Détecteur de panneaux de signalisation pour voiture

Colin de Seroux, Tristan Larguier, Quentin Maurois,
Skander Meziou, Mathilde Razafimahatratra, Yu Song

Objectif du projet :

CONCEVOIR UN SYSTÈME DE DÉTECTION DE SIGNALISATION POUR LES VOITURES EN UTILISANT L'INTELLIGENCE ARTIFICIELLE

Le système :

- RÉCUPÉRATION DE LA POSITION GPS ET DE LA VITESSE
- ANALYSE D’IMAGE ET RECONNAISSANCE DE PANNEAUX
- PRISE DE DÉCISION ET ACTION TEMPS RÉEL
- OBJECTIF : 5 INFÉRENCES PAR SECONDE

Choix matériels :

Produit	Usage	Description
Raspberry PI4 & PI5	Gestion de la caméra, asservissement de la vitesse	Faible consommation énergétique, processeur assez puissant pour gérer les opérations à réaliser
TPU (Tensor Processing Unit)	Accélère les inférences de l'IA permettant une accélération du processus pour l'analyse de l'image	Accélère théoriquement les calculs d'inférence
GPS	Avoir accès à la position de la Raspberry qui se trouve dans la voiture.	Grove-GPS
Caméra	Fournit les données pour la reconnaissance d'objet	Capture les images pour l'analyse visuelle

Détecter les panneaux

Algorithmes choisis :

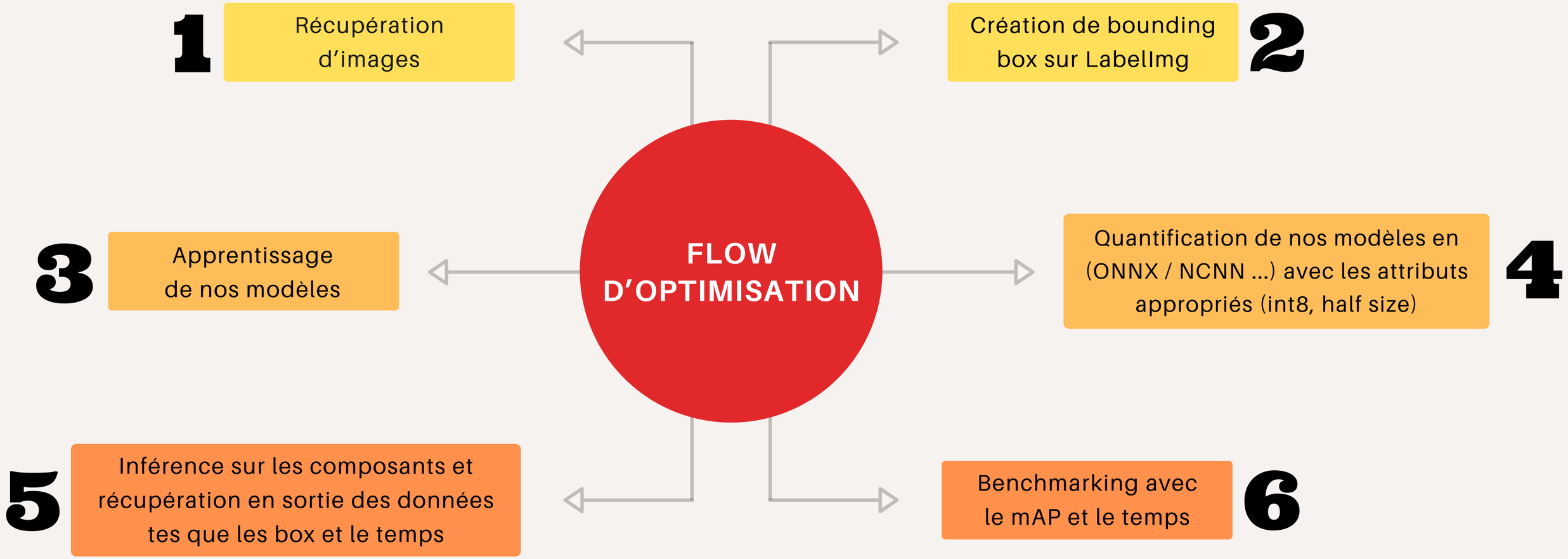
- YOLO11 : Rapidité et précision
- Fast R-CNN : Meilleure précision pour des tâches complexes

Optimisations logicielles :

- ONNX / OpenVINO / Edge TPU / NCNN pour la comptabilité multiplateforme et pour améliorer les performances pour chaque matériel

Plateformes matérielles:

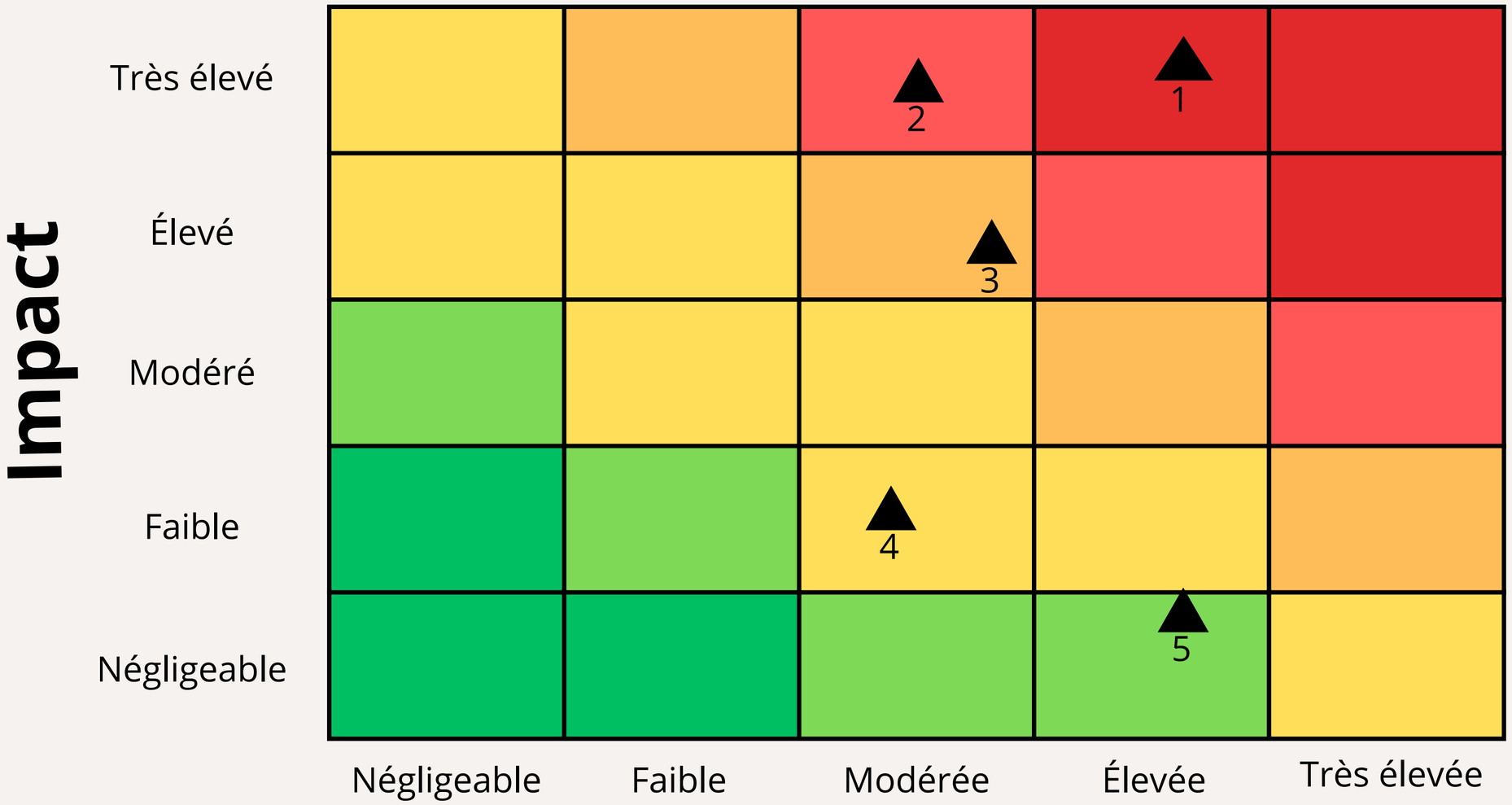
- Raspberry Pi 4 2GB (CPU)
- Raspberry Pi 4 8GB (CPU)
- Raspberry Pi 5 4GB (CPU)
- Google Coral (TPU)
- RTX 3080 TI (GPU)
- Hailo 8L (ASIC)


