acceptée.

3. Sécurisation des fichiers de configuration

En supposant qu'un hacker réussi à s'authentifier, il pourra accéder aux fichiers de configuration pour perturber le système. Alors nous faisons une restriction sur les fichiers de configurations en les octroyant les droits de propriétaire sur les fichiers (root). Dans ce cas, tout accès non autorisé aux fichiers nécessitera une escalade de privilège.

4. Sécurisation de l'interface Web

Pour assurer la sécurité de l'interface Web, plusieurs mesures ont été mises en place:

- Une authentification des utilisateurs est requise pour accéder à l'interface.
- Les authentications sont confirmées par mail pour renforcer la sécurité.
- Les utilisateurs sont sensibilisés à l'ingénierie sociale afin de prévenir les tentatives de manipulation.
- Un antivirus est installé sur l'appareil où est déployée l'interface Web pour détecter et prévenir les menaces potentielles.

Conclusion partielle

Au stade de notre étude, nous retenons qu'après une analyse approfondie, nous avons pu concevoir notre système de perception. Dans la suite du travail, il sera question de présenter les résultats obtenus et de les évaluer.

TROISIÈME PARTIE : RÉSULTATS ET DISCUSSION

CHAPITRES VIII : ENVIRONNEMENT DE DÉVELOPPEMENT

Ce chapitre nous présente l'environnement de travail.

I. ENVIRONNEMENT MATÉRIEL

1. Machine utilisée

Le tableau ci-dessous présente les caractéristiques de deux ordinateurs portables dont nous disposons durant la réalisation du projet.

Tableau 3 : Caractéristiques des machines

Ordinateurs	Caractéristiques
	HP Intel i7 11è Generation, 16GB RAM, 1TB SSD Ecran: 15.6, Tactile GPU :2GB GeForce: MX350
	Macbook pro M1 Résolution d'affichage : 1080*1920 Type de disque dur : SSD Capacité mémoire vidéo : 16 GB Processeur : 2,53 GHz Taille de l'écran : 15,6" RAM: 16 GB Wi-Fi: Wi-Fi 2,4 GHz/5G

2. Pièce de la structure

Pour la réalisation de la structure, nous avons utilisé diverses pièces et matériaux. Le tableau 7 présente la liste complète de ce qui a été utilisé. Une description plus détaillée est donnée en annexe 1.

Tableau 4 : Liste complète des éléments de la structure du robot

Nom	Description
Tige filetée	Tige filetée acier zingué, longueur 500*10 (longueur*diamètre), pour axe du rocker
Roulement à bille	Roulements 6200, 606, et 9300 pour la rotation des axes du rocker, bogie et du différentiel
Contreplaqué	Pour concevoir la plateforme du robot
Vis	Permettent d'assembler les découpes en contreplaqué et les pièces du bras
Boulons et écrous	Servent d'axes aux roulements et servent à fixer les roues
PLA	Filament plastique pour imprimer les pièces en 3D
Bombe de peinture	Bombe de peinture noir brillant pour peindre la plateforme du rover
Tube en aluminium	Tube en aluminium provenant d'un balai à nettoyer

3. Composants électronique et mécanique

Ce projet est principalement électronique et mécanique. Ainsi, nous présentons dans le tableau 8 l'ensemble des composants électroniques et mécaniques qui ont permis de concevoir le projet.

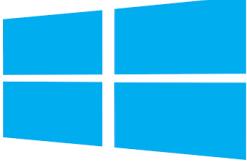
Tableau 5 : Liste des composants électronique et mécanique

Nom des composants	Description
Carte Arduino	Arduino Uno et Méga pour contrôler qui serve de processeur aux couches navigation et E/S d'environnements
Raspberry Pi	Raspberry Pi 4 B+ pour le contrôle de tout le système
PCA9685	Permet de contrôler les servo moteur du bras
DRV8825	Contrôle le moteur stepper la base du bras robotique
BTS7960	Contrôleur des moteurs des roues
HC SR04	Capteur à ultrason pour calculer la distance à un objet
CJMCU	Capteur de CO2
DHT 11	Capteur d'humidité et de température
LED	Permet d'avoir un signal lumineux sur le robot
Buzzer	Dispositif piézoélectrique permettant d'obtenir un signal sonore
Ecran nextion	Ecran tactile permettant de configurer le robot et d'avoir des informations d'état
Caméra de profondeur	Caméra Intel, RealSENSE pour la vision par ordinateur
Capteur Lidar	MB1RT pour la cartographie de l'environnement
NFR24L01	Module radiofréquence pour la commande à distance
HC-05	Module Bluetooth pour permettent au bras de communiquer avec la couche intelligence
Shield Ethernet	Permet de donner la connexion Ethernet à la carte Arduino
Motoréducteurs	Permettent de faire tourner les roue
Servo moteur	Représente les articulations du bras robotique
Moteur pas à pas	Sert à faire tourner la base du bras
Plaque perforée	Plaque PCB pour réaliser le circuit de distribution d'énergie

II. ENVIRONNEMENT LOGICIELS

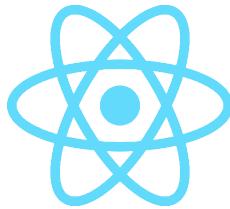
1. Système d'exploitation

Tableau 6 : Système d'exploitation

Système d'exploitation	Description
	Ubuntu est le système d'exploitation open source moderne sur Linux pour le serveur d'entreprise, le bureau, le cloud et l'IoT.
	Le système d'exploitation Windows développé par Microsoft. Il est conçu pour gérer les ressources matérielles et logicielles d'un ordinateur personnel (PC) ou d'un serveur.
	Le système d'exploitation macOS est un système d'exploitation propriétaire développé par Apple Inc. Il est conçu pour les ordinateurs Macintosh d'Apple et est basé sur le noyau Unix.
 ubuntu MATE	Ubuntu MATE est conçu pour fonctionner sur une large gamme de matériel, des ordinateurs anciens et peu performants aux systèmes informatiques modernes et puissants. Il comprend un ensemble complet d'applications préinstallées, notamment des logiciels de productivité, de navigation web, de messagerie électronique, de gestion de fichiers...

2. Langages de programmation

Tableau 7 : Langages de programmation

Langages de programmation	Description
	JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre une partie essentielle des applications web. Il s'agit du langage informatique qui nous a permis de dynamiser notre application Web.
	React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état.
	Python est un langage de programmation interprété, orienté objet, de haut niveau avec une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées à un typage dynamique et à une liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation comme langage de script.
	Django est un framework Web Python de haut niveau qui encourage un développement rapide et une conception propre et pragmatique [32], s'appuyant sur les technologies comme HTML et CSS. Cet outil intervient dans la conception de l'application web.

