Formation ISIE

Canne Blanche Intelligente

Rédaction

Rév	Date	Emetteur	E-mail
1	2024-02-29	G. LUSAKUMUNU DIMBU	dimbugrace@gmail.com

Table 1 – Revision history

Validation

Rév	Date	Emetteur	E-mail
1		G.Venturini	gilles.venturini@univ-tours.fr

Table 2 – Revision history

Révision

Rév	Date	Emetteur	E-mail
1			

Table 3 – Revision history

Table des matières

Remerciement

Tout d'abord, je tiens à exprimer ma profonde gratitude à mon encadrant et client, **M. Gilles VENTURINI**, pour son soutien et ses conseils indéfectibles tout au long de ma recherche ou projet de fin d'étude. Son expertise et ses connaissances ont été inestimables pour moi dans la réalisation de ce projet.

Je tiens également à remercier **Polytech Tours** et le département **ISIE** pour m'avoir donné l'opportunité de réaliser ce projet et pour leur soutien financier.

Enfin, je tiens à remercier ma famille et mes amis pour leur amour et leur encouragement tout au long de mes études. Leur soutien a été une source constante de motivation et d'inspiration.

Chapitre 1

Introduction

L'accès à l'information et à la mobilité sont des droits fondamentaux de tout être humain, quelles que soient ses capacités. Pourtant, les personnes atteintes de cécité ou de malvoyance font face à de nombreux défis dans leur vie quotidienne en raison d'un environnement souvent peu adapté à leurs besoins spécifiques. C'est dans ce contexte que s'inscrit le projet de fin d'études présenté dans ce rapport : une canne intelligente conçue pour améliorer l'autonomie et la sécurité des personnes aveugles dans leurs déplacements.

Ce rapport décrit en détail les différentes étapes de la conception, du développement et des tests de cette canne intelligente, qui vise à offrir une alternative innovante et performante aux cannes traditionnelles. Grâce à des capteurs avancés et à une intelligence artificielle intégrée, cette canne est capable de détecter les obstacles et de guider l'utilisateur de manière précise et fiable, lui permettant ainsi de se déplacer plus facilement et en toute confiance dans des environnements variés.

Au-delà de sa contribution technique à l'amélioration de la mobilité des personnes aveugles, ce projet s'inscrit également dans une démarche plus large de conception inclusive et accessible, qui vise à intégrer les besoins spécifiques des personnes en situation de handicap dès la conception des produits et des services. En ce sens, il constitue un exemple concret de l'importance de l'accessibilité dans la société actuelle et de la nécessité de promouvoir des solutions innovantes et adaptées aux besoins de tous les citoyens.

Chapitre 2

Presentation

2.1 Contexte

Ce projet est en continuité avec une idée et un travail commencé les années précédentes : pour aider à la navigation des personnes malvoyantes, un dispositif de reconnaissance d'obstacles peut être placé sur une canne blanche. Le système embarqué dispose d'une caméra (avec reconnaissance d'images via réseaux de neurones convolutionnels) et d'un capteur ultrasonique. Le principe est de reconnaitre des objets et de calculer la distance à laquelle ils se trouvent. Il faut continuer cette étude, et réfléchir aux méliorations possibles selon deux axes :
Améliorer le "feedback" fait à la personne (synthèse vocale), remplacer le CPU actuel par un CPU plus efficace pour la reconnaissance d'images, de manière à réaliser au moins 4 acquisitions par seconde (actuellement c'est plus proche de 1 acquisition en 4 s).

2.2 Objectifs

L'objectif principal de mon projet est de développer une canne intelligente améliorée pour les personnes aveugles, en mettant l'accent sur l'amélioration de la reconnaissance d'obstacles et de la synthèse vocale. Pour cela, on peut décomposer mes objectifs en deux sous-objectifs :

- Améliorer la reconnaissance d'obstacles et la calcul de la distance à l'aide de capteurs et d'un système embarqué plus performants.
- Améliorer le "feedback" fait à la personne en utilisant une synthèse vocale plus précise et en optimisant la vitesse de reconnaissance d'images.

Pour atteindre ces sous-objectifs, nous avons mis en place les étapes suivantes :

- Évaluer les performances actuelles du système embarqué et des capteurs en termes de vitesse de reconnaissance d'images et de calcul de distance.
- Rechercher et sélectionner des capteurs et des systèmes embarqués plus performants pour améliorer la reconnaissance d'obstacles et le calcul de la distance.

— Optimiser la vitesse de reconnaissance d'images et la précision de la synthèse vocale en ajustant les paramètres de du réseau de neurones convolutionnels.

2.3 Présentation de l'existant

Compte tenu de la reprise du projet et de la nécessité de recentrer les fonctionnalités de la canne intelligente pour les aveugles, nous avons décidé de ne conserver que le capteur de distance comme composant matériel clé. Cette décision a été prise en raison du non-respect du cahier de charges initial et de la nécessité de simplifier le design pour atteindre un produit fonctionnel dans les délais impartis. Le capteur de distance sera utilisé pour détecter les obstacles sur la trajectoire de la canne et avertir l'utilisateur par des signaux sonores ou tactiles. Le choix de ce capteur s'appuie sur sa fiabilité, sa précision et sa facilité d'intégration dans le design global de la canne. Nous sommes conscients que cette décision entraînera la suppression de certaines fonctionnalités prévues initialement, telles que la reconnaissance d'images ou la synthèse vocale. Néanmoins, nous croyons que le capteur de distance est la fonctionnalité la plus critique pour assurer la sécurité et l'autonomie des personnes malvoyantes.

En conséquence, nous allons réviser le cahier de charges en conséquence et axer notre développement sur l'amélioration de la précision et de la fiabilité du capteur de distance. Nous allons également travailler sur l'interface utilisateur pour fournir des feedbacks clairs et intuitifs à l'utilisateur. Nous sommes convaincus que cette approche simplifiée permettra de livrer un produit fonctionnel et utile pour les personnes malvoyantes, tout en respectant les contraintes de coût, de délai et de qualité.

Chapitre 3

Stratégie d'approche et éléments de départ

3.1 Motivations

Une motivation majeure pour le choix de ce projet de canne intelligente pour les aveugles est l'opportunité d'avoir un impact social positif en améliorant la qualité de vie des personnes malvoyantes. Les obstacles et les dangers dans l'environnement peuvent représenter des risques importants pour les personnes aveugles, limitant leur autonomie et leur mobilité.

En développant une canne intelligente avec une reconnaissance d'obstacles améliorée, une synthèse vocale et une Mesure de la distance à l'obstacle, ce projet offre la possibilité de résoudre certains de ces défis. Cela permettra non seulement aux personnes aveugles de se déplacer plus facilement.

En outre, ce projet s'inscrit dans une démarche de conception inclusive et accessible, qui vise à intégrer les besoins des personnes en situation de handicap dès la conception des produits et des services. Grâce à cette approche, les personnes aveugles peuvent bénéficier d'une technologie adaptée à leurs besoins, ce qui peut améliorer leur participation et leur inclusion dans la société.

En somme, le choix de ce projet est motivé par la volonté de contribuer à une société plus inclusive, accessible et équitable, en offrant une solution innovante et utile pour les personnes aveugles

3.2 Stratégie d'approche

Pour aborder ce projet de canne intelligente pour les aveugles, une stratégie d'approche efficace consiste à adopter une démarche itérative et incrémentale. Cela implique de commencer par une version minimale viable du produit, qui peut être testée auprès d'un petit groupe d'utilisateurs cibles. Les commentaires et les retours de ces utilisateurs peuvent ensuite être utilisés pour améliorer le produit et ajouter de nouvelles fonctionnalités.