1
Récupération d'images
2
Création de bounding box sur LabelImg
FLOW D'OPTIMISATION
3
Apprentissage de nos modèles
4
Quantification de nos modèles en (ONNX / NCNN ...) avec les attributs appropriés (int8, half size)
5
Inférence sur les composants et récupération en sortie des données tes que les box et le temps
6
Benchmarking avec le mAP et le temps

Analyse des risques

Risque	Description de l'impact	Impact	Probabilité	Solution
Environnement de la caméra	Si la caméra est mal placée, elle ne pourra pas récupérer les informations se trouvant sur la route	Très élevé	Élevée	Il faut placer la caméra au niveau du tableau de bord
Identification des panneaux manqués	La caméra ne détecte pas correctement les panneaux (en raison de la distance, d'une mauvaise météo ou autre obstruction).	Très élevé	Modérée	Utiliser les données GPS couplée à une carte des vitesses maximales autorisées pour déterminer la vitesse maximale locale.
Circulation en zone de travaux	Risque de conflit dû à un mauvais masquage des panneaux habituels ou à une utilisation de la vitesse de la carte	Élevé	Modérée	Traiter en priorité les panneaux de travaux.
Stabilité de la caméra	Les images peuvent être légèrement floues, ce qui peut amener des erreurs.	Faible	Modérée	Stabilisateur physique ou logiciel
OpenStreetMap non à jours	Certaines routes ne seront pas indiquées ainsi que les travaux, indiquent seulement la limitation	Négligeable	Élevée	

