



Université de Monastir

L'ÉCOLE POLYTECHNIQUE PRIVÉE DE MONASTIR D'INGÉNIERIE

Détection intelligente de nids-de-poule pour robot autonome

PRÉSENTÉE PAR :ABDKARIM KARDEMINE

4^{ème} année, Ingénierie des Systèmes Embarqués et IoT

ENCADRANT

M. NIZAR HABACHI

Table des matières

1	Contexte et problématique	2
1.1	Présentation de l'entreprise	2
1.1.1	Historique et mission de CRMN	2
1.1.2	Domaines d'activité	2
1.2	Sujet du stage	3
1.3	Problématique	3
1.4	Objectifs de projet	4
2	Matériel et logiciels utilisés	5
2.1	Matériel utilisé	5
2.1.0.1	carte raspberry pi 4	5
2.1.0.2	webcam usb	5
2.1.0.3	Moteur à courant continu et module de commande de moteur	6
2.2	Logiciels utilisés	7
2.2.0.1	Python 3	7
2.2.0.2	colab	8
2.2.0.3	Raspberry Pi OS	8
2.2.0.4	Roboflow	8
3	Réalisation et développement	9
3.1	Aperçu général sur l'IA	9
3.1.1	Bibliographique des algorithmes	9
3.1.1.1	L'intelligence Artificielle	9
3.1.1.2	L'apprentissage automatique (Machine Learning)	9
3.1.1.3	L'apprentissage profond (Deep Learning)	9
3.1.1.4	La vision par ordinateur	10
3.2	Étapes de développement de projet	10
3.2.1	Base de données	10
3.2.2	Annotation avec Roboflow	10
3.2.3	Division des données	10
3.2.4	Entraînement du modèle	11
3.2.4.1	Configuration de l'environnement d'entraînement	11
3.2.4.2	Définition des paramètres d'entraînement	11
3.2.4.3	Lancement de l'entraînement	12
3.2.4.4	Suivi et évaluation intermédiaire	12
3.2.5	Matrice de confusion	13
3.2.6	Analyse de précision et de perte	13

3.2.7	Évaluation du modèle	14
3.2.8	Configuration de la Raspberry Pi 4	14
3.2.8.1	Installation du système d'exploitation	14
3.2.8.2	Configuration du réseau et des accès	15
3.2.8.3	Installation des dépendances logicielles	15
3.2.8.4	Intégration du modèle YOLOv8 dans la carte Raspberry Pi 4	15
3.2.8.5	Validation et test sur le prototype	16

Table des figures

1	les différents type de Machine Learning	2
2	webcam usb	3
3	carte raspberry pi 4	5
4	webcam usb	6
5	Composants utilisés : moteur à courant continu et module de commande L298N	7
6	Division des données	7
7	Division des données	8
8	Division des données	8
9	Division des données	8
10	Division des données	10
11	Division des données	11
12	La précision	12
13	La perte (loss)	13
14	images de test	13
15	Matrice de confusion	14
16	Installation du système d'exploitation	15
17	Configuration du réseau et des accès	15

Résumé

Ce projet a pour objectif de développer un système intelligent de détection de nids-de-poule à l'aide de l'intelligence artificielle. En utilisant le modèle YOLOv8, nous avons conçu une solution capable d'identifier les nids-de-poule sur la route en temps réel à partir d'images capturées par une caméra. Le modèle a été entraîné sur un dataset annoté et optimisé afin d'obtenir une détection rapide et précise.

L'intégration du modèle sur une carte Raspberry Pi 4 permet une exécution embarquée, sans dépendance à un ordinateur externe. Le système pilote ensuite un robot mobile équipé d'un moteur à courant continu contrôlé par un module L298N, simulant ainsi le comportement d'un véhicule intelligent capable d'adapter sa vitesse ou sa trajectoire selon l'état de la route.

Ce travail nous a permis de maîtriser plusieurs aspects techniques, tels que la collecte de données, l'entraînement d'un modèle de Deep Learning, et l'intégration de l'IA dans un système embarqué, tout en contribuant à une solution innovante pour la sécurité routière..

Abstract

This project aims to develop an intelligent pothole detection system using artificial intelligence. Based on the YOLOv8 model, the proposed solution can identify road potholes in real time from images captured by a camera. The model was trained on an annotated dataset and optimized to achieve fast and accurate detection.