Voici les étapes clés de cette stratégie d'approche :

1. Identifier les besoins et les exigences des utilisateurs cibles (personnes aveugles) en matière de navigation.
2. Concevoir une version minimale viable de la canne intelligente, qui comprend les fonctionnalités de base telles que la reconnaissance d'obstacles et la synthèse vocale.
3. Tester la version minimale viable auprès d'un petit groupe d'utilisateurs cibles et recueillir leurs commentaires et retours. Analyser les commentaires et les retours pour identifier les points forts et les points faibles du produit, ainsi que les opportunités d'amélioration.
4. Améliorer la canne intelligente en ajoutant de nouvelles fonctionnalités, telles que la géolocalisation ou autres modalités
5. Tester à nouveau la canne intelligente auprès d'un groupe plus large d'utilisateurs cibles et recueillir leurs commentaires et retours.

Répéter les étapes 4 à 5 jusqu'à ce que le produit réponde pleinement aux besoins et aux exigences des utilisateurs cibles. Cette stratégie d'approche permet non seulement de s'assurer que le produit répond aux besoins des utilisateurs cibles, mais également de réduire les risques de développement en testant le produit à chaque étape du processus. Cela peut également permettre de réduire les coûts de développement en identifiant et en résolvant les problèmes dès le début du processus.

3.3 Éléments de départ

3.3.1 Cahier de spécification

Le cahier de spécification est un document essentiel dans le développement d'un projet, qui définit les exigences fonctionnelles et techniques du produit à développer. Voici donc le cahier de spécification pour la canne intelligente pour les aveugles :

1. **Exigences fonctionnelles**
   Reconnaissance d'obstacles :
   Le produit doit être capable de détecter les obstacles à proximité de l'utilisateur et de les signaler via une synthèse vocale.
2. **Exigences techniques**
   Capteurs : Le produit doit être équipé de capteurs de détection d'obstacles, tels que des caméras et des capteurs ultrasons.

   Système embarqué : Le produit doit être équipé d'un système embarqué capable de traiter les données des capteurs et de fournir une synthèse vocale.
3. **Spécifications techniques détaillées**
   Capteurs :

— Caméra :
Résolution ou champ de vision de 200 degrés.

— Capteurs ultrasons :
Portée minimale de 1 mètre, précision de détection de 5 cm.
Système embarqué :
Processeur : Processeur ARM Cortex-A7 ou équivalent.
Mémoire : Mémoire vive de 512 Mo, mémoire flash de 4 Go.
Synthèse vocale : Synthèse vocale texte-à-parole, avec une voix féminine ou masculine au choix.
Autonomie :

— Batterie :
Batterie lithium-ion rechargeable, capacité minimale de 2000 mAh. Temps de charge : Moins de 4 heures. Autonomie en utilisation : Au moins 8 heures.

Ce cahier de spécification définit les exigences fonctionnelles et techniques pour le développement d'une canne intelligente pour les aveugles. Le produit doit offrir une reconnaissance d'obstacles

3.3.2 Le cahier d'analyse

I. Analyse des besoins
Identification des besoins des utilisateurs :
L'analyse des besoins a identifié les principales fonctionnalités requises pour la canne intelligente, telles que la détection des obstacles, la géolocalisation et la surveillance des signes vitaux. **Par contre dans ce projet nous visons uniquement la détection d'abjects, la mesure de la distance et le feedback à l'utilisateur.**

II. Analyse des risques
Risques techniques : Les risques techniques identifiés comprennent les problèmes de détection des obstacles, les problèmes de connectivité et les problèmes de batterie.

Risques de sécurité :
Les risques de sécurité identifiés comprennent les risques de piratage, les risques de fuite de données et les risques de divulgation d'informations personnelles. **A savoir ce risque est identifié que si notre système se connecte sur interne concernant l'approche actuelle nous sommes hors de danger**

III. Analyse des coûts
Coûts de développement :
Les coûts de développement comprennent les coûts de conception et les coûts de tests.

Coûts matériels :
Les coûts matériels comprennent les coûts des composants électroniques, les coûts des capteurs et les coûts de la batterie.

Le cahier d'analyse a permis d'identifier les besoins des utilisateurs, les risques potentiels et les coûts associés au développement de la canne intelligente pour les aveugles. Sur la base de cette analyse, une stratégie de développement a été élaborée pour garantir la réussite du projet.

Chapitre 4

Choix dans le projet

4.1 Choix organisationnels

Pour la réalisation de notre projet canne intelligente pour aveugle, nous avons fait des choix organisationnels qui nous permettrons de respecter les jalons imposés par Polytech Tours pour les livrables de spécifications et d'analyse. Ces choix concernent la structure et le fonctionnement du projet, la répartition des tâches, la planification et le suivi des activités.

Un outil utile pour gérer ces choix est le diagramme de Gantt établi. Il nous permet de représenter visuellement l'ensemble des tâches à réaliser, leur durée, leur dépendance et leur chevauchement.
En outre, ça me permet d'assurer que les livrables de spécification et d'analyse sont de qualité et répondent aux attentes de Polytech Tours.

Enfin, j'ai prévu des mécanismes de communication et de coordination pour ce projet. Cela me permettra de gérer les risques, de résoudre les problèmes et de prendre des décisions éclairées en temps utile.

4.1.1 Cycle de développement

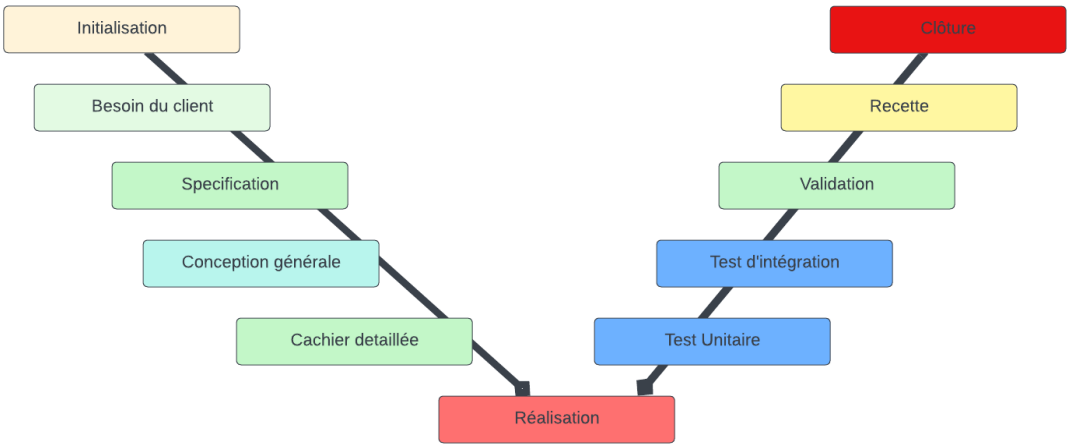


FIGURE 4.1 – Cycle en V

4.1.2 Définitions et validation des spécifications fonctionnelles

Ce point marque la traduction technique des besoins exprimés par les parties prenantes. Les attentes sont converties en fonctionnalités ou modules fonctionnels du produit attendu.

4.1.3 Définition et validation des spécifications techniques

C'est la suite de la traduction de l'expression des besoins en caractéristiques techniques. Ce point marque un niveau important dans la définition de la conformité du résultat attendu.

4.1.4 Définition et validation des plans et/ou des architectures

C'est la phase de conception effective du produit : le modèle, les dimensions, les détails techniques sont définis. Définition et validation du cahier des tests.
Le cahier des tests contient tous les détails relatifs à la canne.

4.1.5 Validation de la conformité

Ce troisième point de la dernière étape du cycle V s'appuie sur les résultats des tests unitaires et d'intégration, pour valider la conformité selon les attentes initialement exprimées. C'est pourquoi le cahier des tests reste un élément important à définir dans la première étape du projet.

4.1.6 Recette

C'est un formalisme administratif qui consiste à transmettre officiellement le résultat du projet au « client ». Selon la formule de mise en production, la recette

peut être découpée en deux parties : une recette provisoire et une recette définitive.

Après avoir identifié et évalué les risques potentiels associés à notre projet de canne intelligente, nous allons maintenant présenter notre planification détaillée en utilisant un diagramme de Gantt.
Ce diagramme nous permettra de visualiser les tâches à accomplir, leur durée et leur dépendance les unes par rapport aux autres, ce qui nous aidera à coordonner notre travail et à respecter les échéances.
Le cycle en V se caractérise par une phase de spécification détaillée en amont, suivie d'une phase de développement itératif et incrémental, et enfin d'une phase de validation et de vérification en aval.