Matrice des risques



1 Environnement caméra

2 Panneaux manqués

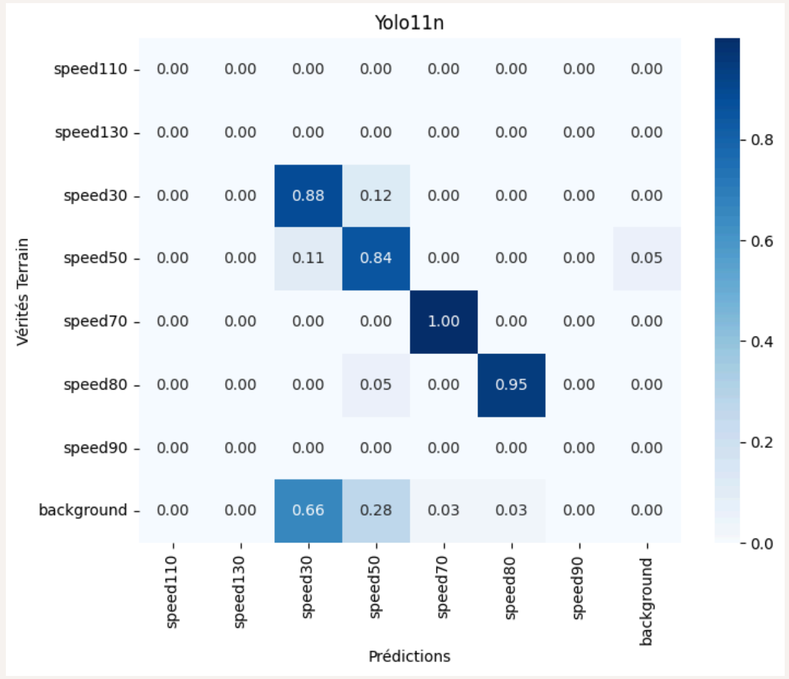
3 Zone de travaux

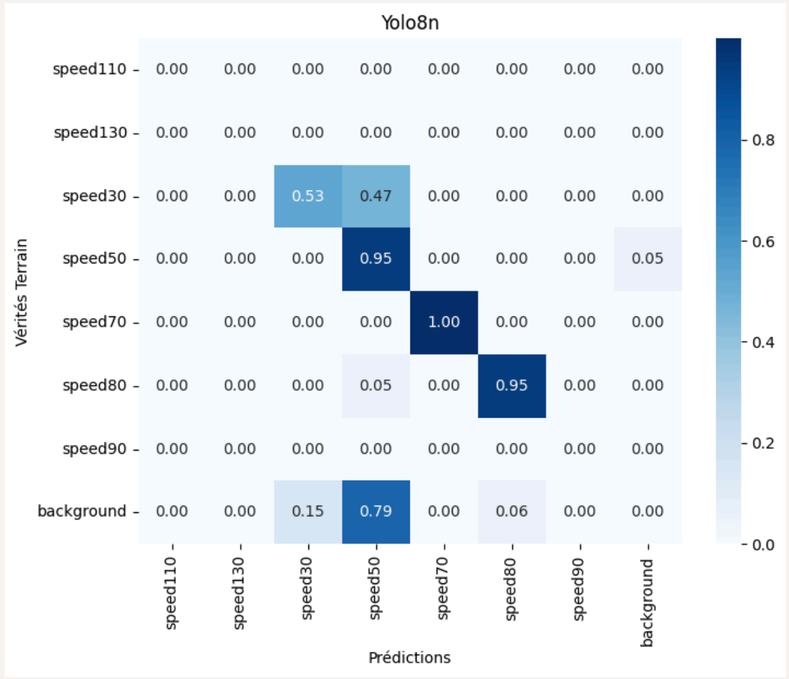
4 Stabilité caméra

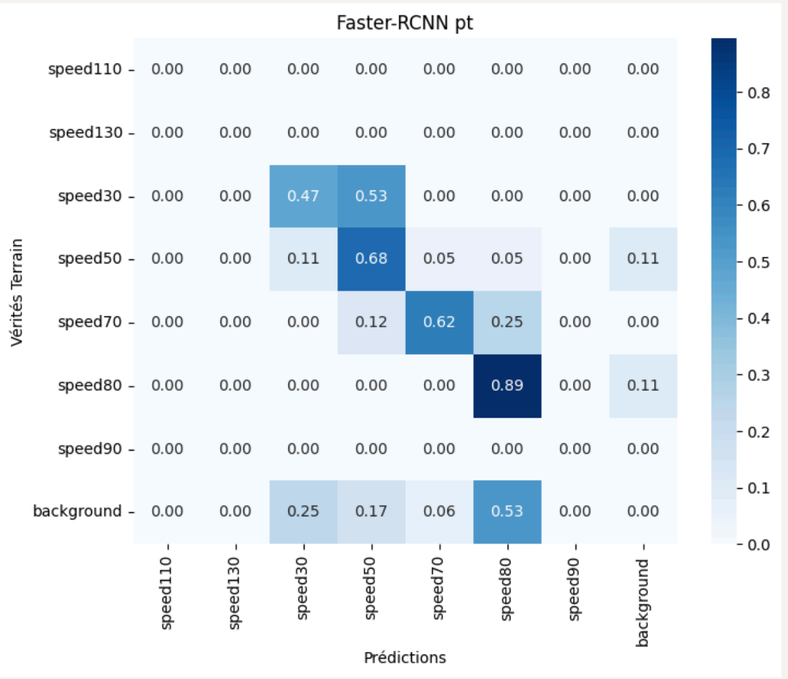
5 OpenStreetMap

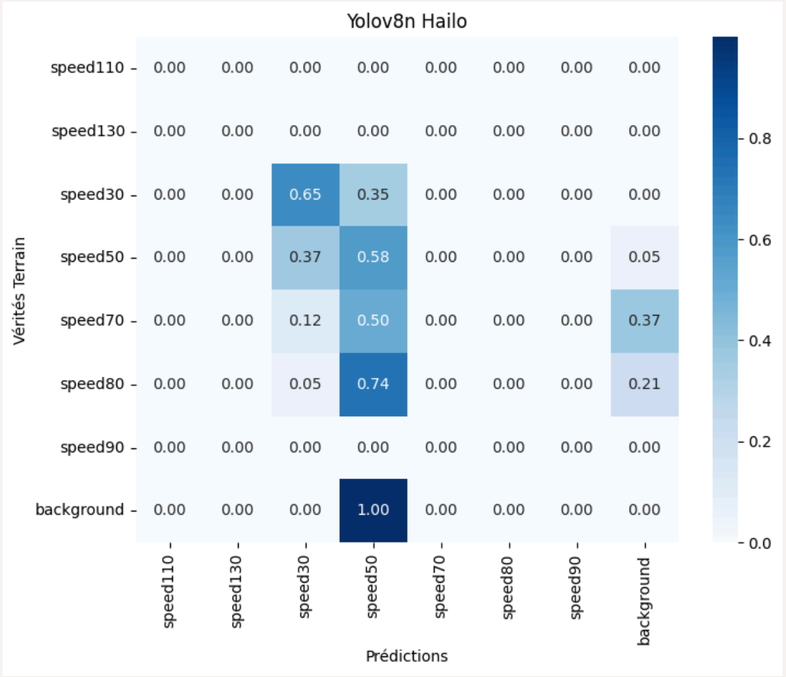
Contraintes

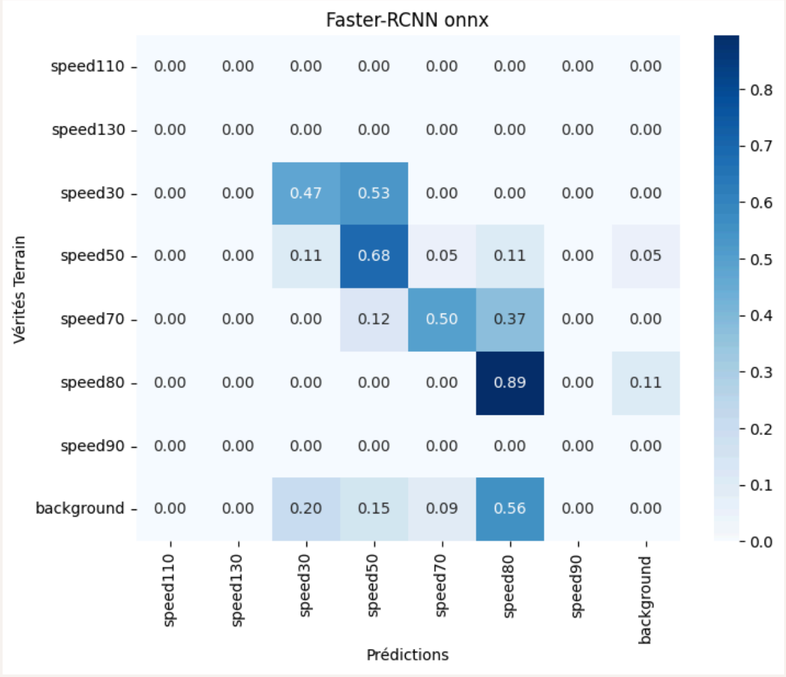
Materielles	Ressources limitées	Raspberry PI 4 ou 5, limite les opérations en tps réel sur des images toutes les 200 ms.
Performances	Temps de traitement	Dois être inférieur ou égal à 200ms
	Distance de détection maximale	Distance minimale de 15 mètres
Environnementales	Zone géographique	Les data sets utilisés ne concerne que la France
	Heures de fonctionnement	Doit être fonctionnel à toutes les heures (à condition que l'éclairage de la voiture sois fonctionnel)
Logicielle	Modèles légers	YOLO11 compatibles avec les plateformes embarquée