The integration of the trained model into a Raspberry Pi 4 board allows for embedded execution without relying on an external computer. The system then controls a mobile robot equipped with a DC motor driven by an L298N module, simulating the behavior of an intelligent vehicle capable of adjusting its speed or direction according to road conditions.

This work enabled us to acquire practical experience in data collection, Deep Learning model training, and AI deployment on embedded systems, contributing to an innovative approach to road safety improvement.

Introduction Générale

Aujourd'hui, l'intelligence artificielle (IA) joue un rôle important dans de nombreux domaines, notamment dans la robotique et les systèmes intelligents. Grâce à l'apprentissage automatique et à la vision par ordinateur, il est possible de créer des machines capables d'analyser leur environnement et de prendre des décisions sans intervention humaine.

Dans le cadre de notre stage, nous avons travaillé sur un projet visant à détecter les nids-de-poule sur la route à l'aide d'un modèle d'IA basé sur YOLOv8. L'objectif principal est de concevoir un système embarqué intelligent capable d'identifier automatiquement les nids-de-poule et d'adapter la vitesse ou la trajectoire d'un robot mobile pour les éviter.

Ce projet s'inscrit dans la continuité des recherches menées au Centre de Recherche en Microélectronique et Nanotechnologie (CRMN), et combine à la fois des aspects matériels et logiciels. Nous avons ainsi développé un modèle d'intelligence artificielle entraîné à partir d'un dataset annoté, puis intégré ce modèle dans une carte Raspberry Pi 4, reliée à un moteur à courant continu et à un module de commande L298N.

Ce travail nous a permis de découvrir toutes les étapes de la conception d'un système intelligent : de la collecte des données à l'entraînement du modèle, jusqu'à son intégration dans un prototype fonctionnel.

Chapitre 1

Contexte et problématique

Ce chapitre expose le contexte général du projet, les besoins identifiés ainsi que la problématique à laquelle ce stage tente d'apporter une solution.

1.1 Présentation de l'entreprise

1.1.1 Historique et mission de CRMN

Le Centre de Recherche en Microélectronique et Nanotechnologie (CRMN) de Sousse, créé en 2012 et installé au Technopôle de Sousse, est un établissement public scientifique et technologique rattaché au ministère de l'Enseignement supérieur et de la Recherche scientifique. Sa mission principale est de promouvoir la recherche et l'innovation dans les domaines de la microélectronique, des nanotechnologies et des microsystèmes, en développant des technologies avancées pour des applications stratégiques telles que la santé, l'environnement, l'agriculture de précision et la sécurité. Le CRMN participe également à des projets collaboratifs internationaux, notamment dans le cadre du programme européen Horizon 2020, et s'implique dans des projets phares comme le programme « Pacte » qui vise à développer des nanocapteurs intelligents, des zones multi-capteurs sécurisées et des systèmes d'information géographique pour la sécurité nationale, mobilisant une équipe d'environ 70 chercheurs.



FIGURE 1 – les différents type de Machine Learning

1.1.2 Domaines d'activité

Le Centre de Recherche en Microélectronique et Nanotechnologie (CRMN) intervient dans plusieurs domaines d'activité, notamment :

Microélectronique : conception et fabrication de circuits intégrés et de dispositifs électroniques avancés.

Nanotechnologies : développement de matériaux et de systèmes à l'échelle nanométrique pour des applications variées.

Microsystèmes : création de capteurs intelligents, MEMS (Micro-Electro-Mechanical Systems) et systèmes embarqués.

Applications stratégiques : santé, environnement, agriculture de précision, sécurité et ingénierie.

Projets collaboratifs et recherche appliquée : participation à des programmes internationaux comme Horizon 2020, et développement de solutions innovantes pour la sécurité nationale et industrielle.