Notre diagramme de Gantt reflète cette structure en trois phases. La première phase, intitulée **"Spécification"**, comprend les tâches de définition des exigences, de conception de l'architecture et de planification détaillée du projet. La deuxième phase, intitulée **"Développement"** , comprend les tâches de développement itératif et incrémental, de tests unitaires et d'intégration continue. Enfin, la troisième phase, intitulée **"Validation"**, comprend les tâches de tests d'acceptation, de déploiement et de maintenance du logiciel.

Nous sommes convaincus que notre planification détaillée et notre utilisation du cycle en V nous permettront de livrer un produit de haute qualité dans les délais impartis, et de répondre à toutes les questions ou préoccupations qu'ils peuvent avoir.

Sur ce, nous avons établi notre suivi comme suite :

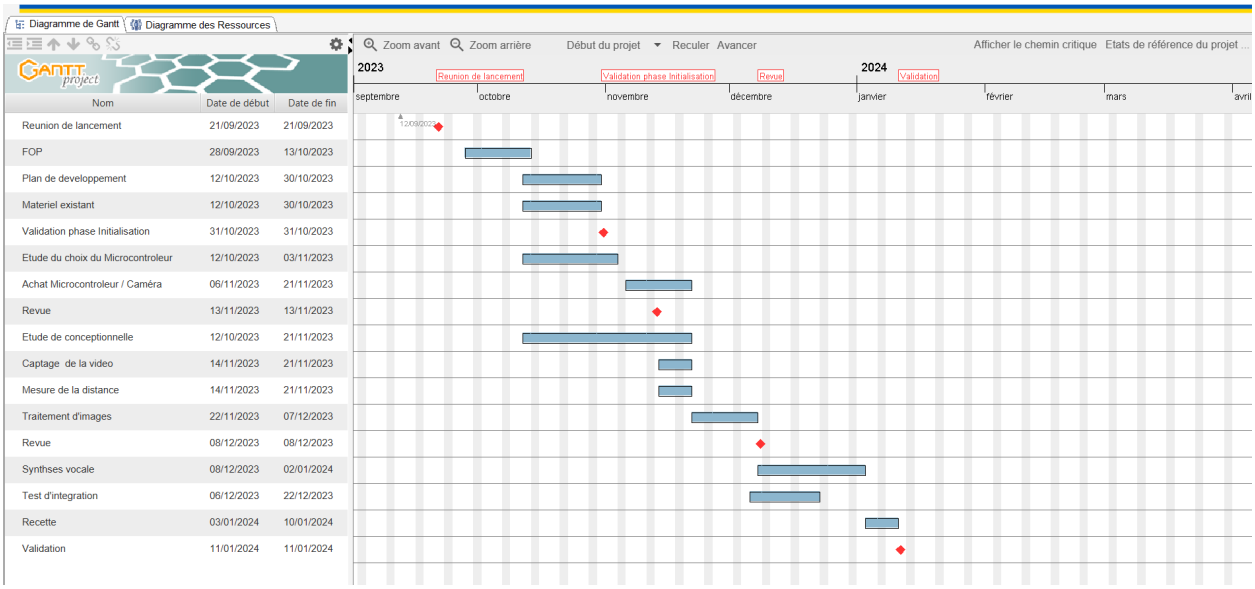


FIGURE 4.2 – Gantt

4.2 Tableau des risques

Un tableau de risque est un outil de gestion de projet qui permet d'identifier, d'évaluer et de suivre les risques potentiels associés à un projet. Voici un exemple de tableau de risque pour notre projet de canne intelligente :

Risque	Probabilité	Impact	Mitigation Strategies
Défaillance Matérielle	Moyenne	Elévé	Sélectionner des composants de haute qualité et fiables - Planifier des tests de contrôle qualité réguliers
Problèmes Bliothèques	Élevée	Elévé	- Utiliser des langages de programmation - des frameworks approuvés - Planifier des tests de validation et de vérification rigoureux
Coûts dépassant le budget	Moyenne	Elévé	- Établir un budget réaliste et détaillé - Prévoir une marge de manœuvre pour les imprévus
Délais dépassant l'échéance	Moyenne	Elévé	- Établir un échéancier réaliste et détaillé - Prévoir une marge de manœuvre pour les imprévus
Choix Matériel	Moyenne	Elévé	- Établir un échéancier réaliste et détaillé - Prévoir une marge de manœuvre pour les imprévus
Analyse d'images	Moyenne	Elévé	Prévoir une marge de de temps pour résoudre le problème
Problème d'intégration	Moyenne	Elévé	- Prévoir une marge de de temps pour résoudre le problème

TABLE 4.1 – Tableau des risques

Le tableau de risque peut être mis à jour régulièrement tout au long du projet pour refléter les changements dans les risques identifiés, leur probabilité et leur impact, ainsi que les stratégies de mitigation mises en place.

4.3 L'équipe de projet

Ingenieur qualité	Grâce LUSAKUMUNU DIMBU
Test d'intégration	Grâce LUSAKUMUNU DIMBU
Client	M. Gilles Venturini

TABLE 4.2 – Revision history

4.4 Choix techniques

4.4.1 Choix MCU

Pour cette approche, nous avons choisi d'utiliser une NVIDIA Jetson Nano pour la puissance de traitement. La Jetson Nano est une petite carte informatique conçue pour les applications d'intelligence artificielle et de deep learning. Elle est équipée d'un processeur quad-core ARM A57 et d'un GPU NVIDIA Maxwell avec 128 cœurs CUDA.

Avant de décrire les spécifications techniques de la Jetson Nano, il est important de noter que la carte doit être alimentée par un bloc d'alimentation externe

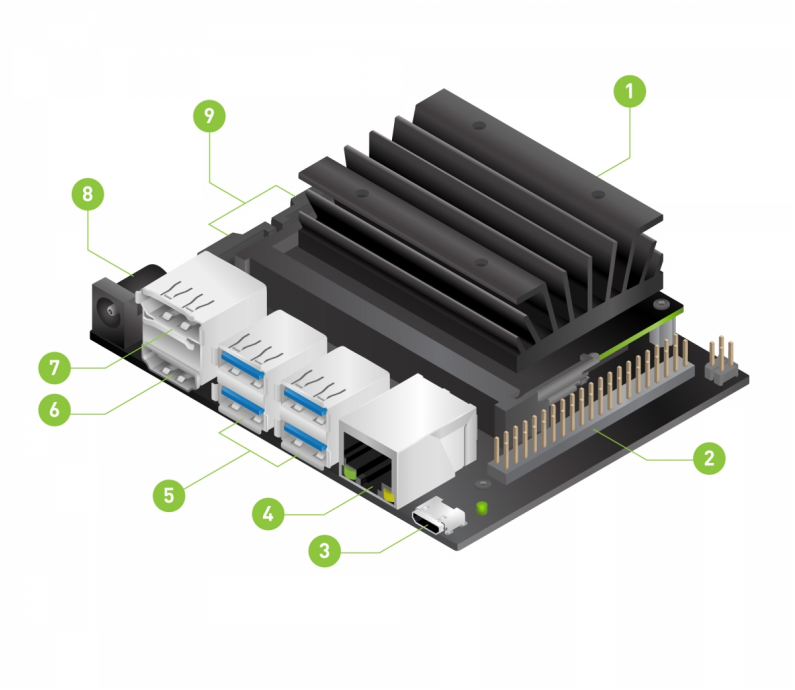


FIGURE 4.3 – jetson Kit

capable de fournir une tension de 5 V et un courant de 2 A. La carte utilise une carte microSD comme support de stockage principal et il est recommandé d'utiliser une carte d'au moins 32 Go pour pouvoir installer le système d'exploitation et les logiciels nécessaires.

La Jetson Nano est disponible en deux versions : la Developer Kit et la Module Kit. La Developer Kit est une carte autonome avec des ports HDMI, USB, Ethernet, et un slot pour carte microSD. La Module Kit, quant à elle, est une version plus compacte de la carte, conçue pour être intégrée dans des systèmes embarqués.