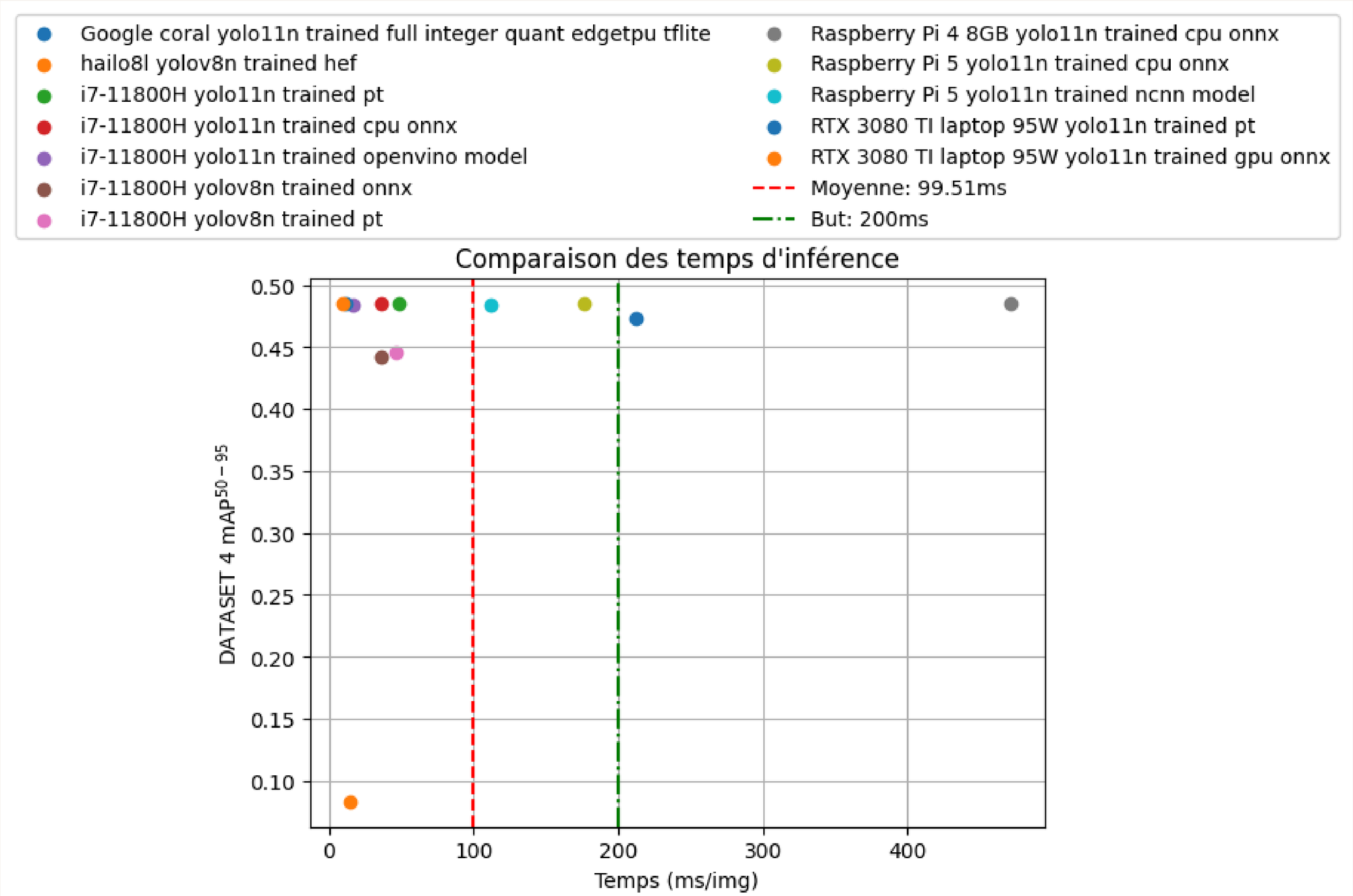

Yolo11n
Vérités Terrain
Prédictions
speed110 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed130 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed30 0.00 0.00 0.88 0.12 0.00 0.00 0.00 0.00
speed50 0.00 0.00 0.11 0.84 0.00 0.00 0.00 0.05
speed70 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
speed80 0.00 0.00 0.00 0.05 0.00 0.95 0.00 0.00
speed90 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
background 0.00 0.00 0.66 0.28 0.03 0.03 0.00 0.00
speed110 speed130 speed30 speed50 speed70 speed80 speed90 background
0.0 0.2 0.4 0.6 0.8