	<p>ROS (Robot Operating System) est un cadre de développement logiciel open-source pour la construction de robots et de systèmes robotiques. ROS fournit une collection de bibliothèques, d'outils et de processus pour aider les développeurs à créer des logiciels pour des robots autonomes ou semi-autonomes.</p>
	<p>MongoDB est un système de gestion de base de données NoSQL (non relationnel) open-source, conçu pour stocker et gérer des données de manière flexible et évolutive. MongoDB utilise une structure de données appelée BSON (Binary JSON) pour stocker les données, qui est un format binaire pour représenter les objets JSON (JavaScript Object Notation).</p>

CHAPITRE IX : PRÉSENTATION DES RÉSULTATS

I. PRÉSENTATION DU MODÈLE DE VISION PAR ORDINATEUR

1. Détection des objets

Nous avons appliqué des détecteurs d'objets sur notre ensemble de test d'images, comme résultat l'on observe sur 4 images extraits de cet exemple de tests que notre modèle à détecter correctement, les trois premières images représentent respectivement une chaise, une souris d'ordinateur et une paire de basket. Une mauvaise détection a été effectuée sur l'image 1 représentants un téléphone au lieu d'une souris d'ordinateur.



Figure 23 : Détection d'un téléphone à 45%



Figure 22 : Détection d'une paire de basket à 78%



Figure 21 : Détection d'une souris à 71%



Figure 20 : Détection d'une chaise à 35%

2. Suivi des objets

Ici, nous présentons les résultats du suivi d'une bouteille par le biais de notre algorithme de suivi d'objets. L'id 17 permet de signifier que c'est le même objet qui est suivants dans les 3 environnements présentés par les figures suivantes.



Figure 24 : Suivi de la bouteille de base

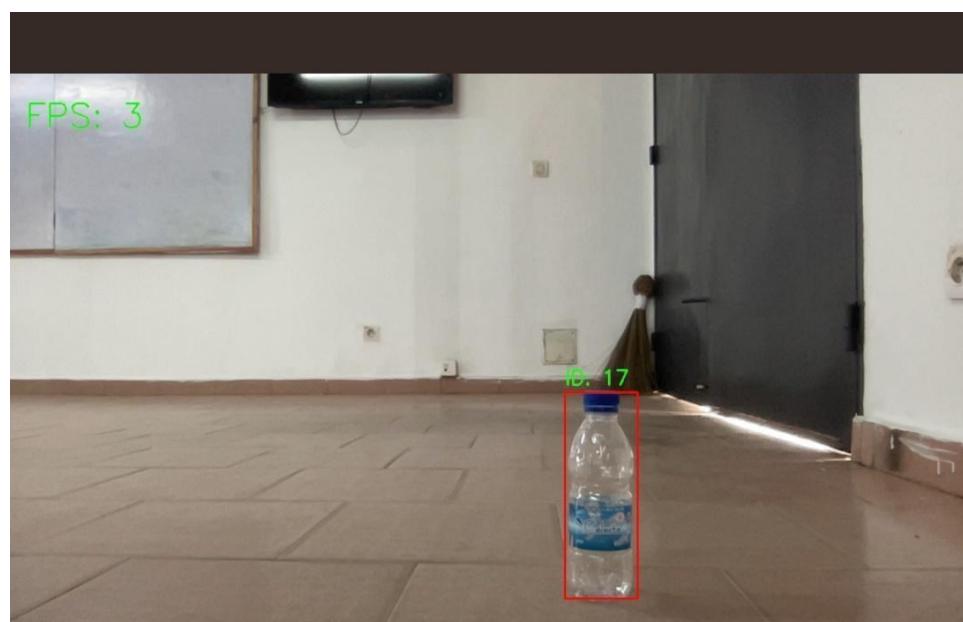


Figure 25 : Suivi de la bouteille mouvement vers la droite

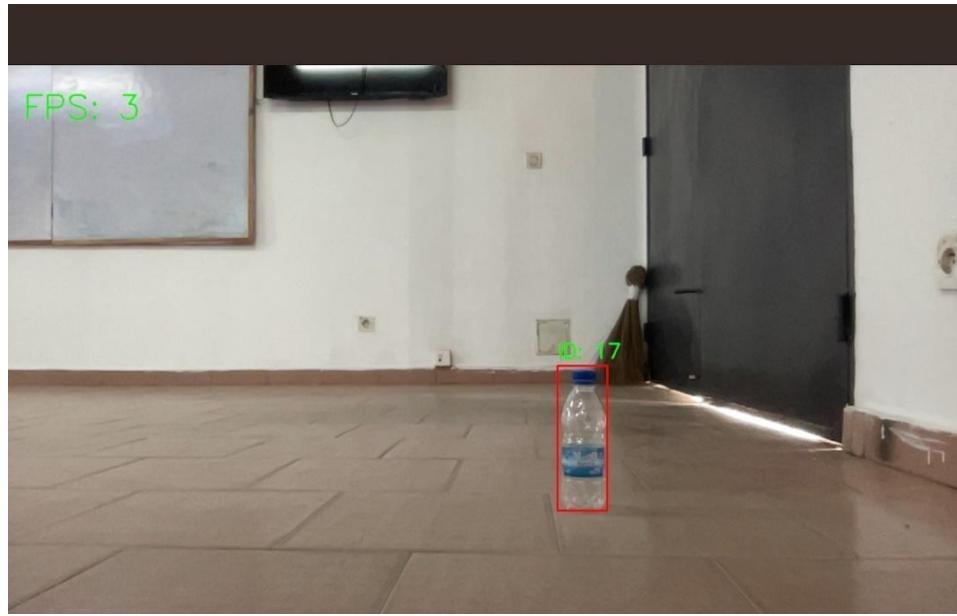


Figure 26 : Suivi de la bouteille mouvement vers l'arrière

II. PRÉSENTATION DU ROBOT

1. Présentation mécanique

Le robot mobile est un système mécanique qui mesure 35 cm de hauteur et 48 cm de diamètre. Il est équipé d'un système de transmission à deux roues, connu sous le nom de differential drive, qui permet au robot de se déplacer dans différentes directions. Les roues sont contrôlées par des moteurs, qui sont connectés à un Arduino pour la gestion de la commande des moteurs et des encodeurs pour la mesure de la vitesse et de la position des roues.