Dans les sections suivantes, nous décrirons les caractéristiques techniques de la Jetson Nano et expliquerons comment installer et configurer l'environnement de développement pour notre projet.

Performance :
Le Jetson Nano dispose d'un GPU Maxwell à 128 cœurs et d'un processeur quad-core ARM Cortex-A57, ce qui le rend beaucoup plus puissant qu'une Raspberry Pi ou un BeagleBone. Il peut gérer des algorithmes d'apprentissage machine plus complexes et des jeux de données plus volumineux.

Accélération GPU :
Le GPU du Jetson Nano est spécialement conçu pour les applications d'apprentissage machine, offrant jusqu'à 472 GFLOPS de performance de calcul. Cela permet des temps de formation et d'inférence plus rapides, ce qui est essentiel pour les applications en temps réel.
Connectivité : Le Jetson Nano dispose de plusieurs options de connectivité, notam-

ment le Gigabit Ethernet, l'USB 3.0 et le M.2 Key E pour les modules sans fil. Cela permet un transfert de données rapide et une communication avec d'autres appareils.

Efficacité énergétique :
Malgré sa plus grande puissance de traitement, le Jetson Nano reste économe en énergie, ne consommant que 5 à 10 watts d'énergie. Cela le rend adapté aux applications alimentées par batterie.
Support logiciel : NVIDIA propose un ensemble d'outils logiciels complets pour le Jetson Nano, notamment le SDK JetPack, TensorRT et CUDA. Ces outils facilitent le développement et le déploiement d'applications d'apprentissage machine sur le Jetson Nano.

Dans l'ensemble, le Jetson Nano est un choix excellent pour les applications d'apprentissage machine qui nécessitent une puissance de traitement supérieure à celle d'une Raspberry Pi ou d'un BeagleBone. Sa accélération GPU, ses options de connectivité, son efficacité énergétique et son support logiciel en font une plateforme idéale pour le développement d'appareils intelligents tels que votre canne intelligente pour les aveugles.

4.4.2 Choix caméra

FIGURE 4.4 – Caméra

Nous avons choisi la caméra IMX219-200,la caméra IMX219-200 est une caméra de 80 mégapixels conçue pour offrir une qualité d'image supérieure dans une variété d'applications, telles que la vision industrielle, la robotique et la réalité augmentée.

Voici quelques-uns des avantages de la caméra IMX219-200 : Haute résolution : La caméra offre une résolution de 80 mégapixels, ce qui permet de capturer des images haute résolution avec une grande précision et une grande clarté.

Grand angle de vue : La caméra offre un angle de vue de 200 degrés, ce qui permet de capturer une grande zone d'image.

Faible bruit : La caméra est équipée d'un capteur d'image haute sensibilité et d'un processeur ISP (Image Signal Processor) intégré qui permettent de réduire le bruit de l'image, même dans des conditions de faible luminosité.

Compatibilité : La caméra est compatible avec une variété de plates-formes de calcul, telles que Jetson Nano, Raspberry Pi, et d'autres plates-formes basées sur Linux. Flexibilité : La caméra offre une grande flexibilité en termes de taille d'image, de fréquence d'images et de formats d'image, ce qui permet de l'adapter à une variété d'applications.

Facilité d'utilisation : La caméra est facile à installer et à utiliser, grâce à son interface plug-and-play et à sa documentation complète. Dans l'ensemble, la caméra IMX219-200 offre une qualité d'image supérieure et une grande flexibilité pour une variété d'applications.

4.4.3 Capteur de distance

FIGURE 4.5 – Capteur de distance

Le capteur de distance A02YYUW est un capteur de distance à ultrasons développé par Microdrones. Il s'agit d'un capteur compact et léger qui offre une précision de mesure de distance élevée et une grande fiabilité.

Voici quelques-unes des caractéristiques :

Précision élevée : Le capteur est capable de mesurer les distances avec une précision de ±3 mm, ce qui en fait un choix idéal pour les applications qui nécessitent une grande précision. Grande portée : Le capteur peut mesurer les distances jusqu'à 5 mètres.

Interface numérique : Le capteur utilise une interface numérique I²C pour communiquer avec le microcontrôleur hôte, ce qui permet une intégration facile dans une variété de systèmes.

Fonctionnement à haute vitesse : Le capteur peut fonctionner à des fréquences d'échantillonnage élevées, ce qui permet de mesurer les distances rapidement et de manière fiable.

Faible consommation d'énergie : Le capteur a une faible consommation d'énergie, ce qui permet de l'utiliser dans des applications où l'alimentation est limitée.

Compact et léger : Le capteur est compact et léger, ce qui permet de l'intégrer facilement dans une variété de systèmes et de produits. Dans l'ensemble, le capteur de distance A02YYUW offre une précision élevée, une grande portée et une interface numérique facile à utiliser, ce qui en fait un choix idéal pour une variété d'applications, telles que la robotique, la logistique et l'automatisation industrielle.

pour communiquer avec le microcontrôleur hôte, et non I²C. L'interface UART permet au capteur de transmettre et de recevoir des données sous forme de caractères, ce qui le rend facile à intégrer dans une variété de systèmes. Le capteur utilise une baud rate de 115200 bps et prend en charge les protocoles de communication UART standard tels que TTL et RS-232.

En plus de sa précision élevée et de sa grande portée, le capteur de distance A02YYUW offre une faible consommation d'énergie, une faible latence et une grande fiabilité, ce qui en fait un choix idéal pour les applications où la précision et la rapidité sont essentielles.

4.4.4 Intelligence du système

L'intelligence du système, également appelée intelligence artificielle (IA), fait référence à la capacité d'un système à effectuer des tâches qui nécessiteraient normalement l'intelligence humaine, telles que la perception, la raisonnement, l'apprentissage, la planification et la prise de décision.

Les systèmes intelligents sont conçus pour être capable d'apprendre, de s'adapter et de s'améliorer en fonction des données et des expériences qu'ils collectent. Ils utilisent des algorithmes d'apprentissage automatique, de la vision par ordinateur, du traitement du langage naturel et d'autres techniques pour traiter et analyser des données complexes et prendre des décisions éclairées en conséquence.

En résumé, l'intelligence du système est une forme avancée de traitement de l'information qui permet aux systèmes informatiques de simuler l'intelligence humaine et de prendre des décisions éclairées en fonction des données et des expériences qu'ils

collectent.

4.4.5 Réseaux des neurones convolutionnels

Les réseaux de neurones convolutionnels (CNN) sont une forme de réseau de neurones artificiels (ANN) qui sont particulièrement bien adaptés au traitement de données multidimensionnelles telles que les images et les vidéos. Les CNN sont conçus pour imiter le fonctionnement du cerveau humain en traitant les données en plusieurs étapes, en utilisant des couches de neurones connectées en cascade.

Le principe de base des CNN est la convolution, qui est une opération mathématique qui permet de détecter les caractéristiques spatiales d'une image. Les CNN utilisent des filtres (également appelés noyaux) pour appliquer la convolution aux pixels de l'image et détecter les caractéristiques telles que les bords, les textures et les formes.

Les CNN sont composés de plusieurs couches, chacune ayant une fonction spécifique. Les couches d'entrée reçoivent les données d'entrée, telles que les images. Les couches de convolution appliquent la convolution aux données d'entrée pour détecter les caractéristiques. Les couches de pooling réduisent la taille des données en regroupant les valeurs voisines. Les couches entièrement connectées (également appelées couches de neurones) traitent les données pour produire les résultats finaux.

Les CNN sont couramment utilisés pour des tâches telles que la reconnaissance d'objets, la détection de visages, la classification d'images et la segmentation d'images. Ils peuvent être entraînés à partir de données d'entraînement pour apprendre à détecter les caractéristiques spécifiques et améliorer leur précision au fil du temps.