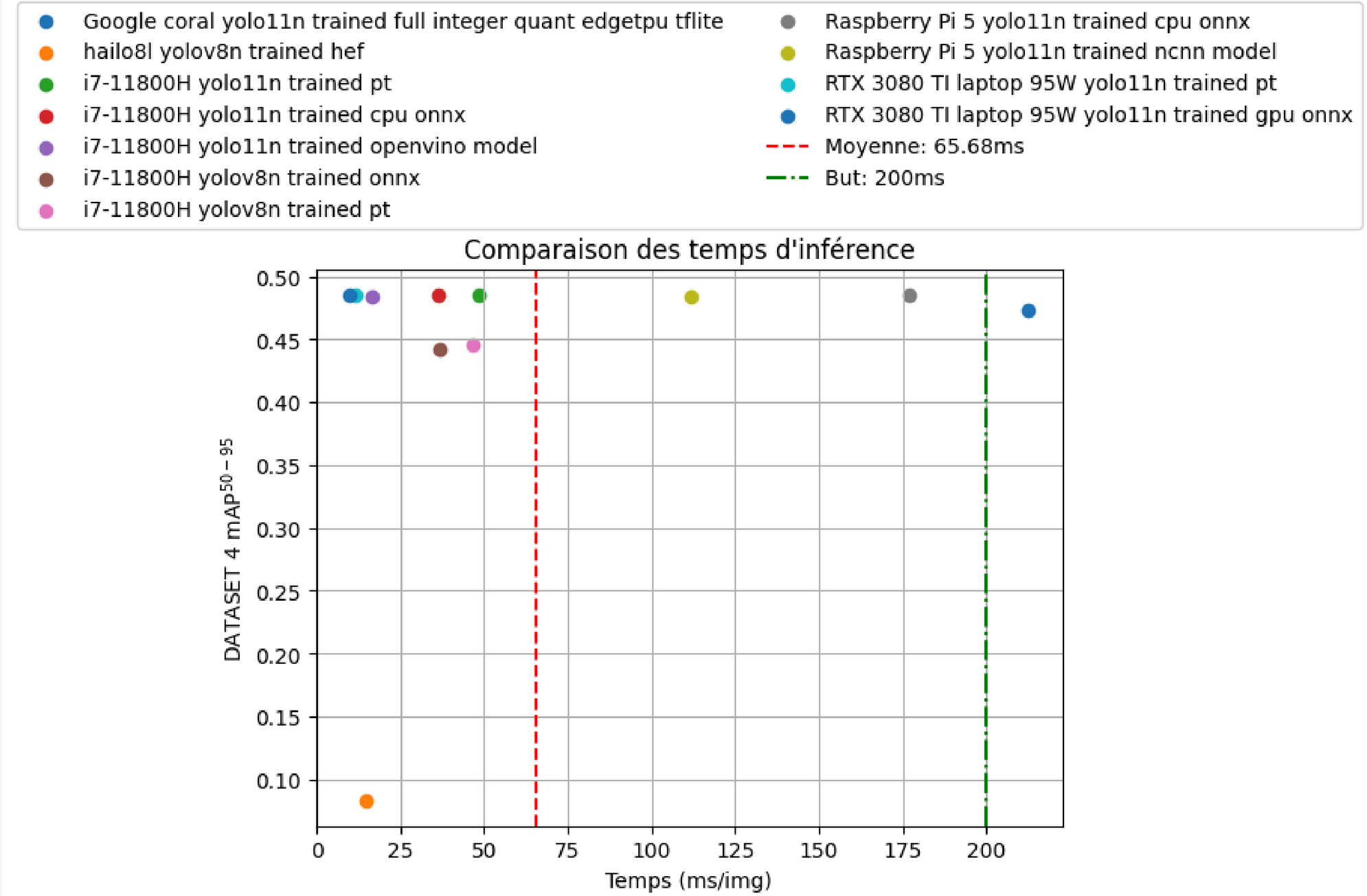

Yolo8n
Vérités Terrain
Prédictions
speed110 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed130 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed30 0.00 0.00 0.53 0.47 0.00 0.00 0.00 0.00
speed50 0.00 0.00 0.00 0.95 0.00 0.00 0.00 0.05
speed70 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
speed80 0.00 0.00 0.00 0.05 0.00 0.95 0.00 0.00
speed90 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
background 0.00 0.00 0.15 0.79 0.00 0.06 0.00 0.00
speed110 speed130 speed30 speed50 speed70 speed80 speed90 background
0.0 0.2 0.4 0.6 0.8

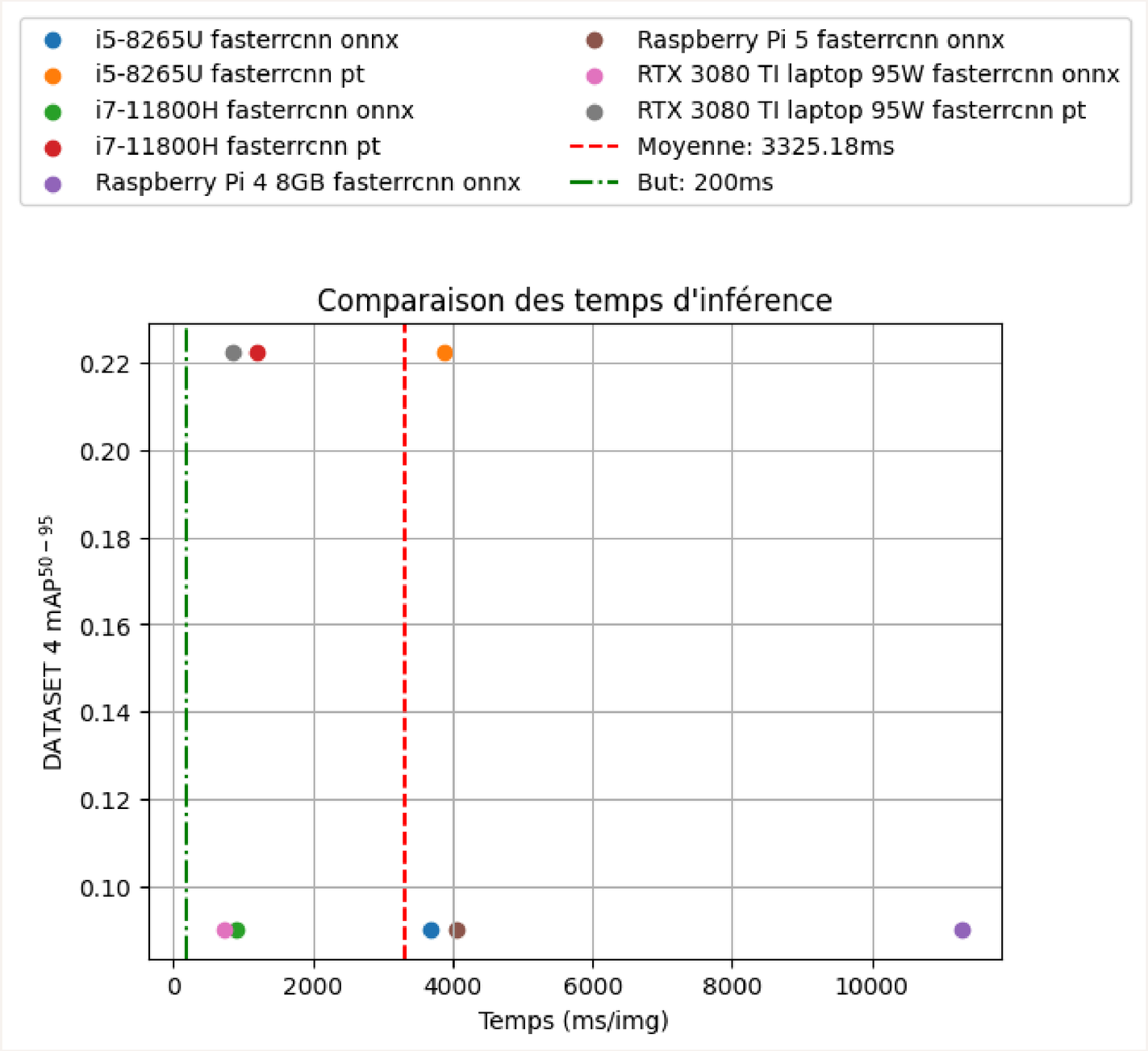

Faster-RCNN pt
Vérités Terrain
Prédictions
speed110 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed130 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed30 0.00 0.00 0.47 0.53 0.00 0.00 0.00 0.00
speed50 0.00 0.00 0.11 0.68 0.05 0.05 0.00 0.11
speed70 0.00 0.00 0.00 0.12 0.62 0.25 0.00 0.00
speed80 0.00 0.00 0.00 0.00 0.00 0.89 0.00 0.11
speed90 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
background 0.00 0.00 0.25 0.17 0.06 0.53 0.00 0.00
speed110 speed130 speed30 speed50 speed70 speed80 speed90 background
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8

