



Cat Recognition choix de l'ordinateur embarqué

1. Introduction

Dans le cadre de notre projet, l'objectif initial était d'implémenter une intelligence artificielle sur le **TurtleBot3 Burger**, un robot reconnu pour son efficacité en navigation autonome. Ce robot est fourni avec un **Raspberry Pi 4** en tant que système embarqué. Afin de minimiser le nombre d'ordinateurs embarqués et de simplifier l'architecture du système, nous avons choisi d'exécuter l'ensemble des tâches, y compris celles liées à l'IA, directement sur le Raspberry Pi 4.

Avant de finaliser ce choix, une comparaison des capacités GPU entre le Raspberry Pi 4 et d'autres solutions a été effectuée. Le Raspberry Pi 4 est équipé d'un GPU **VideoCore VI**. Bien que ce GPU soit moins puissant que celui du **NVIDIA Jetson Nano**, qui dispose de 128 cœurs CUDA et est spécifiquement conçu pour des tâches d'IA, nous avons estimé que les performances du Raspberry Pi 4 seraient suffisantes pour notre application, compte tenu de la nature des tâches d'IA envisagées et des contraintes budgétaires.

2. Problèmes rencontrés sur le Raspberry Pi

Limitations matérielles

Lors de l'implémentation de l'IA sur le **Raspberry Pi 4**, plusieurs limitations matérielles ont rapidement freiné notre progression :



- **Mémoire insuffisante** : Avec **4 Go de RAM**, le Raspberry Pi 4 montre vite ses limites pour les tâches d'IA gourmandes en ressources. En particulier, l'inférence de modèles de vision par ordinateur consomme rapidement toute la mémoire disponible, provoquant des ralentissements et des crashes.
- **Incompatibilité avec les versions récentes des bibliothèques d'IA et de traitement d'image** :
 - De nombreuses bibliothèques comme **TensorFlow, PyTorch et OpenCV** ne sont pas optimisées pour l'architecture ARM du Raspberry Pi.
 - Certaines versions récentes nécessitent des **backports**, des compilations spécifiques ou l'utilisation de versions obsolètes, limitant l'accès aux dernières optimisations et fonctionnalités.
- **GPU sous-performant** : Le **VideoCore VI** du Raspberry Pi 4 est nettement insuffisant pour accélérer les calculs d'IA, contrairement aux GPU CUDA des cartes NVIDIA.
 - Ce manque de puissance affecte particulièrement les tâches nécessitant du rendu graphique en temps réel.



- Par exemple, lors de la simple récupération du flux vidéo de la caméra, au lieu d'afficher une image fluide et continue, le Raspberry Pi ouvrait **des milliers de fenêtres successives** à chaque rafraîchissement d'image, montrant une mauvaise gestion de l'affichage et des ressources.

3. Éventuelles solutions tentées avant d'abandonner cette piste

Face aux limitations matérielles et logicielles du **Raspberry Pi 4**, plusieurs optimisations ont été mises en place pour tenter d'améliorer les performances et la stabilité du système.

Utilisation de versions légères des bibliothèques d'IA

- Installation de **TensorFlow Lite** et d'**OpenCV optimisé pour ARM** afin de réduire l'empreinte mémoire et d'améliorer l'exécution des modèles.
- Réduction de la taille des modèles d'IA en diminuant le nombre de paramètres et en appliquant une **quantification** (passage de 32 bits à 8 bits), ce qui permet d'exécuter les réseaux de neurones sur du matériel limité, au prix d'une légère perte de précision.

Optimisation du traitement vidéo

- Mise en place d'une architecture modulaire sous **ROS** avec un **nœud de lancement indépendant** dédié à la gestion de la caméra en USB.
- Optimisation des paramètres de capture et de traitement pour limiter l'impact sur les ressources CPU et mémoire.

Malgré ces ajustements, le **manque de puissance brute** et les **limitations logicielles** du Raspberry Pi ont rapidement constitué un frein majeur à l'implémentation d'une IA efficace pour notre projet. Il a donc été décidé d'abandonner cette approche au profit d'une **Jetson**, bien mieux adaptée aux charges de travail liées à l'intelligence artificielle et à la vision par ordinateur.

4. Passage à la Jetson

Raisons du choix

Face aux limitations du **Raspberry Pi 4**, la décision a été prise de basculer sur une **Jetson** pour l'implémentation de l'IA. Plusieurs raisons ont motivé ce choix :



- **Puissance du GPU** : Contrairement au VideoCore VI du Raspberry Pi, les cartes Jetson intègrent un **GPU NVIDIA** avec une prise en charge native du **calcul parallèle**, ce qui accélère considérablement l'exécution des modèles d'IA.



- **Compatibilité avec les modèles IA** : La Jetson est conçue pour fonctionner avec **TensorFlow, PyTorch, et OpenCV** sans nécessiter de modifications ou d'optimisations complexes.
- **Support CUDA et TensorRT** : Grâce à l'accélération matérielle par **CUDA**, il est possible d'optimiser les inférences pour obtenir des performances bien supérieures à celles du Raspberry Pi.
- **Expérience préalable** : Ayant déjà travaillé avec la Jetson, je savais qu'elle offrait une solution **fiable et efficace** pour des tâches d'IA et de traitement d'image en temps réel.

Avantages et inconvénients

Le passage à la Jetson présente **de nombreux avantages** :

- Performances nettement **supérieures** pour l'IA et le traitement vidéo.
- Meilleure **compatibilité logicielle** avec les frameworks d'apprentissage automatique.
- Accès aux **outils NVIDIA** (CUDA, TensorRT, DeepStream) pour optimiser les modèles IA.

Cependant, un inconvénient majeur s'est posé :

- Il faut **choisir entre deux configurations** :
 1. **Embarquer deux ordinateurs** : utiliser la Jetson pour l'IA et **conserver le Raspberry Pi** pour la gestion des moteurs et de la navigation autonome.
 2. **Tout intégrer sur la Jetson** : mais cela implique de **réinstaller tous les drivers** et logiciels de navigation autonome (ROS, communication avec les moteurs, etc.), ce qui ajoute une complexité supplémentaire.

Processus d'installation et mise en place de l'IA

Le processus d'installation et de configuration de l'IA sur la Jetson sera détaillé dans le rapport suivant, où seront décrites les étapes précises d'installation des bibliothèques et des optimisations mises en place.