En résumé, les réseaux de neurones convolutionnels sont une forme de réseau de neurones artificiels qui sont particulièrement bien adaptés au traitement de données multidimensionnelles telles que les images et les vidéos. Ils utilisent des couches de convolution, de pooling et entièrement connectées pour détecter les caractéristiques spatiales et produire des résultats précis. Les CNN sont largement utilisés dans des applications telles que la reconnaissance d'objets, la détection de visages et la classification d'images.

4.4.6 Nvidia Learning

La carte NVIDIA Jetson Nano offre une plateforme idéale pour le développement de DNN grâce à sa puissante architecture de traitement GPU et son processeur quad-core ARM Cortex-A57. La carte Jetson Nano est capable de traiter des charges de travail de deep learning complexes et de fournir des performances élevées pour les applications de vision par ordinateur et d'intelligence artificielle.

Pour développer des DNN sur la carte Jetson Nano, vous pouvez utiliser des frameworks populaires de deep learning tels que TensorFlow, PyTorch et Caffe. NVIDIA fournit également des outils et des bibliothèques de développement pour aider les développeurs à accélérer le développement et le déploiement de DNN sur Jetson Nano.

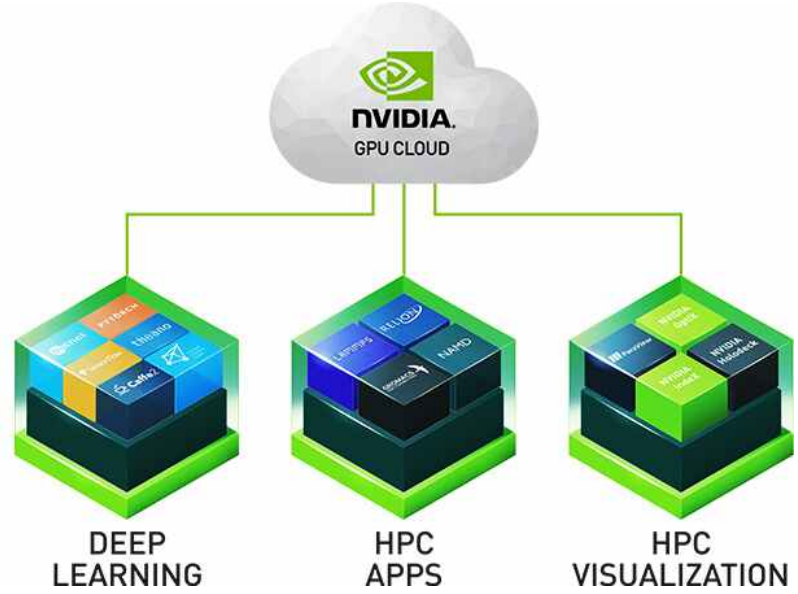


FIGURE 4.6 – Nvidia learning

Les DNN sont gourmands en calcul et nécessitent de la puissance de traitement importante. C'est pourquoi ils sont parfaitement adaptés aux systèmes embarqués comme Jetson Nano, qui dispose d'un processeur NVIDIA powerful et d'une grande quantité de mémoire vive. L'entraînement d'un DNN peut être chronophage, nécessitant de la patience et de la persévérance. Toutefois, les performances obtenues peuvent être très satisfaisantes, en particulier dans les domaines de l'intelligence artificielle.

Il est important de garder à l'esprit que l'entraînement d'un DNN nécessite une grande quantité de données. Pour optimiser les performances d'un DNN, il est donc conseillé de disposer d'un ensemble de données adéquat et étoffé. Pour réduire les temps d'entraînement et les coûts de calcul, il est possible d'utiliser des techniques d'optimisation, comme le transfert d'apprentissage, qui consiste à réutiliser les poids d'un modèle pré-entraîné pour initier un autre modèle. En résumé, les DNN offrent de nombreuses possibilités pour créer des systèmes intelligents et puissants, tels que des robots, des voitures autonomes ou des applications mobiles interactives. Grâce à l'adaptabilité de Jetson Nano et à sa capacité de traitement important, les DNN sont particulièrement adaptés à l'utilisation dans des systèmes embarqués.

4.4.7 Bibliothèque jetson inference

La bibliothèque **jetson_inference** est une bibliothèque open-source développée par NVIDIA pour la carte Jetson Nano qui permet aux développeurs de créer facilement des applications de vision par ordinateur et d'intelligence artificielle.

La bibliothèque **jetson_inference** offre une variété de fonctionnalités pour le traitement d'images et de vidéos, y compris la détection d'objets, le suivi d'objets, la classification d'images et la segmentation d'images. Elle prend en charge plusieurs frameworks de deep learning populaires tels que TensorFlow, PyTorch et Caffe, et

offre des interfaces simples pour intégrer ces frameworks dans les applications Jetson Nano.

La bibliothèque **jetson_inference** est optimisée pour les architectures GPU de NVIDIA et offre des performances élevées pour le traitement de charges de travail de vision par ordinateur et d'intelligence artificielle. Elle inclut également des outils de profilage et de débogage pour aider les développeurs à optimiser les performances de leurs applications.

4.5 Lien symbolique de la bibliothèque

Vous pouvez consulter le GitHub de NVIDIA pour en savoir plus : ici NVIDIA. Les démarches que j'ai mis en place pour le fonctionnement de ce projet sont :

```
$sudo apt-get update
$sudo apt-get install git cmake libpython3-dev python3-numpy
$git clone --recursive --depth=1 https://github.com/dusty-nv/
   jetson-inference
$cd jetson-inference
$mkdir build
$cd build
$cmake ../
$make -j$(nproc)
$sudo make install
$sudo ldconfig
```

Listing 4.1 – Resources bibliothèque

cette configuration pour utilisation de la bibliothèque **inference de Jetson nano** Mais pour cela voyons comment mettre l'image système sur jetson nano :

Chapitre 5

Réalisation

5.1 Installation du noyau

Depuis le site de Jetson nano telecharger le noyau qui va bien depuis le lien : Jetson Nano Developer Kit. Tout en bas vous verrez l'onglet :

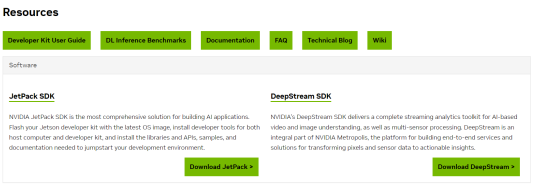


FIGURE 5.1 – Download jetsonPack

ce qui vous retourne vers la page : Et nous sélectionnons la version SD cars nous

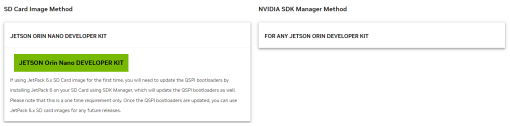


FIGURE 5.2 – Download jetsonPack

allons mettre notre noyau vers une carte SD
pour créer notre image nous utilisons **balenaEtcher** ce qui ressemble à :

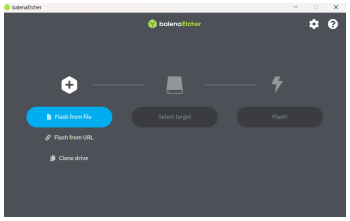


FIGURE 5.3 – Boot système

5.2 Choix organisationnels

Après que l'image soit crée nous avons pu utilisé notre système d'exploitation. **Le projet est dans le répertoire /Documents/Projet_Grace** il faut noté que :

le dossier jetson-inference à ne jamais supprimer ce dossier qui fait le lien symbolique de la bibliothèque pour IA de la jetson nano.
Et mon projet est dans le dossier **/Documents/Projet_Grace/Projet** où vous verrez les code sources de tout ce qui est fait.

5.2.1 Mot de passe

Mot de passe : canneblanche

5.3 Gestion d'objets ou obstacles

La détection d'objets est une étape cruciale pour permettre à un système de comprendre son environnement et de prendre des décisions en conséquence. Les réseaux de neurones profonds (DNN) sont un choix populaire pour la détection d'objets en raison de leur capacité à apprendre des caractéristiques complexes et à fournir des résultats précis.