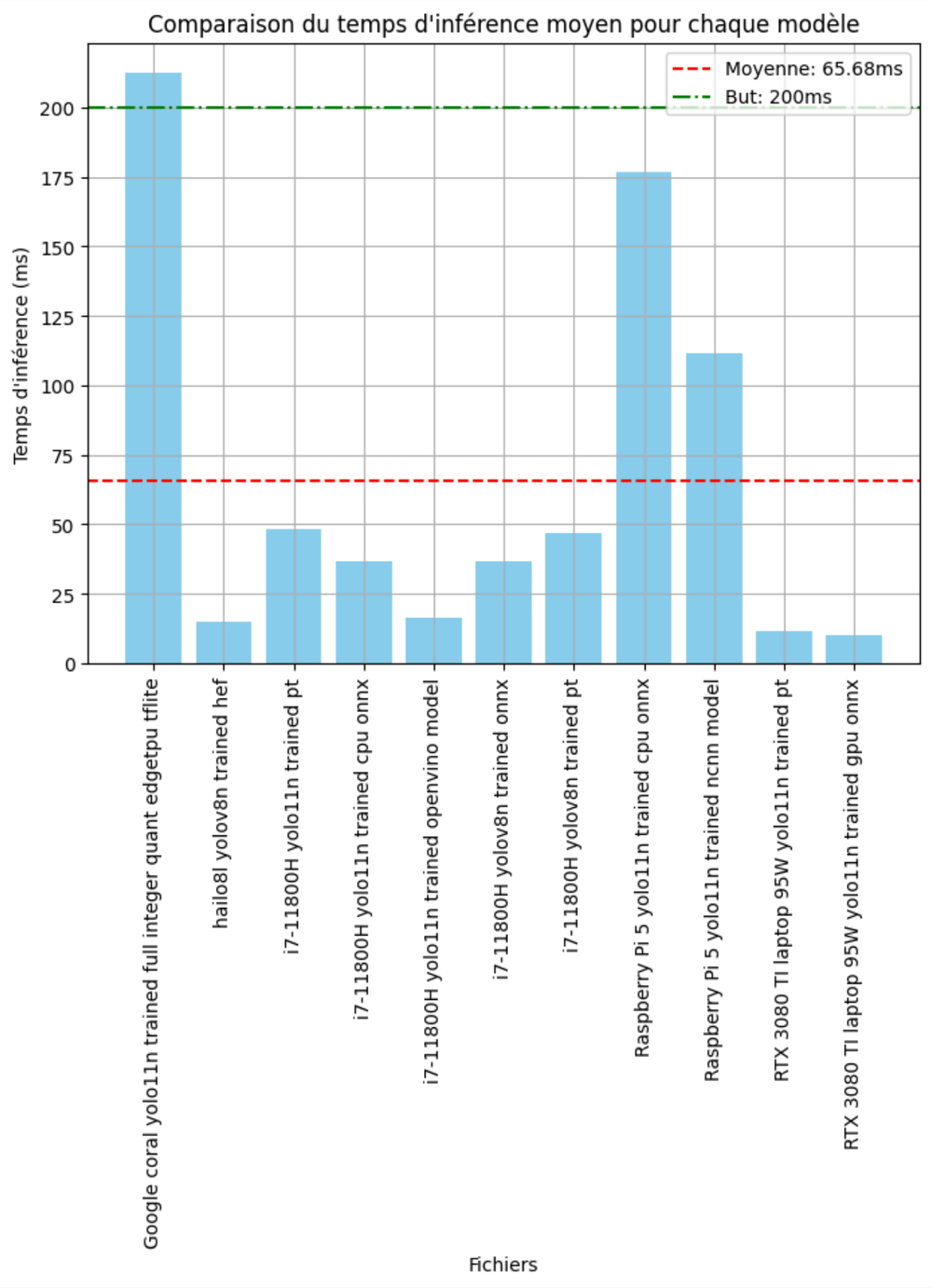

Yolov8n Hailo
Vérités Terrain
Prédictions
speed110 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed130 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed30 0.00 0.00 0.65 0.35 0.00 0.00 0.00 0.00
speed50 0.00 0.00 0.37 0.58 0.00 0.00 0.00 0.05
speed70 0.00 0.00 0.12 0.50 0.00 0.00 0.00 0.37
speed80 0.00 0.00 0.05 0.74 0.00 0.00 0.00 0.21
speed90 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
background 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00
speed110 speed130 speed30 speed50 speed70 speed80 speed90 background
0.0 0.2 0.4 0.6 0.8