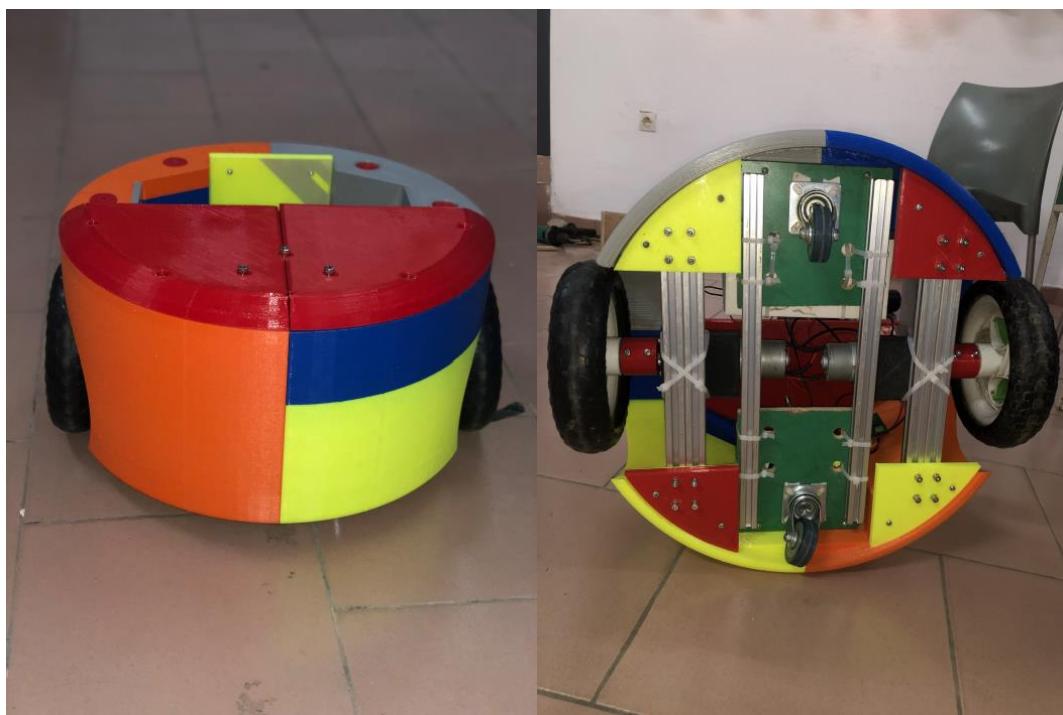


Figure 27 : Nouveau robot

La mécanique est donc principalement constituée de la structure du robot, des roues et des moteurs, ainsi que des encodeurs pour mesurer la vitesse et la position des roues.

2. Présentation électronique

Le système électronique est essentiellement constitué d'un Raspberry Pi, qui est utilisé comme ordinateur embarqué pour le traitement des données de perception et pour le contrôle du robot. Le Raspberry Pi est connecté à l'Arduino via une interface série pour la gestion des moteurs et des encodeurs.

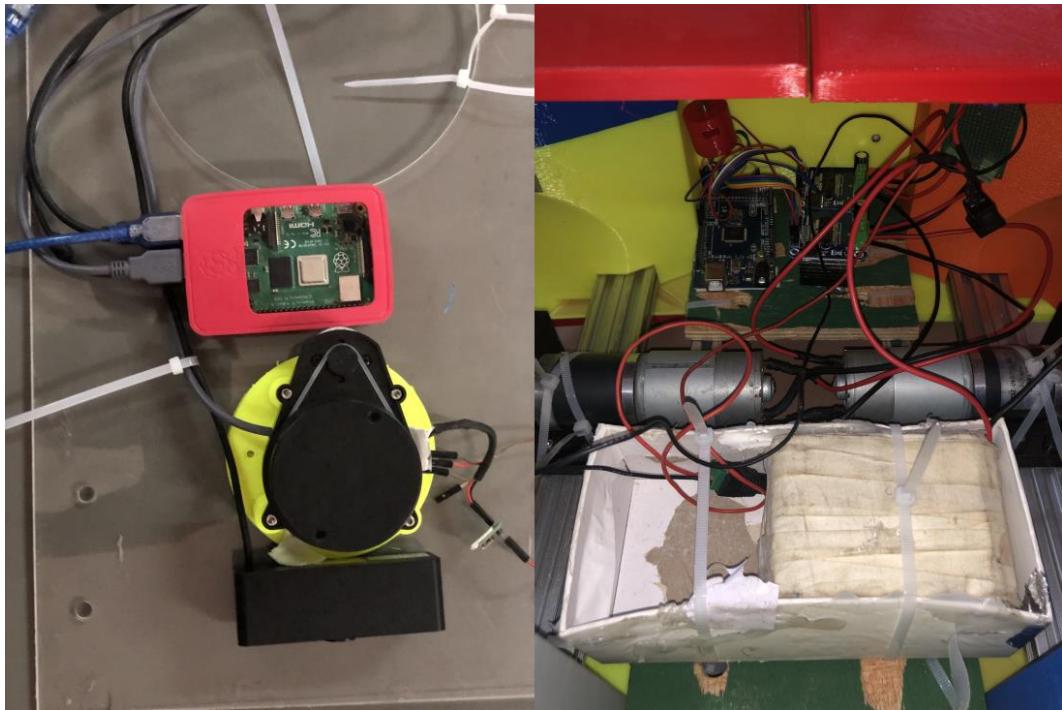


Figure 28 : Architecture électronique du robot

Le système électronique comprend également des capteurs de perception tels qu'un lidar et une caméra de profondeur pour percevoir l'environnement du robot. Le lidar est utilisé pour obtenir une vue à 360 degrés de l'environnement du robot, tandis que la caméra de profondeur est utilisée pour mesurer les distances et la géométrie de l'environnement. Les données provenant de ces capteurs sont traitées par le Raspberry Pi pour la navigation et le contrôle du robot.

III. PRÉSENTATION DU LOGICIEL

1. Logiciel du robot

La partie logicielle du projet se divise en deux parties distinctes. Tout d'abord, il y a la partie logicielle du robot, qui se compose de ROS, YOLO et OpenCV. ROS est utilisé pour la communication entre les différents nœuds du robot, notamment entre les capteurs et les moteurs. YOLO est utilisé pour la détection d'objets et la reconnaissance d'images en temps réel, tandis qu'OpenCV est utilisé pour le traitement d'images et la perception de l'environnement.