Pour gérer les objets et les obstacles, on peut utiliser un réseau de détection d'objets pré-entraîné, tel que YOLO (You Only Look Once) ou SSD (Single Shot MultiBox Detector). Ces réseaux sont entraînés sur de vastes ensembles de données pour reconnaître et localiser des objets dans des images.

5.3.1 Avantages des DNN

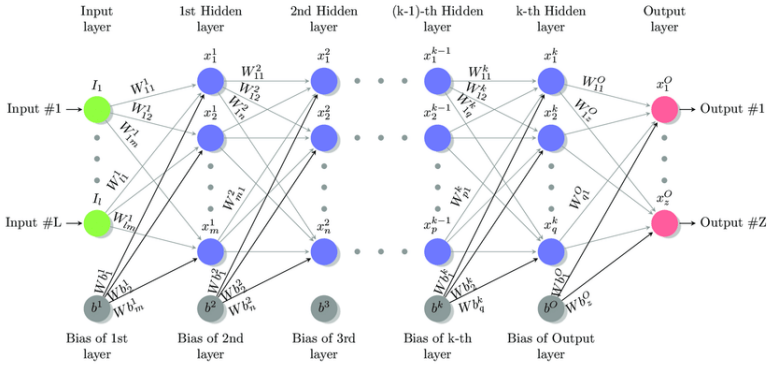


FIGURE 5.4 – DNN

Les DNN peuvent analyser et comprendre le sens des mots. Par exemple, pour un bot de conversation, les DNN peuvent être utilisés pour analyser les requêtes des

utilisateurs et répondre de manière appropriée. Classification d'images.

Les DNN peuvent être utilisés pour reconnaître et classer des objets dans les images. Un DNN peut être entraîné pour détecter les objets spécifiques dans une image. Traduction automatique : Les DNN peuvent être utilisés pour traduire des phrases d'une langue à une autre. Par exemple, pour traduire des textes d'une langue en une autre, les DNN peuvent être utilisés pour analyser le texte source et générer le texte traduit.

Remarques :
Les DNN sont gourmands en calcul et nécessitent de la puissance de traitement importante. C'est pourquoi ils sont parfaitement adaptés aux systèmes embarqués comme Jetson Nano, qui dispose d'un processeur NVIDIA powerful et d'une grande quantité de mémoire vive. L'entraînement d'un DNN peut être chronophage, nécessitant de la patience et de la persévérance. Toutefois, les performances obtenues peuvent être très satisfaisantes, en particulier dans les domaines de l'intelligence artificielle.

Il est important de garder à l'esprit que l'entraînement d'un DNN nécessite une grande quantité de données. Pour optimiser les performances d'un DNN, il est donc conseillé de disposer d'un ensemble de données adéquat et étoffé. Pour réduire les temps d'entraînement et les coûts de calcul, il est possible d'utiliser des techniques d'optimisation, comme le transfert d'apprentissage, qui consiste à réutiliser les poids d'un modèle pré-entraîné pour initier un autre modèle.

En résumé, les DNN offrent de nombreuses possibilités pour créer des systèmes intelligents et puissants, tels que des robots, des voitures autonomes ou des applications mobiles interactives. Grâce à l'adaptabilité de Jetson Nano et à sa capacité de traitement important, les DNN sont particulièrement adaptés à l'utilisation dans des systèmes embarqués.

5.3.2 Choix de la bibliothèque jetson.inference

Le choix de la bibliothèque **jetson.inference** de NVIDIA est judicieux, car elle est spécialement conçue pour les cartes Jetson. Cela garantit une optimisation et des performances élevées, adaptées aux applications d'IA en temps réel sur ces dispositifs.

5.3.3 Fonctionnalités clés de jetson.inference

La description des fonctionnalités clés de **jetson.inference** est précise et informative. Elle met en évidence les capacités importantes telles que la détection d'objets, la reconnaissance d'images et la segmentation d'images, qui sont toutes essentielles pour la conception d'une canne intelligente.

5.3.4 Modèles de réseaux de neurones

La mention des modèles de réseaux de neurones pré-entraînés tels que **SSD-MobileNet-V2, YOLOv3, ResNet, GoogLeNet, FCN, U-Net** est pertinente. Ces modèles sont bien connus pour leur efficacité dans différentes tâches d'inférence.

5.4 Approche approfondie

Nous avons réussi à collecter des images et à détecter des objets. Cependant, un défi subsiste, car il est possible que nous traitons des données qui ne se situent pas sur notre trajectoire. Dans cette perspective, nous adoptons l'approche suivante :

5.4.1 Algorithme des Régions d'Intérêt (ROI)

Les Régions d'Intérêt (ROI) représentent des zones spécifiques d'une image qui sont susceptibles de contenir des informations cruciales. Chaque ROI est définie par le contour, un chemin fermé qui entoure la région d'intérêt. Par exemple, dans le cas d'un ROI rectangulaire, le contour est composé de quatre segments de ligne reliant les quatre points d'angle.

L'algorithme ROI peut être implémenté en utilisant diverses techniques telles que la segmentation d'image, le filtrage spatial, la détection de contours, et le suivi d'objets.

5.4.2 Avantages

Amélioration de la précision et de la performance : En se concentrant sur des zones spécifiques, l'algorithme ROI contribue à une meilleure précision et efficacité du traitement d'image.

Réduction du nombre de détections :
Plutôt que de traiter l'ensemble de l'image, l'algorithme se concentre sur des régions d'intérêt, réduisant ainsi la charge de traitement.

Suivi simultané de plusieurs objets ou régions :
L'algorithme permet le suivi efficace de plusieurs objets ou régions en même temps.

Limitations :
Dépendance aux caractéristiques spécifiques de l'image : Le choix des ROI peut dépendre des propriétés particulières de l'image, ce qui peut nécessiter un ajustement des paramètres pour différents types d'images.

Ajustement des paramètres requis :
Pour obtenir des résultats optimaux, les paramètres de l'algorithme doivent parfois être adaptés en fonction des caractéristiques de l'image.

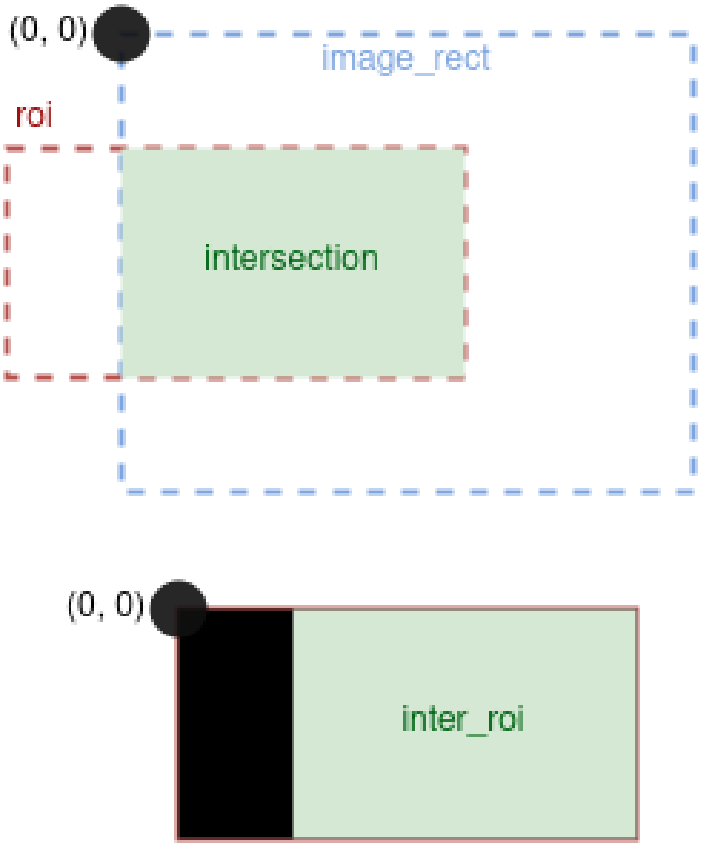


FIGURE 5.5 – Algorithme ROI

Possibilité de dégradation des performances :
Si l'algorithme ne parvient pas à détecter correctement les ROI, cela peut entraîner une dégradation des performances globales.