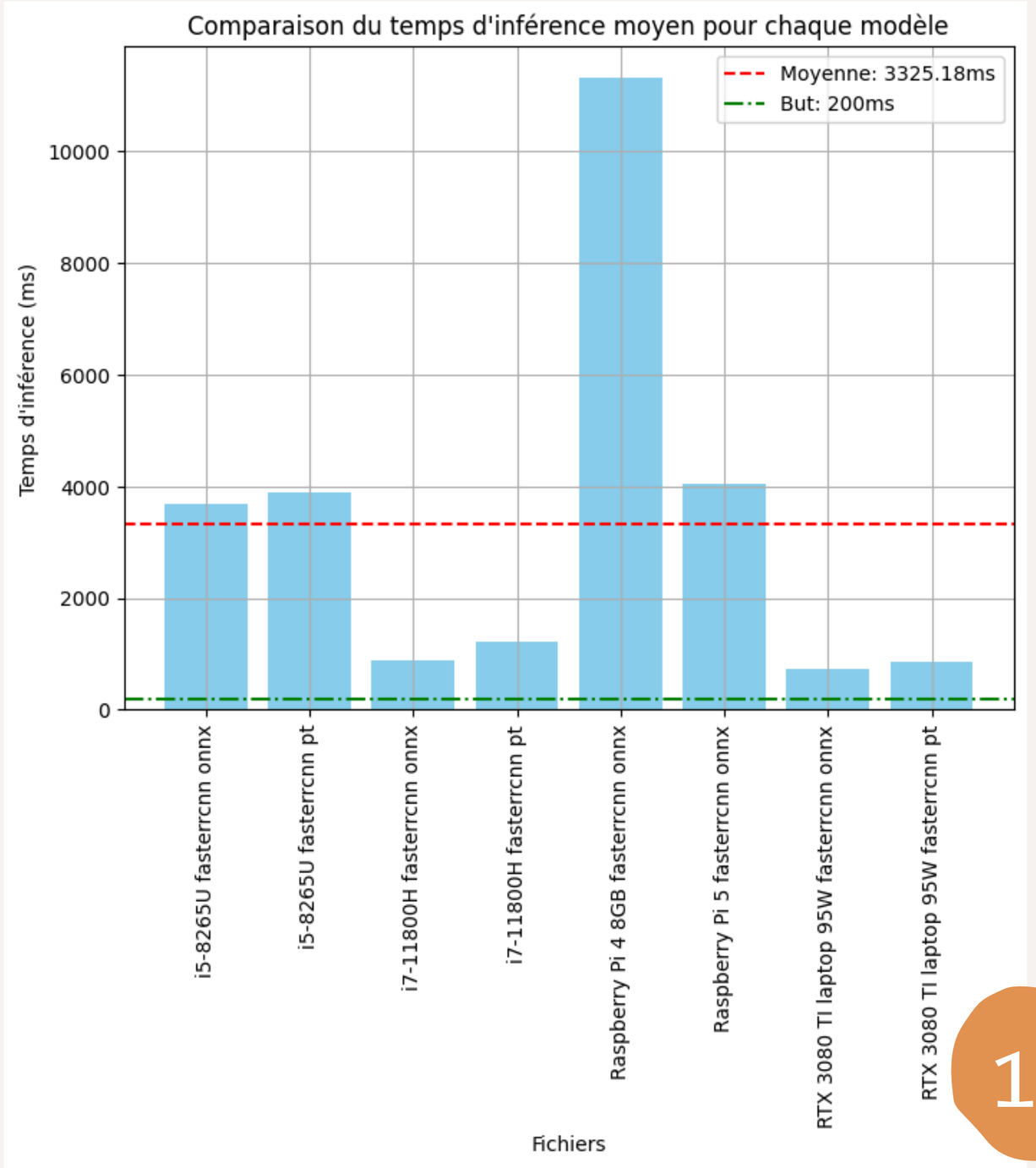

Faster-RCNN onnx
Vérités Terrain
Prédictions
speed110 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed130 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
speed30 0.00 0.00 0.47 0.53 0.00 0.00 0.00 0.00
speed50 0.00 0.00 0.11 0.68 0.05 0.11 0.00 0.05
speed70 0.00 0.00 0.00 0.12 0.50 0.37 0.00 0.00
speed80 0.00 0.00 0.00 0.00 0.00 0.89 0.00 0.11
speed90 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
background 0.00 0.00 0.20 0.15 0.09 0.56 0.00 0.00
speed110 speed130 speed30 speed50 speed70 speed80 speed90 background
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8


Google coral yolo11n trained full integer quant edgetpu tflite
hailo8l yolov8n trained hef
i7-11800H yolo11n trained pt
i7-11800H yolo11n trained cpu onnx
i7-11800H yolo11n trained openvino model
i7-11800H yolov8n trained onnx
i7-11800H yolov8n trained pt
Raspberry Pi 4 8GB yolo11n trained cpu onnx
Raspberry Pi 5 yolo11n trained cpu onnx
Raspberry Pi 5 yolo11n trained ncnn model
RTX 3080 TI laptop 95W yolo11n trained pt
RTX 3080 TI laptop 95W yolo11n trained gpu onnx
Moyenne: 99.51ms
But: 200ms
Comparaison des temps d'inférence
DATASET 4 mAP50 – 95
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
100
200
300
400
Temps (ms/img)

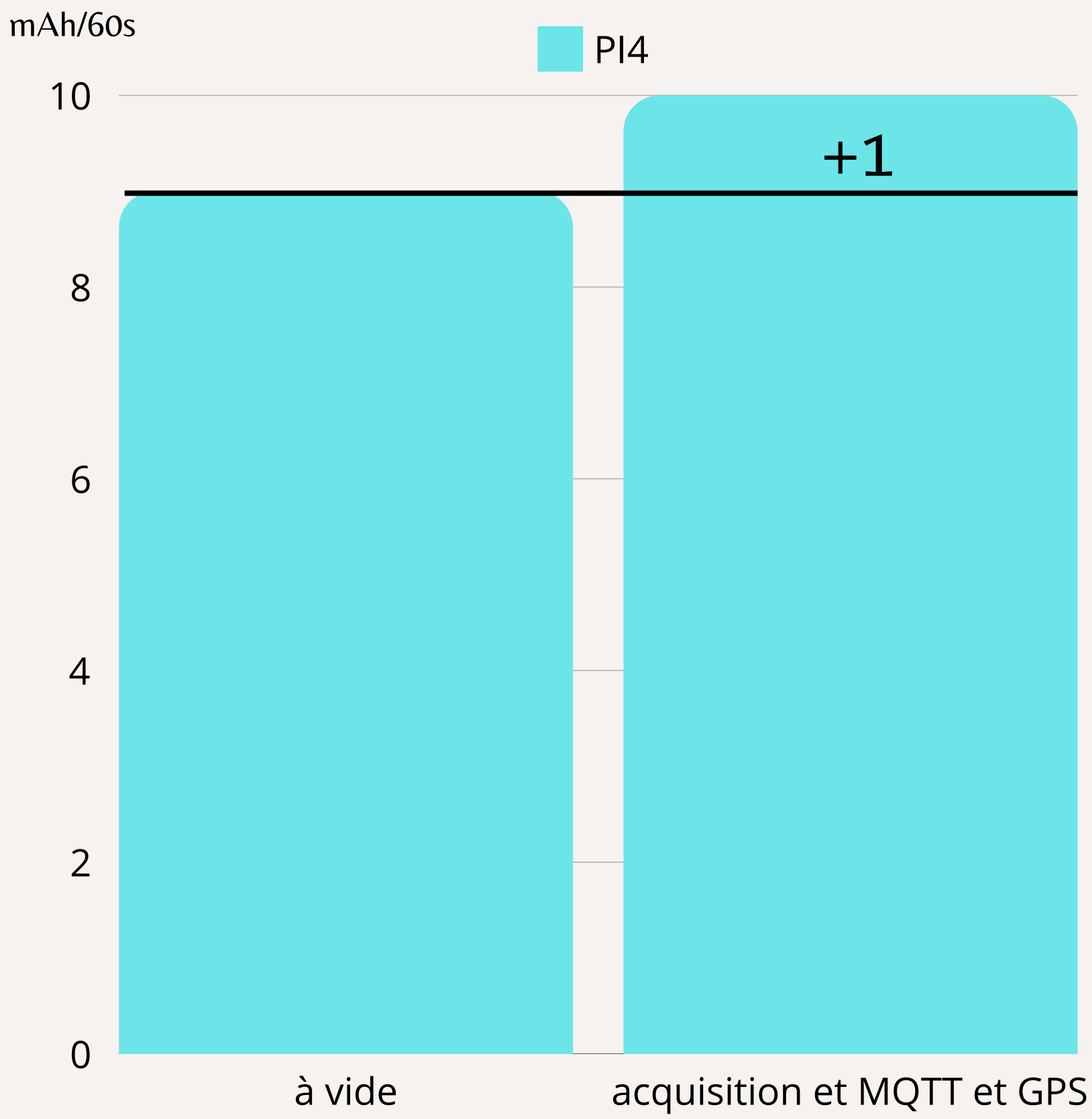

Google coral yolo11n trained full integer quant edgetpu tflite
hailo8l yolov8n trained hef
i7-11800H yolo11n trained pt
i7-11800H yolo11n trained cpu onnx
i7-11800H yolo11n trained openvino model
i7-11800H yolov8n trained onnx
i7-11800H yolov8n trained pt
Raspberry Pi 5 yolo11n trained cpu onnx
Raspberry Pi 5 yolo11n trained ncnn model
RTX 3080 TI laptop 95W yolo11n trained pt
RTX 3080 TI laptop 95W yolo11n trained gpu onnx
Moyenne: 65.68ms
But: 200ms
Comparaison des temps d'inférence
DATASET 4 mAP50 – 95
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
25
50
75
100
125
150
175
200
Temps (ms/img)