1.2 Sujet du stage

Le sujet de ce stage porte sur le développement d'un système ADAS (Advanced Driver Assistance System) intelligent basé sur l'Intelligence Artificielle, dont l'objectif est de détecter automatiquement les nids-de-poule sur la route et de contrôler la vitesse du véhicule en conséquence. Ce projet vise à améliorer la sécurité et le confort de conduite en utilisant des modèles de vision par ordinateur pour identifier les dégradations de la chaussée, tout en intégrant un mécanisme d'adaptation de la vitesse afin de réduire les risques liés aux obstacles. L'approche proposée combine ainsi l'IA, la perception embarquée et l'automatisation du contrôle, contribuant à l'évolution des systèmes de conduite assistée et à la prévention des accidents.

1.3 Problématique

Malgré les avancées des systèmes ADAS, la détection et la gestion des dégradations routières, telles que les nids-de-poule, restent encore limitées. Ces anomalies peuvent provoquer des dommages matériels aux véhicules, réduire le confort de conduite et surtout représenter un risque pour la sécurité routière. La question centrale est donc : comment concevoir un système intelligent capable de détecter en temps réel les nids-de-poule à l'aide de l'IA et d'adapter automatiquement la vitesse du véhicule afin de prévenir les accidents et d'améliorer la sécurité des usagers de la route ?

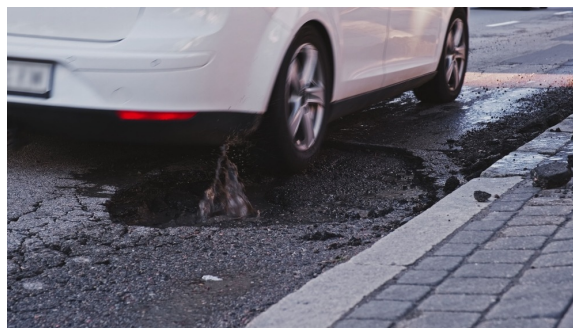


FIGURE 2 – webcam usb

1.4 Objectifs de projet

Les objectifs de ce stage consistent tout d'abord à réaliser une étude de l'état de l'art sur les systèmes ADAS existants ainsi que sur les approches d'intelligence artificielle appliquées à la détection des anomalies routières. Par la suite, il s'agira de constituer et de préparer un jeu de données d'images et de vidéos de routes contenant des nids-de-poule, en procédant à un étiquetage précis. L'objectif majeur est de développer et d'entraîner un modèle d'IA, tel que YOLOv8, capable de détecter en temps réel les nids-de-poule. Ce modèle sera ensuite intégré sur une plateforme embarquée, qu'il s'agisse d'un robot, d'un prototype de véhicule ou d'un microcontrôleur connecté à une caméra. En parallèle, un module de contrôle de vitesse sera conçu afin de permettre au système de réagir automatiquement à la détection d'un nid-de-poule. Enfin, le stage vise à tester et évaluer la performance du système en conditions simulées et/ou réelles, à analyser les résultats obtenus et à proposer des pistes d'amélioration en vue d'une intégration dans des systèmes ADAS plus avancés.

Chapitre 2

Matériel et logiciels utilisés

Ce chapitre présente l'ensemble du matériel et des logiciels que nous avons utilisés pour la conception, le développement et la mise en œuvre de notre système ADAS de détection des nids-de-poule et de contrôle de vitesse. Ces outils ont joué un rôle essentiel dans l'acquisition des données, l'entraînement du modèle d'intelligence artificielle et l'intégration du système sur la plateforme embarquée

2.1 Matériel utilisé

2.1.0.1 carte raspberry pi 4

Nous avons choisi d'utiliser la carte Raspberry Pi 4 comme unité de traitement principale dans notre projet, en raison de ses performances élevées et de sa compatibilité avec les applications d'intelligence artificielle. Cette carte offre un processeur puissant, une mémoire suffisante et plusieurs interfaces de connexion (USB, HDMI, GPIO, caméra), ce qui nous a permis d'intégrer facilement la caméra embarquée et les différents capteurs nécessaires à notre système. Grâce à sa taille compacte et à sa faible consommation d'énergie, la Raspberry Pi 4 convient parfaitement à une application embarquée comme la détection des nids-de-poule en temps réel. De plus, elle prend en charge les environnements de développement Python et les bibliothèques IA comme OpenCV et YOLOv8, ce qui nous a facilité la mise en œuvre et le test de notre modèle.