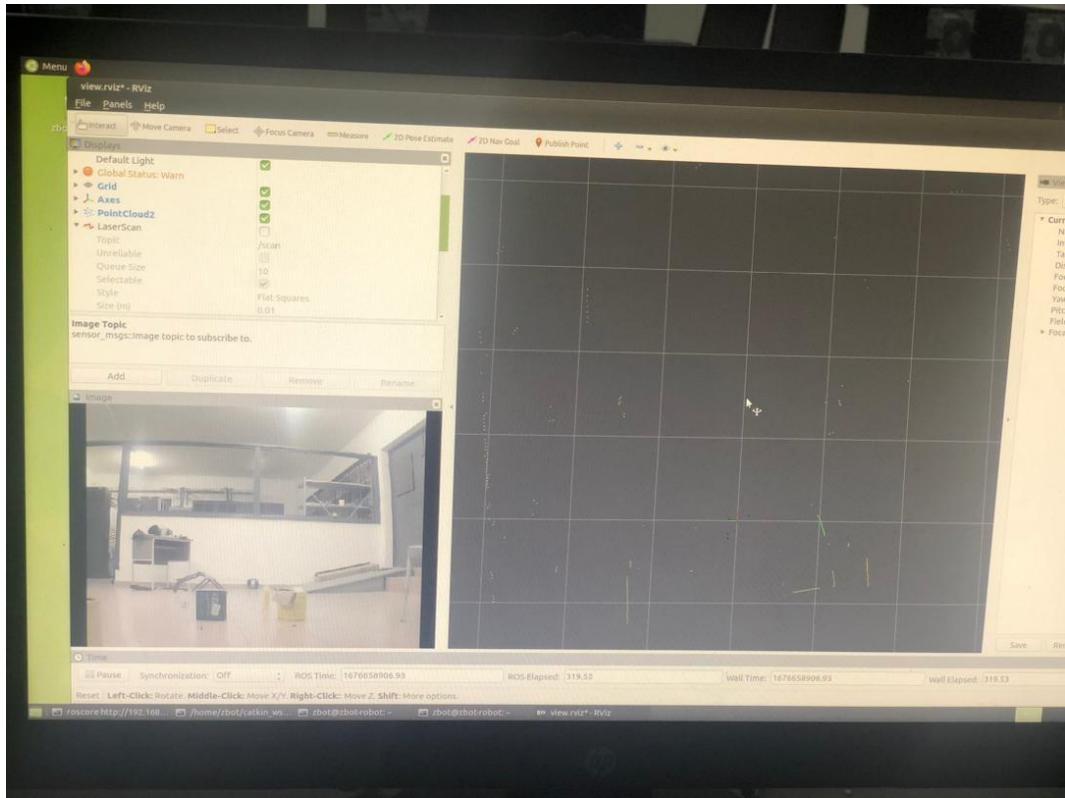


Figure 29 : Logiciel Robot

2. Application web de commande

Ensuite, il y a la partie de l'interface web, qui utilise Django et React. La page web hébergée en local sur le Raspberry Pi permet à l'utilisateur de commander le robot à distance via des boutons de commande, d'afficher la carte de l'environnement et de visualiser le retour vidéo de la caméra et du Lidar. Pour la transmission en temps réel de la caméra et du Lidar sur la page web, nous utilisons WebRTC. Le module `ros_webRTC` est utilisé pour publier les données du Lidar et de la caméra, tandis que `ros_django` permet la communication entre le serveur Web Django et ROS. L'interface de commande et d'observation de notre robot est représentée sur la figure suivante. Nous avons des boutons de directions et un retour vidéo de l'environnement perçu.

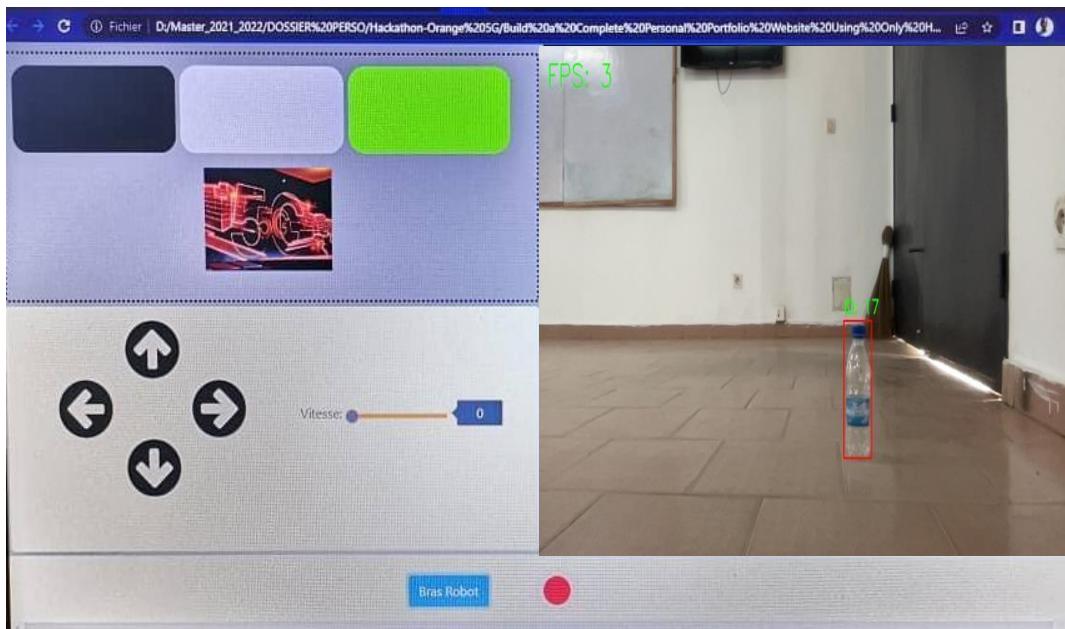


Figure 30 : Interface de commande

CHAPITRE X : DISCUSSION ET ESTIMATION FINANCIÈRE

I. DISCUSSION DES RÉSULTATS

1. Modèle de vision par ordinateur

Aux vues des résultats obtenus suite à l'évaluation des performances ($mAP = 62,4\%$ et autres) de notre modèle de détection d'objets en utilisant les images. Nous pouvons constater que le premier modèle que nous réalisons en customisant Yolo V8 n'a pas encore atteint les performances souhaitées. Ces résultats auront des répercussions immédiates sur le suivi des objets et également sur la localisation des objets au sein de l'environnement de navigation. Pour accroître les performances de notre modèle, nous devons :

- augmenter le nombre de nos images ;
- varier la distribution des images collecte ;
- entraîner notre modèle sur plus d'époque ;
- implémenter d'autres algorithmes pour la détection d'objets.

2. Robot

Nous avions conçu un robot mobile capable d'assister les personnes à mobilité réduite dans certaines tâches. Cependant, ce robot n'était doté que de la commande manuelle et manquait d'autonomie dans sa commande. Ainsi, nous avons conçu un système qui lui permet désormais de percevoir l'environnement dans lequel il se trouve puis de se déplacer dans celui-ci en évitant les obstacles et en reconnaissant même des objets du quotidien. Ces fonctionnalités ajoutées rendent beaucoup plus ludique l'utilisation de notre robot. Cependant, le robot n'est pas encore précis dans l'exécution de ces tâches. En effet, la réception tardive des équipements de perception et de navigation (Caméra de profondeur et Encodeur) ne nous a pas permis d'ajuster aux mieux les paramètres de notre module afin qu'il soit optimal. De plus, le fait que le ZBOT soit en construction,

nous avons dû construire un robot pour tester le module de perception. Ainsi, il faudra revoir certains paramètres tels que la taille du robot, le type d'entraînement, le poids et bien d'autres afin qu'il réagisse bien sur le robot cible. Enfin, pour la visualisation du flux vidéo du robot, nous avons opté pour du WebRTC pour la transmission sur le cloud. Puis, nous pouvons le récupérer de partout dans le monde. Cependant, la contrainte avec cette technologie c'est qu'on a besoin d'avoir accès au flux vidéo du robot.

3. Interface Web

L'application web de commande de robot présenté dans ce document permet aux utilisateurs de contrôler le robot à distance de manière intuitive et conviviale grâce à des boutons de commande, une carte de l'environnement et un retour vidéo. L'utilisation de technologies appropriées telles que Django et React permet de séparer clairement la logique de l'interface utilisateur et le traitement des données. Cependant, des inconvénients tels que les problèmes de latence peuvent survenir lors de l'utilisation des services en ligne. De plus, il est important de noter que l'utilisation de la connexion WI-FI peut être sujette à des problèmes de fiabilité, tels que les interruptions de signal ou des interférences.