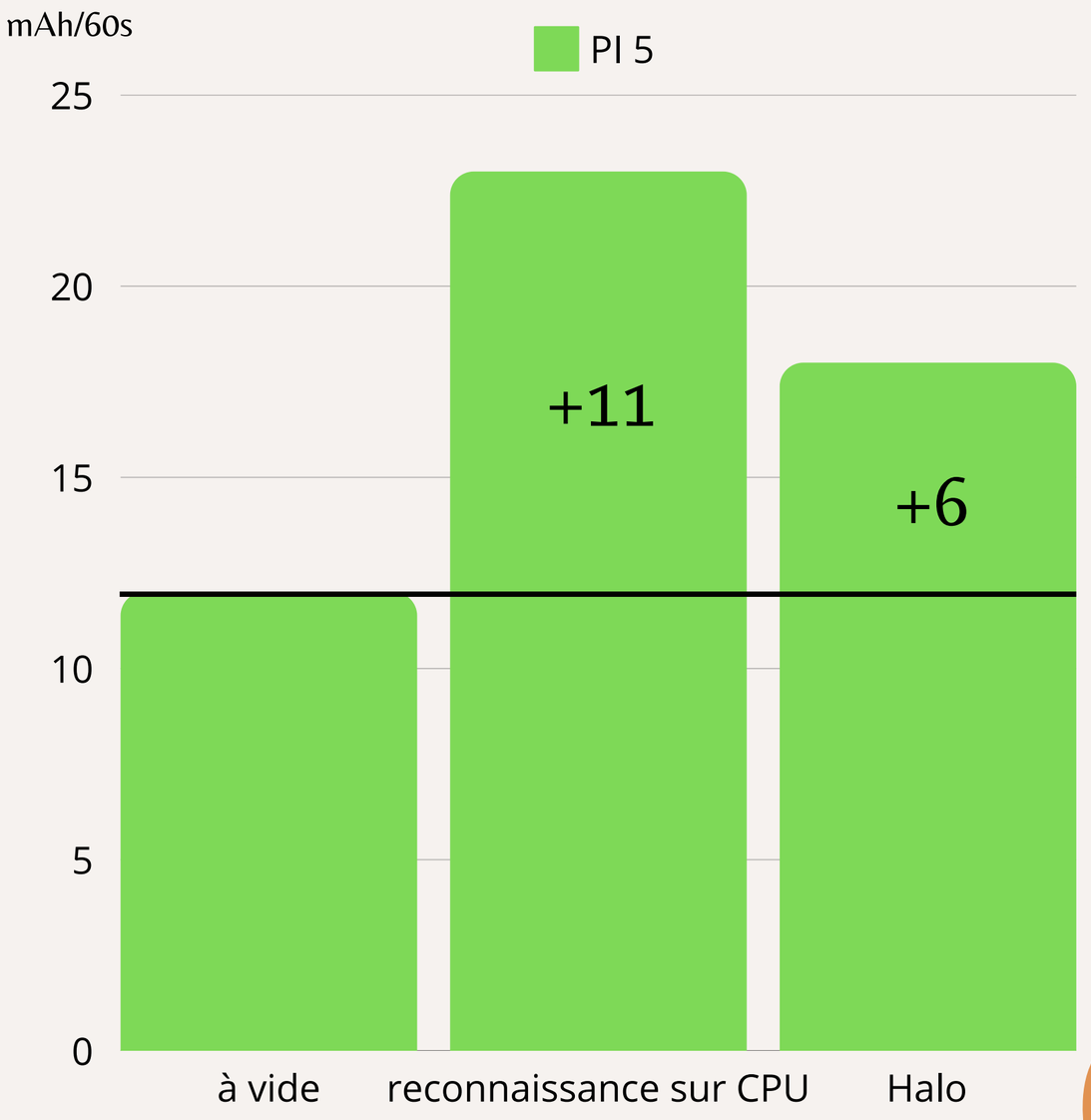

i5-8265U fasterrcnn onnx
i5-8265U fasterrcnn pt
i7-11800H fasterrcnn onnx
i7-11800H fasterrcnn pt
Raspberry Pi 4 8GB fasterrcnn onnx
Raspberry Pi 5 fasterrcnn onnx
RTX 3080 TI laptop 95W fasterrcnn onnx
RTX 3080 TI laptop 95W fasterrcnn pt
Moyenne: 3325.18ms
But: 200ms
Comparaison des temps d'inférence
DATASET 4 mAP50−95
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0
2000
4000
6000
8000
10000
Temps (ms/img)


Comparaison du temps d'inférence moyen pour chaque modèle
Moyenne: 65.68ms
But: 200ms
Temps d'inférence (ms)
0
25
50
75
100
125
150
175
200
Google coral yolo11n trained full integer quant edgetpu tflite
hailo8l yolov8n trained hef
i7-11800H yolo11n trained pt
i7-11800H yolo11n trained cpu onnx
i7-11800H yolo11n trained openvino model
i7-11800H yolov8n trained onnx
i7-11800H yolov8n trained pt
Raspberry Pi 5 yolo11n trained cpu onnx
Raspberry Pi 5 yolo11n trained ncnn model
RTX 3080 TI laptop 95W yolo11n trained pt
RTX 3080 TI laptop 95W yolo11n trained gpu onnx
Fichiers


Comparaison du temps d'inférence moyen pour chaque modèle
Moyenne: 3325.18ms
But: 200ms
Temps d'inférence (ms)
0
2000
4000
6000
8000
10000
i5-8265U fasterrcnn onnx
i5-8265U fasterrcnn pt
i7-11800H fasterrcnn onnx
i7-11800H fasterrcnn pt
Raspberry Pi 4 8GB fasterrcnn onnx
Raspberry Pi 5 fasterrcnn onnx
RTX 3080 TI laptop 95W fasterrcnn onnx
RTX 3080 TI laptop 95W fasterrcnn pt
Fichiers

Consommation énergétique





Difficultés rencontrés

- Data: Image de panneaux
  - Diversité(2 classes à 5 classes)
  - Quantité(~100 à ~780)
- Modèle à choisir
  - ~~classification~~, détection objet
  - vitesse d’exécution <200ms
- Problèmes de compatibilité
  - version non concordantes entre le framework source, onnx et onnx runtime
  - librairies python non compatibles
  - lidar (ancien projet)

Pistes d’améliorations

Avoir 2 modèles différents plus petits :

- 1 pour la détection de panneaux ronds
- 1 pour la classification
- Peuvent fonctionner simultanément

Tester avec un modèle SSD

Avoir plus de données d’entraînement pour une meilleur précision

Lien Github

https://github.com/Colin-de-Seroux/Projet_developpement_logiciel_application_IA_embarquee