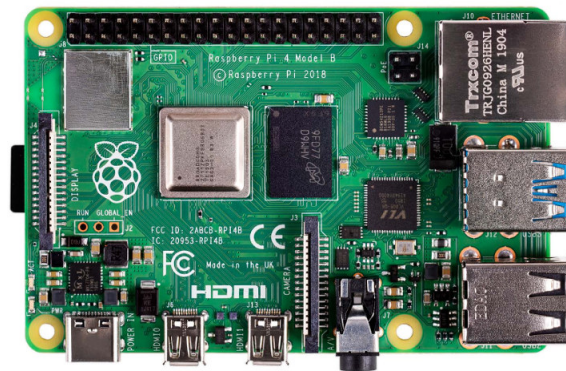


FIGURE 3 – carte raspberry pi 4

2.1.0.2 webcam usb

Nous avons choisi d'utiliser une webcam USB pour la capture d'images et de vidéos de la route, car elle représente une solution simple, économique et compatible avec la carte Raspberry Pi 4. Ce choix nous a permis d'acquérir des données visuelles en temps réel avec une résolution suffisante pour la détection précise des nids-de-poule. La webcam USB est facile à configurer et à intégrer dans notre

système grâce à sa compatibilité avec les bibliothèques de traitement d'images comme OpenCV. De plus, elle offre une bonne stabilité du flux vidéo et une connectivité plug-and-play, ce qui nous a permis de nous concentrer davantage sur le développement et l'entraînement du modèle d'intelligence artificielle plutôt que sur des contraintes matérielles complexes



FIGURE 4 – webcam usb

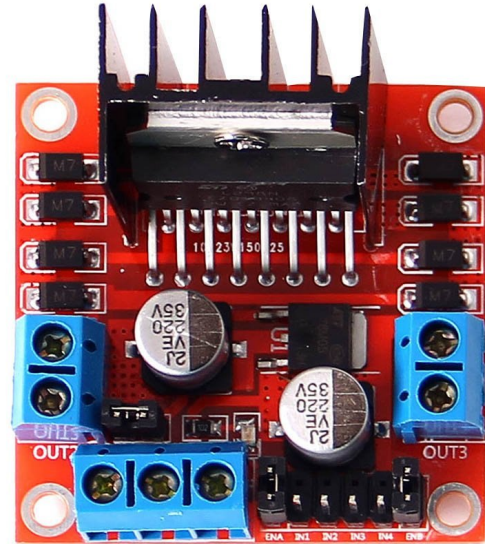
2.1.0.3 Moteur à courant continu et module de commande de moteur

Nous avons choisi d'utiliser un moteur à courant continu (DC motor) pour notre prototype, car il offre une solution simple, fiable et facile à contrôler pour le déplacement du système. Ce type de moteur nous permet de réguler la vitesse de rotation en fonction des signaux envoyés depuis la carte **Raspberry Pi 4**, ce qui est essentiel pour notre projet de contrôle automatique de la vitesse en présence de nids-de-poule.

De plus, le moteur DC est léger, peu coûteux et compatible avec les modules de commande tels que le **L298N**, ce qui nous a permis d'assurer une intégration stable et efficace dans notre système embarqué. Grâce à cette configuration, nous avons pu simuler le comportement d'un véhicule intelligent capable d'ajuster sa vitesse selon les conditions de la route.



(a) Moteur à courant continu



(b) Module L298N

FIGURE 5 – Composants utilisés : moteur à courant continu et module de commande L298N

2.2 Logiciels utilisés

Pour la réalisation de notre projet, nous avons utilisé plusieurs logiciels et environnements de développement, chacun jouant un rôle précis dans le traitement des données, l'entraînement du modèle et le contrôle du robot :

2.2.0.1 Python 3

langage principal pour le développement des scripts de traitement d'images, d'entraînement du modèle et de contrôle du robot.



FIGURE 6 – Division des données

2.2.0.2 colab

environnements de développement utilisés pour coder, tester et visualiser les étapes d'entraînement et de validation du modèle.



FIGURE 7 – Division des données

2.2.0.3 Raspberry Pi OS

système d'exploitation de la carte Raspberry Pi 4, nécessaire pour exécuter le modèle YOLOv8 et piloter le robot.