5.4.3 Désavantage

Un inconvénient potentiel de cette approche est lié au rognage de l'image en référence au centre de l'image. Ceci peut poser problème si l'image est coupée, compromettant ainsi la précision du traitement d'image. Des stratégies supplémentaires peuvent être nécessaires pour atténuer ce désavantage, telles que des mécanismes de correction en cas de découpage inadéquat de l'image.

5.5 Boîte Englobante (BoundingBox)

La Boîte Englobante, souvent appelée BoundingBox, est une représentation graphique essentielle d'un objet ou d'une région au sein d'une image. Il s'agit d'un rectangle aligné avec les axes de coordonnées de l'image, délimité par des points correspondant aux coordonnées minimales et maximales de l'objet ou de la région.

En pratique, une détection précise de la BoundingBox est souvent cruciale. Cela peut être réalisé à l'aide de techniques avancées telles que la **détection de contours, le filtrage spatial ou la segmentation d'image**. Ces approches permettent d'obtenir une délimitation précise de l'objet, offrant ainsi une représentation fidèle de sa position et de sa forme dans l'image.

5.5.1 Avantages

Précision dans la délimitation : La BoundingBox offre une délimitation précise de l'objet ou de la région, ce qui est essentiel pour des applications nécessitant une analyse détaillée.

Adaptabilité aux formes variées : Étant un rectangle aligné sur les axes, la BoundingBox s'adapte à différentes formes d'objets, offrant ainsi une solution flexible.

Simplicité de représentation : En tant que représentation simplifiée, la BoundingBox sert de point de départ pratique pour des analyses plus approfondies.

5.5.2 Limitations

Sensibilité au choix de la méthode de détection : La précision de la BoundingBox dépend de la méthode de détection utilisée, ce qui peut nécessiter des ajustements en fonction des caractéristiques de l'image.

5.5.3 Représentation simplifiée

Bien que la BoundingBox soit utile, elle ne capture pas toutes les subtilités de l'objet ou de la région, ce qui peut être un inconvénient dans des scénarios nécessitant une analyse plus fine.

En conclusion, la BoundingBox constitue une approche efficace pour représenter graphiquement des objets ou des régions dans une image. Elle offre un équilibre entre la simplicité de représentation et la précision, facilitant ainsi des analyses plus approfondies telles que la classification de l'objet ou la détection de caractéristiques spécifiques. Pour déterminer l'object proche du centre nous procédons comme suite :

```
1 center_x = img.width // 2
2 center_y = img.height // 2
3 # Boucle sur les d tections pour trouver l'objet le plus
     proche du centre de l'image.
4 for detection in detections:
5     # Calcul de la distance entre l'objet et le centre de l'
     image.
6     distance = ((detection.Left + detection.Right) // 2 -
     center_x)**2 + ((detection.Top + detection.Bottom) // 2 -
     center_y)**2
```

5.6 Évaluation des Modèles de Formation

Dans le cadre de notre projet, nous avons entrepris des tests avec deux modèles de formation populaires pour la détection d'objets, à savoir **ssd-mobilenet-v2 et ssd-inception-v2.** Ces modèles sont bien établis et largement utilisés dans la communauté de la vision par ordinateur.

5.7 ssd-mobilenet-v2

5.7.1 Avantages

Légèreté :
Mobilenet-v2 est connu pour sa légèreté et sa capacité à maintenir des performances élevées même sur des dispositifs embarqués avec des ressources limitées. Vitesse : Sa conception spécifique permet des temps d'inférence rapides, ce qui est crucial pour une application en temps réel comme la détection d'obstacles. Limitations :

Moins précis pour certaines tâches complexes :
Bien que très performant pour de nombreuses applications, **ssd-mobilenet-v2** peut montrer des limitations dans la détection d'objets plus complexes ou détaillés.

5.8 ssd-inception-v2

5.8.1 Avantages

Précision accrue :
Inception-v2, avec son architecture plus complexe, peut offrir une meilleure précision dans la détection d'objets complexes. Adaptabilité à des scénarios variés : Convient aux tâches de détection d'objets nécessitant une analyse fine. Limitations :

Plus lourd en termes de ressources : Du fait de son architecture plus complexe, **ssd-inception-v2** peut demander plus de ressources computationnelles, ce qui peut être un défi sur des dispositifs embarqués.

5.9 Choix du Modèle de Training

Après des tests approfondis et une évaluation comparative, nous avons pris la décision de travailler avec le modèle **ssd-mobilenet-v2**. Notre choix repose sur plusieurs considérations :

Efficacité :
La légèreté de **ssd-mobilenet-v2**, tout en maintenant des performances acceptables, est cruciale pour notre application embarquée où les ressources sont limitées.

Temps d'inférence :
La rapidité des temps d'inférence de **ssd-mobilenet-v2** s'aligne parfaitement avec notre exigence d'une détection en temps réel, améliorant ainsi l'expérience utilisateur.

Adéquation au Contexte :
Bien que **ssd-inception-v2** offre une précision supérieure dans certaines situations, **ssd-mobilenet-v2** répond adéquatement à notre besoin de détecter des obstacles dans des environnements variés.

Conclusion :
Le choix du modèle **ssd-mobilenet-v2** repose sur une évaluation pragmatique des avantages et des limitations de chaque modèle, avec un accent particulier sur l'efficacité, la vitesse d'inférence et l'adéquation à notre contexte d'application. Nous sommes convaincus que ce choix garantira des performances optimales dans le cadre spécifique de notre projet de canne intelligente pour aveugles.

5.10 Algo : Pseudo code

5.10.1 Les fonctions

Différentes fonctions :Fonction pour synthèse vocale : text_to_speech() Fonction pour capteur de distance : sensor_lidar_front() – file uart_sensor_a02yyuw.py Fonction pour la détection d'objets : threading_camera_front()

```
threading_camera_front() :
#differents model de training
  listModelTraining =["ssd-mobilenet-v2","ssd-inception-v2" ]
  #on precure notre choix de training
  modelTraining = listModelTraining[1]
  #definition de model de traitement avec seuil de detection
  detectNet(modelTraining, threshold=0.5)

  while IsStreaming() :

    #lire le flux vid o
    img = captutreImage()

    #detection  d image
    detection  = Detect(img)

    #recupere les images proches du centre de  l image
    center_x = img.width // 2
          center_y = img.height // 2

    #pour chaque element d tecte on calcul  distance en
   fonction du centre

```

```
    closest_detection = elementProcheduCentre()
     nom_element = getClass(closest_detection)
    setNom(nom_elemnt)

sensor_lidar_front() :

  sensor_A02YYUW_init(portCom, baudRate)
  while True :
    calcul_distance(peripheral_conf)
    if distance >=0 && distance <=2 :
      getNameObjet()
    if threadSynthese() is None or not is_alive() :
      thread(text_to_speech(), message)

text_to_speech(text) :
  tts = gTTs(text, fr)
  tts.save()
  os.system(audio)

Main () :
  Thread (sensor_lidar_front())
   threading_camera_front()
```

5.10.2 Problème Rencontré :

Synthèse Vocale :

Au cours de notre développement, nous avons rencontré un défi spécifique lié à la synthèse vocale. La librairie gTTS (Google Text-to-Speech) que nous avons initialement choisie pour générer des fichiers audio à partir de texte présente une limitation majeure :
Elle nécessite une connexion internet pour fonctionner correctement.

Solution Adoptée

Pour surmonter cette contrainte, nous avons opté pour une approche alternative en utilisant une commande système pour générer le son. Concrètement, nous avons développé un **script shell** nommé **text_to_speech.sh** qui prend en paramètre le texte à synthétiser et génère le fichier audio correspondant. Par exemple

```
./text_to_speech.sh "Hello, World !"
```

5.10.3 Avantages de la Nouvelle Approche

Indépendance Internet
: En évitant la dépendance à une connexion en ligne, notre solution permet l'utilisation de la synthèse vocale même dans des environnements où la connectivité Internet

peut être limitée ou absente.

Contrôle Local : En utilisant une commande système locale, nous avons un contrôle direct sur le processus de synthèse vocale, offrant ainsi une solution plus robuste et autonome.