II. ESTIMATION FINANCIÈRE

Dans le tableau qui suit, nous avons fait une évaluation financière de notre projet. Pour cette évaluation, nous prenons en compte les composants du robot, les machines et outils utilisés et la main d'œuvre, ce qui nous fait un total de 6 389 125 FCFA. Les détails sont consignés dans l'annexe 3.

Tableau 8 : Estimation financière

Désignation	Prix (FCFA)
Composants Du Robot (Annexe 3)	712 125
Machines Et Outils Utilisés (Annexe 4)	2 677 000
Main D'œuvre (03 Mois)	3 000 000
Estimation Financière	6 389 125

Conclusion partielle

Nous avons quelques limites de notre système qui seront prises en compte dans nos perspectives. Aussi, le bilan financier du projet s'élève à 6 38 125 FCFA.

CONCLUSION

Au terme de notre projet, nous retenons que, la navigation des robots mobile est domaine qui à de nombreuses applications pratiques, notamment dans l'assistance aux personnes à capacité réduite ou aux handicapées. L'objectif de ce projet était donc de d'améliorer les compétences fonctionnelles du robot ZBOT conçu spécialement dans le but d'apporter une assistance aux personnes à mobilité limitée en lui permettant de s'émouvoir de façon autonome et sécurisée. La solution proposée était donc de concevoir un module de perception sécurisé pour une navigation autonome du robot ZBOT. Dans le déroulement de ce projet, nous avons d'abord fait une étude générale sur les différentes technologies existantes pour la navigation autonome des robots mobiles, suivie de la présentation de quelques robots mobiles dotés de cette fonctionnalité. Cette première partie s'est achevée par la présentation du cahier de charge qui met en évidence les objectifs et besoins fonctionnels du projet. Ensuite, nous avons mené une étude conceptuelle qui nous a imposé une étude du robot ZBOT pour la détermination des limites. Cette étude a abouti à la conception du module de vision par ordinateur, à la conception de l'interface web ainsi que le Hardware. En outre, nous avons défini quelques mesures de sécurité pour la navigation sécurisée du robot. Notre projet s'est clôturé par les tests, le déploiement et l'évaluation de la solution. Par ailleurs, nous ne sommes pas restées en marge de quelques difficultés pendant la réalisation de notre solution.

Le module de perception qui a été mis en place assure effectivement au robot une navigation autonome en cartographiant avec précision son environnement. Cependant, Toujours dans le souci de faciliter la vie de l'utilisateur cible. Il serait plus intéressant d'automatiser le processus de détection et de récupération d'objets par le bras mécanique. Aussi, il faudra configurer une communication bidirectionnelle qui permet d'établir une conversation vocale entre le robot et l'utilisateur.

BIBLIOGRAPHIE

- [1] Hassan Nehme, **Perception de l'environnement de navigation pour un robot autonome en milieu agricole**, Sciences de l'information et de la communication Normandie Université, 202, Français. NNT : 2022NORMR019 . tel-03773581.
- [2] Marwa Chakroun, **Perception de scène par un système multi-capteurs, application à la navigation dans des environnements d'intérieur structuré. Modélisation et simulation**, Ecole nationale supérieure Mines-Télécom Atlantique; École nationale d'ingénieurs de Sfax (Tunisie), 2022, Français, NNT: 2021IMTA0278, tel-03519333.
- [3] Zadi Jonathan, **Conception et réalisation d'un robot d'assistance aux personnes en situation de mobilité réduite**, Licence en Système Réseau Informatique et Télécommunication. École Supérieure Africaine des Technologies de l'Information et de la Communication (ESATIC), 2021, 92 p.
- [4] Dabone Souleymane, **Mise en œuvre d'un module d'anamnèse médical basé sur un modèle d'apprentissage automatique (AA) : Cas du service médecine général de la Grande Clinique de Dokui**, Licence en Système Réseau Informatique et Télécommunication. École Supérieure Africaine des Technologies de l'Information et de la Communication (ESATIC), 2021, 79 p.
- [5] **Fiche technique Intel RealSense D400 Series**, Numéro du document: 337209-005, 2019, 119
- [6] LENTIN Joseph, **Learning Robotics using Python**, 2018, 273 pages

WEBOGRAPHIE

- [7] <https://ultralytics.com> : Presentation de YOLO.
- [8] <https://opencv.org> : Presentation d'OpenCV.
- [9] <https://thekalmanfilter.com/kalman-filter-explained-simply> : Equation du filtre de Kalman.
- [10] <https://fr.ids-imaging.com/ensenso-stereo-3d-camera.html> : Caméra stereo.
- [11] <https://www.intelrealsense.com> : Caméra de profondeur.
- [12] <https://blent.ai/detection-images-yolo-tensorflow> : Detection d'objets
- [13] https://www.academia.edu/3168672/Perception_pour_la_robotique_mobile_en_environnement_humain : **Thèse sur la perception pour la robotique mobile en environnement humain**.
- [14] <https://meanwhile-france.com> : **Robot de Meanwhile**
- [15] <https://www.cnetfrance.fr> : **Robot de surveillance de Knightscope**.
- [16] GANGLOFF (Jacques), **Cours de robotique de manipulation**, Télécom Physique Strasbourg, Strasbourg, 2016, [document électronique],
<https://drive.google.com/file/d/0ByaIRKZHDhW-bW5NbkdGeXpuLVU/view?usp=sharing&resourcekey=0-IxrY-zt8BIOrCXAeN4rEJQ>.
- [17] TZAFESTAS (Spyros G.), **Introduction to Mobile Robot Control**, [document électronique], School of Electrical and Computer Engineering, National Technical University of Athens, Athènes, Grèce, <https://b-ok.africa/book/2278383/674a95>.
- [18] SIEGWART Roland et al. **Introduction to Autonomous Mobile Robots**, Seconde édition, The MIT Press, Cambridge, Massachusetts, [document électronique], <https://b-ok.africa/book/2089064/37c6f9>.

- [19] BADDOU Mohamed et Ismail AIT, **Robot Superviseur : Acquisition et Echange de Données par un Protocole de Communication Sans Fil**, Projet de Développement et de Prototypage Réel d'un système Relevant du Génie Electrique, [document en ligne], 15 juin 2016, Université Mohamed V de Rabat, [https://www.slideshare.net/baddoumohamed7/conception-et-realisation-dun-robot-de-supervision-acquisition-et-change-de-donnes-distance-via-une-liaison-sans-fil](https://www.slideshare.net/baddoumohamed7/conception-et-realisation-d-un-robot-de-supervision-acquisition-et-change-de-donnes-distance-via-une-liaison-sans-fil).
- [20] LOVINE (John), **Robots Androids And Animatrons 12 Incredibles Projects You Can Build**, Seconde édition, McGraw-Hill, [document en ligne], [https://b- ok.africa/book/488848/826249](https://b-ok.africa/book/488848/826249).