Intégration Facile : La modification vers une approche de commande système n'affecte pas de manière significative l'intégration de la synthèse vocale dans notre application, permettant une transition fluide.

Considérations :
Bien que cette solution résolve le problème initial lié à la dépendance Internet de gTTS, il est important de noter que cela introduit une dépendance au système d'exploitation sous-jacent et aux logiciels associés pour la synthèse vocale. Les détails de cette dépendance doivent être pris en compte pour assurer une compatibilité et une stabilité continues.

Implémentation shell

```
#!/bin/bash

# Set the sound card index here (e.g., 2 for the USB sound card)
SOUND_CARD_IDX=2

# Check if the espeak command is available
if ! [ -x "$(command -v espeak)" ]; then
  echo 'Error: espeak could not be found, please install it first.' >&2
  exit 1
fi

# Check if the aplay command is available
if ! [ -x "$(command -v aplay)" ]; then
  echo 'Error: aplay could not be found, please install it first.' >&2
  exit 1
fi

# Convert the input text to speech and play it using the specified sound card
espeak "$@" --stdout | aplay -D plughw:$SOUND_CARD_IDX,0
```

5.11 Automatisation du Lancement au Démarrage du Système

Après avoir réalisé un travail substantiel sur notre projet, il est logique de vouloir mettre en place un système qui s'initialise automatiquement au démarrage, garantissant ainsi une utilisation sans intervention manuelle.
Cette automatisation peut être réalisée en configurant le lancement automatique du script au démarrage du système d'exploitation.

Étapes pour l'Automatisation :
Localisation du Script : s'assurer que le script **main.py** est situé dans un emplacement accessible.

Permission d'Exécution : s'assurer que le script a les permissions d'exécution nécessaires.

5.11.1 Création d'un Fichier de Service

```
[Unit]
Description=service pour aveugles

[Service]
ExecStart=/chemin/vers/le/script/main.py
Restart=always
User=utilisateur
Group=groupe

[Install]
WantedBy=default.target
```

Un service porte l'extension **.service**

- ExecStart : Chemin absolu vers votre script.
- User et Group : L'utilisateur et le groupe sous lesquels vous souhaitez exécuter le script.

Activation du Service : Activez le service Systemd que vous avez créé :

```
sudo systemctl enable aveugles.service
```

Redémarrage du Système : Vous pouvez redémarrer le système ou simplement démarrer le service pour qu'il prenne effet :

```
 sudo systemctl start aveules.service

```

Vérification de l'État du Service : Vous pouvez vérifier l'état du service pour vous assurer qu'il fonctionne correctement :

```
sudo systemctl status aveugles.service
```

5.12 Architecture Matérielle

— Intégration des capteurs : Connexion de la caméra et du capteur ultrasons à la Jetson Nano.

— Traitement des données : La Jetson Nano analyse les données des capteurs pour détecter les obstacles.

5.12.1 Schéma de Câblage

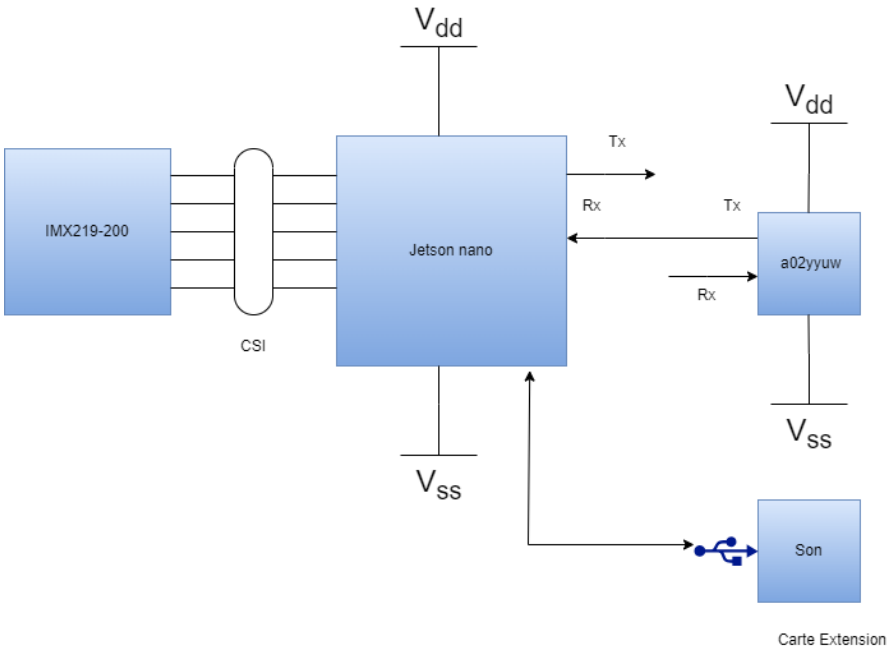
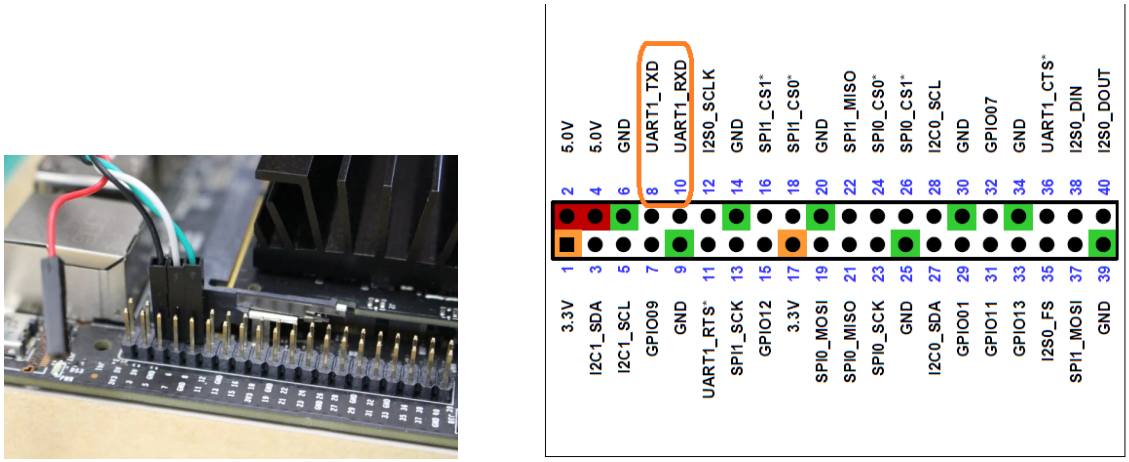


FIGURE 5.6 – Cablage

Pins Jetson nano



(a) Physique Pins (b) Pins

FIGURE 5.7 – Description générale des pins

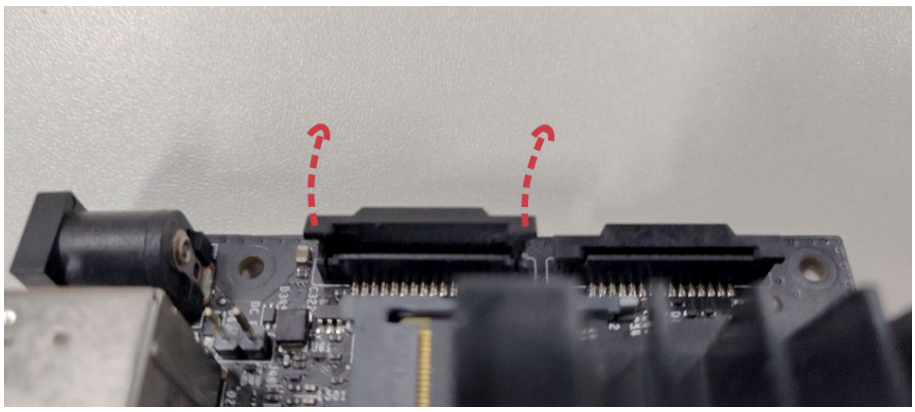


FIGURE 5.8 – Slot Caméra

Informations complémentaires

Bibliographie