ANNEXE

Annexe A : Modèle de VO

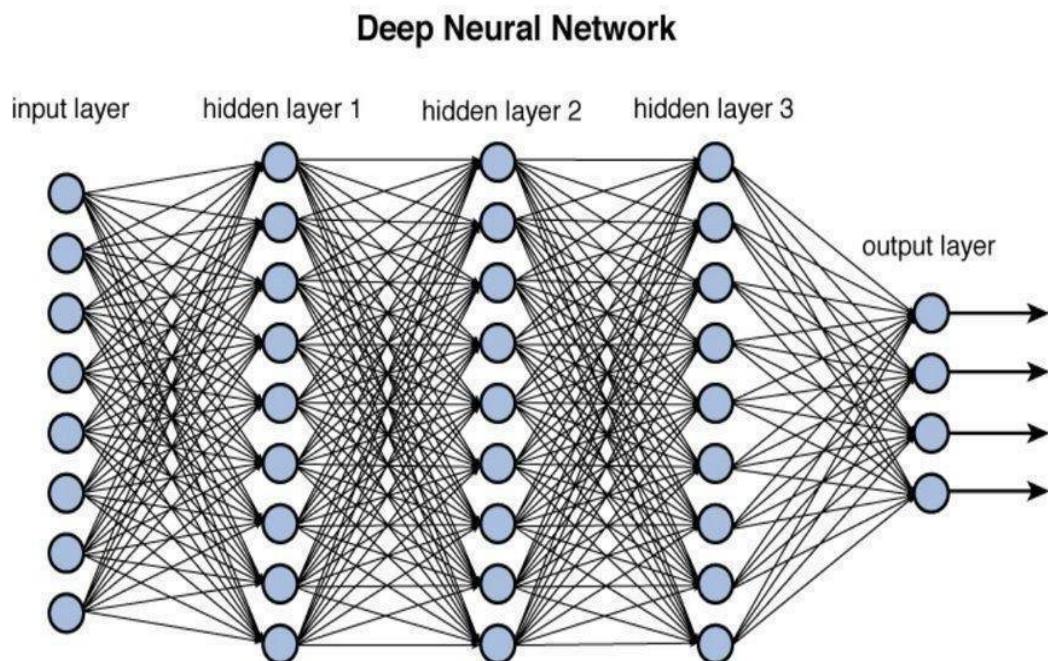


Figure 31 : Réseaux de neurones profonds

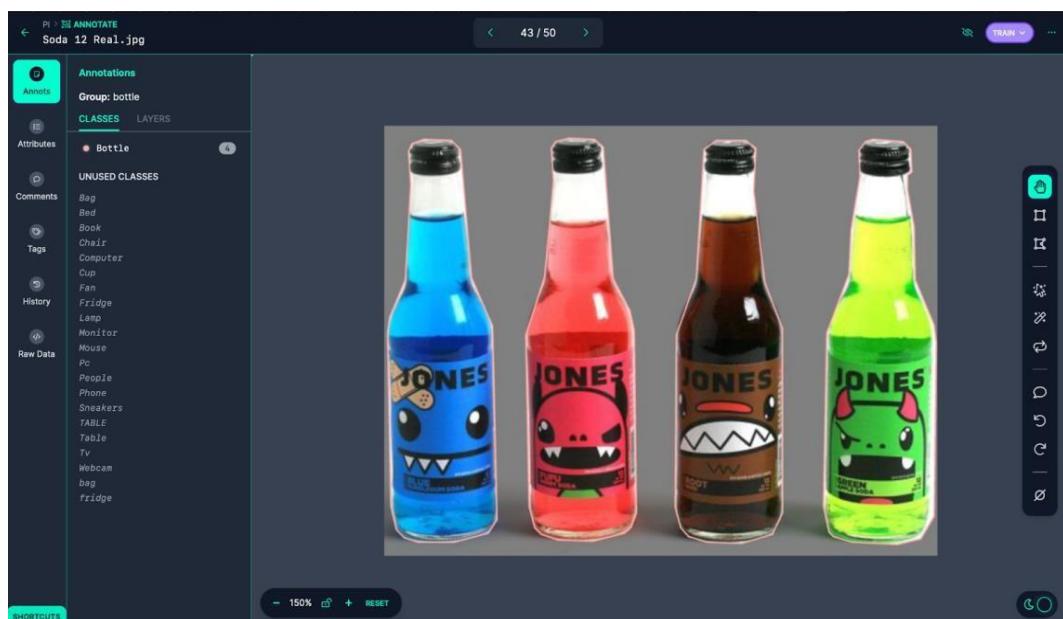


Figure 32 : Étiquetage des images avec Roboflow

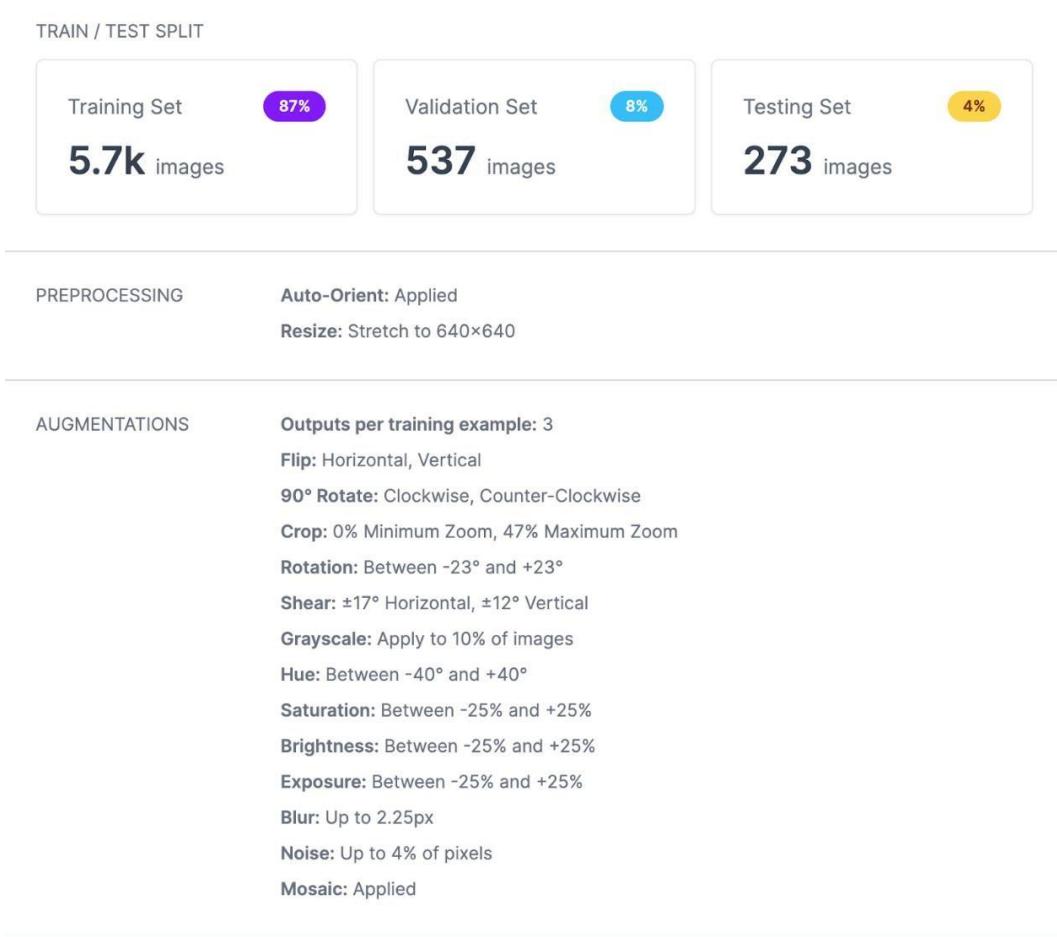


Figure 33 : Augmentation et répartition des objets avec roboflow

Annexe B : Robot

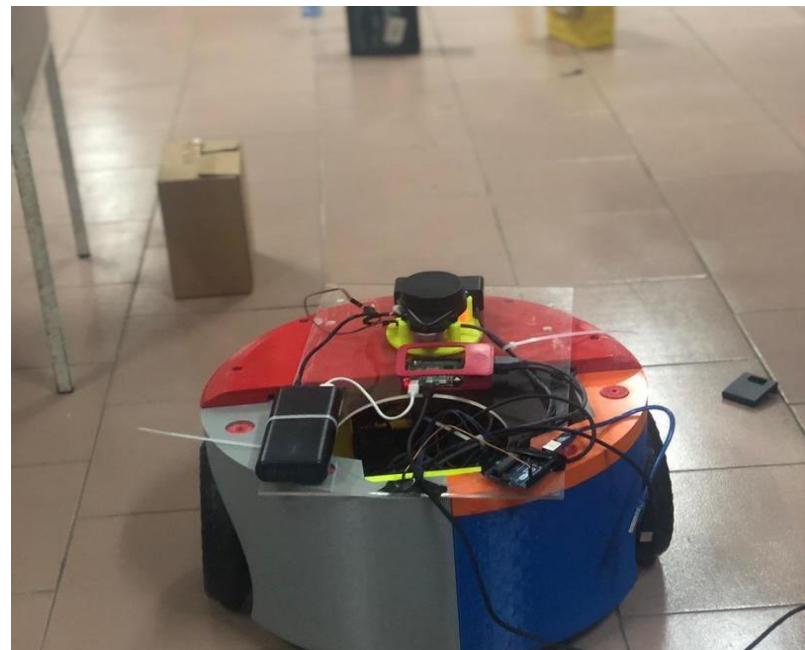


Figure 35 : Vue Avant du Robot



Figure 34 : Vue latérale du robot

Annexe B : Estimation financière

Tableau 9 : Liste des composants du robot

Désignation	Quantité	Prix Unitaire (FCFA)	Prix total (FCFA)
Carte Arduino Uno	3	21 500	64 500
Carte Arduino Méga	1	25 000	25 000
Kit Raspberry Pi	1	100 000	100 000
PCA9685	1	5 000	5 000
DRV8825	1	2 000	2 000
BTS7960	6	4 000	24 000
HC SR04	3	5 000	15 000
CJMCU	1	25 000	25 000
DHT 22	1	3 300	3 300
LED	3	50	150

Buzzer	1	75	75
Ecran nextion tft 7 pouces	1	68 000	68 000
Caméra	1	85 000	85 000
NRF24L01	3	3 000	9 000
HC-05	1	3 500	3 500
Shield Ethernet	1	5 600	5 600
Motoréducteurs	6	23 000	138 000
Servo moteur	6	5 000	30 000
Moteur pas à pas	1	10 000	10 000
Plaque perforée	2	2 000	4 000
Tige filetée	1	5 000	5 000
Roulement à bille	9	1000	9000
Contreplaqué	1	5 000	3 000

Lot de vices	2	300	600
Lot de boulons et écrous	5	500	2500
PLA	4	25 000	64 000
Bombe de peinture	1	1 500	1 500
Convertisseur de suralimentation cc 9A 300W 150W	1	1000	1000
Codeur magnétique Double Hall 5V 16PPR	1	4000	4000
Tube en aluminium	2	2 000	4 000
		Total	712 725

Tableau 10 : Tableau des machines et outils utilisés pour la réalisation

Désignation	Quantité	Prix unitaire (FCFA)	Prix totale (FCFA)
Ordinateur	1	1 500 000	1 500 000
Anet Et5	1	1 000 000	1 000 000
Perceuse vicieuse	1	36 000	36 000
Pistolet à colle	1	3 500	3 500
Lot de bâtonnet	1	2 000	2000
Kit de soudure	1	19 000	19 000
Boîte à outils	1	45 000	45 000
Lot d'attaches	1	2 000	2 000
Intel realsensor D430	1	700000	700000
		Total	2 677 000

Annexe B : Interfaces web

```
python                                Copy code

# serializers.py

from rest_framework import serializers
from .models import Alert

class AlertSerializer(serializers.ModelSerializer):
    class Meta:
        model = Alert
        fields = ['id', 'created_at', 'message', 'is_read']
```

Figure 37 : Alerte Serializer

```
python                                Copy code

# models.py

from django.db import models

class Alert(models.Model):
    created_at = models.DateTimeField(auto_now_add=True)
    message = models.CharField(max_length=255)
    is_read = models.BooleanField(default=False)
```

Figure 36 : Alerte modèle

```
import react, { useState } from 'react'

import {Routes, Route} from 'react-router-dom'
import NavBar from "../components/Nav_bar";
import { Link } from "react-router-dom"
import './Login.css'
const Login= () => {
    const [login, setLogin]=useState('')
    const [password, setPassword]=useState('')
    const [credentials, setCredentials] = useState({
        login:'',
        password:''
    })
    const onChange=(e) => {
        setCredentials({
            ...credentials,
            [e.target.name]:e.target.value
        })
    }
    const onSubmit=(e) => {
        e.preventDefault()
        console.log(credentials)
    }
}
```

Figure 38 : Login

TABLE DE MATIÈRE

DÉDICACE.....	I
RÉMERCIEMENTS	II
AVANT-PROPOS	III
SOMMAIRE	IV
LISTE DES SIGLES ET ABRÉVIATIONS	V
GLOSSAIRE.....	VI
LISTE DES TABLEAUX.....	X
INTRODUCTION.....	1
PRÉMIÈRE PARTIE : GÉNÉRALITÉS.....	3
CHAPITRE I : ETAT DE L'ART	4
I. DEFINITIONS.....	4
1. Perception d'un robot mobile.....	4
2. La navigation sémantique	4
3. Robot mobile.....	4
II. QUELQUES ROBOT DOTÉ DE LA NAVIGATION AUTONOME ET CRITIQUE	5
1. Les robots LAAS-CNRS.....	5
2. Les robots mobiles de Meanwhile.....	5
3. Le robot de surveillance de Knightscope	6
4. Critique.....	6
4.1. Les robots mobiles de Meanwhile.....	6
4.2. Les robots LAAS-CNRS.....	7
4.3. Le robot de surveillance de Knightscope	7
CHAPITRE II : ÉTUDE DE L'EXISTANT.....	8
I. PRÉSENTATION DE L'EXISTANT	8
II. LIMITES DU ROBOT ZBOT	9

CHAPITRE III : CAHIER DES CHARGES	10
I. PRESENTATION GENERALE DU PROJET	10
1. Contexte du projet	10
2. Objectifs du projet.....	10
2.1. Objectif général	10
2.2. Objectifs spécifiques	10
II. BESOINS ET CONTRAINTES DE RÉALISATION ET DIAGRAMME DE PROJET	11
1. Besoins et contraintes de réalisation	11
2. Calendrier de réalisation du projet	11
DEUXIÈME PARTIE : ÉTUDE TECHNIQUE.....	13
CHAPITRE V : MÉTHODES ET OUTILS UTILISÉS	14
I. PRÉSENTATION DES MÉTHODES DE PERCEPTION.....	14
1. Méthode de perception basée sur l'apprentissage automatique	14
2. Méthode de perception basée sur les capteurs	15
II. ETUDE COMPARATIVE DES MÉTHODES ET CHOIX DE LA MÉTHODE	16
1. Étude comparative.....	16
2. Choix de la méthode.....	18
CHAPITRES VI : ANALYSE CONCEPTUELLE.....	19
I. MODÉLISATION DU SYSTÈME.....	19
1. Présentation de la méthode de modélisation SysML	19
2. Diagramme de cas d'utilisation.....	20
3. Diagramme des exigences.....	20
4. Diagramme de séquence	21
II. CONCEPTION DU SYSTÈME DE PERCEPTION PAR LES CAPTEURS	22
1. Description	22
2. Spécification des besoins de perception.....	22
3. Sélection des capteurs pour la perception	22

4.	Conception des interfaces capteurs	23
4.1.	Conception du système de perception par le LIDAR.....	23
4.2.	Conception du système de perception par la caméra de profondeur	23
4.3.	Conception du système de perception par les encodeurs moteur	
	24	
5.	Intégration des différents composants.....	24
III.	CONCEPTION DU MODULE DE VISION PAR ORDINATEUR..	26
1.	Collecte des données	26
2.	Traitement des données.....	26
3.	Visualisation des données	27
4.	Réalisation du modèle de vision par ordinateur	27
4.1.	Choix de YOLO	28
4.2.	Algorithmes de détection d'objet.....	28
4.3.	Les Algorithmes de suivi d'objet	30
4.3.1.	Filtre de Kalman.....	31
4.3.2.	Fonctionnement du filtre de Kalman.....	31
4.4.	Mesure de la distance entre la caméra et l'objet	32
4.4.1.	La caméra stéréo	33
4.4.2.	La caméra de profondeur	34
4.4.3.	Choix de la caméra.....	35
5.	Évaluation des performances du modèle de vision par ordinateur ...	35
5.1.	Rappel	38
5.2.	Précision.....	38
5.3.	Spécificité.....	39
5.4.	Taux de réussite.....	39
5.5.	Courbes	39
IV.	SÉCURISATION DU SYSTÈME.....	41
1.	Vulnérabilités du système	41

2. Sécurisation de la connexion par SSH	42
2.1. Spécifier une seule adresse MAC à se connecter au robot.....	42
2.2. Authentification par SSH	42
3. Sécurisation des fichiers de configuration	43
4. Sécurisation de l'interface Web	43
TROISIÈME PARTIE : RÉSULTATS ET DISCUSSION	44
CHAPITRES VIII : ENVIRONNEMENT DE DÉVELOPPEMENT	45
I. ENVIRONNEMENT MATÉRIEL.....	45
1. Machine utilisée	45
2. Pièce de la structure	46
3. Composants électronique et mécanique	47
II. ENVIRONNEMENT LOGICIELS.....	48
1. Système d'exploitation.....	48
2. Langages de programmation	49
CHAPITRE IX : PRÉSENTATION DES RÉSULTATS	51
I. PRÉSENTATION DU MODÈLE DE VISION PAR ORDINATEUR 51	
1. Détection des objets	51
2. Suivi des objets	52
II. PRÉSENTATION DU ROBOT.....	53
1. Présentation mécanique.....	53
2. Présentation électronique	54
III. PRÉSENTATION DU LOGICIEL.....	55
1. Logiciel du robot.....	55
2. Application web de commande.....	56
CHAPITRE X : DISCUSSION ET ESTIMATION FINANCIÈRE.....	58
I. DISCUSSION DES RÉSULTATS	58
1. Modèle de vision par ordinateur	58
2. Robot.....	58
3. Interface Web	59
II. ESTIMATION FINANCIÈRE.....	59

CONCLUSION	61
BIBLIOGRAPHIE	XI
WEBOGRAPHIE.....	XII
ANNEXE	XIV
ANNEXE A : MODELE DE VO	XIV
ANNEXE B : ROBOT.....	XVI
ANNEXE B : ESTIMATION FINANCIERE	XVII
ANNEXE B : INTERFACAE WEB	XXI

RÉSUMÉ

L'assistance des personnes à capacité réduite est un enjeu important de notre société car il concerne une partie de la population qui peut avoir des difficultés à effectuer certaines activités de la vie quotidienne en raison d'un handicap, d'une maladie ou du vieillissement. Pour leur porter assistance, il existe une multitude d'organisations et de services, mais aussi le développement d'outils technologiques en particulier la robotique. Ce présent mémoire résulte d'un projet d'assistance aux personnes à capacité réduite par la robotique mobile. Pour concrétiser l'utilité de notre robot nous avons accentué notre travail sur la conception d'un module de perception afin de permettre une navigation autonome sécurisée du robot. Dans le souci d'effectuer un travail rigoureux et conforme au cahier des charges, nous nous sommes familiarisés avec les différents travaux déjà réalisés sur le sujet. De plus, une étude technique a été menée pour déterminer les méthodes et outils adéquats pour la conception du module de perception. Nous avons terminé la réalisation de notre solution avec quelques tests de fonctionnement avant le déploiement.

ABSTRACT

Assisting people with reduced capacity is an important issue in our society, as it concerns a part of the population that may have difficulties in carrying out certain activities of daily life due to a disability, an illness or aging. To assist them, there are a multitude of organizations and services, but also the development of technological tools, in particular robotics. This thesis is the result of a project to assist people with reduced capacity using mobile robotics. To concretise the usefulness of our robot, we have focused our work on the design of a perception module to allow safe autonomous navigation of the robot. In order to carry out rigorous work and in conformity with the specifications, we familiarized ourselves with the various works already carried out on the subject. In addition, a technical study was conducted to determine the methods and tools suitable for the design of the perception module. We finished the realization of our solution with some functional tests before the deployment.