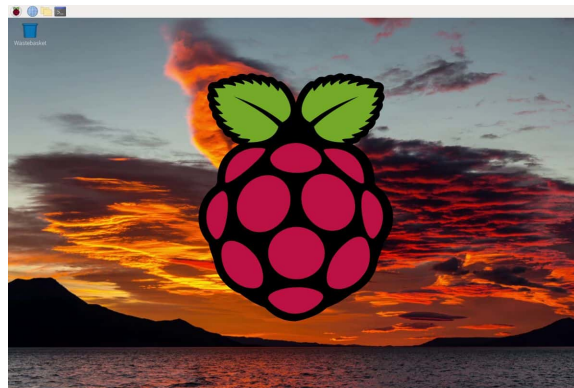


FIGURE 8 – Division des données

2.2.0.4 Roboflow

outils pour l'annotation et la préparation du dataset utilisé pour entraîner le modèle.



FIGURE 9 – Division des données

Chapitre 3

Réalisation et développement

Ce chapitre présente les différentes étapes que nous avons suivies pour la réalisation et le développement de notre système de détection des nids-de-poule et de contrôle de vitesse, depuis la conception du modèle d'intelligence artificielle jusqu'à son intégration sur la plateforme embarquée.

3.1 Aperçu général sur l'IA

L'intelligence artificielle (IA) est une branche de l'informatique qui cherche à reproduire certaines capacités humaines comme l'apprentissage, la perception et la prise de décision. Grâce à des algorithmes capables d'analyser et de traiter de grandes quantités de données, l'IA permet de résoudre des problèmes complexes et d'automatiser des tâches. Aujourd'hui, elle est largement utilisée dans divers domaines tels que la santé, la sécurité, l'industrie et les transports, où elle joue un rôle essentiel dans le développement des véhicules intelligents et des systèmes d'assistance à la conduite (ADAS).

3.1.1 Bibliographique des algorithmes

3.1.1.1 L'intelligence Artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique qui cherche à reproduire certaines capacités humaines telles que la perception, l'apprentissage et la prise de décision. Elle comprend plusieurs approches et techniques permettant aux machines d'analyser des données, de détecter des motifs et de prendre des décisions de manière autonome. L'IA est largement utilisée dans les systèmes ADAS pour améliorer la sécurité et l'efficacité de la conduite.

3.1.1.2 L'apprentissage automatique (Machine Learning)

Le machine learning est une sous-branche de l'IA qui permet aux systèmes d'apprendre à partir des données sans être explicitement programmés. Les algorithmes de machine learning peuvent être supervisés, non supervisés ou par renforcement, et sont utilisés pour la reconnaissance de motifs, la classification d'objets et la prédiction de comportements dans les véhicules intelligents.

3.1.1.3 L'apprentissage profond (Deep Learning)

Le deep learning utilise des réseaux de neurones artificiels multicouches capables de traiter des données complexes comme les images ou les vidéos. Cette approche est particulièrement efficace pour la détection d'objets en temps réel, ce qui la rend adaptée aux systèmes ADAS pour identifier les nids-de-poule ou autres obstacles sur la route.

3.1.1.4 La vision par ordinateur

La vision par ordinateur est une discipline qui permet aux machines de “voir” et d’interpréter des images ou des vidéos. Elle combine des techniques de traitement d’image et d’apprentissage automatique pour détecter et analyser les objets, les obstacles et les anomalies routières, et est donc essentielle pour la détection des nids-de-poule dans les systèmes ADAS.

3.2 Étapes de développement de projet

3.2.1 Base de données

La qualité des données joue un rôle crucial dans le succès des projets d’apprentissage automatique. Pour notre projet, nous avons utilisé des ensembles de données open-source contenant des images de routes avec différents types de nids-de-poule. Nous avons ensuite complété ces données par nos propres captures afin d’enrichir le jeu de données et de garantir une meilleure diversité des situations (différents angles, conditions d’éclairage et types de routes). Cette préparation nous a permis d’entraîner efficacement notre modèle YOLOv8 et d’améliorer sa précision pour la détection en temps réel des nids-de-poule.

3.2.2 Annotation avec Roboflow

Nous avons utilisé Roboflow pour annoter précisément les nids-de-poule sur chaque image, ce qui a permis de créer un jeu de données structuré et exploitable par le modèle YOLOv8.

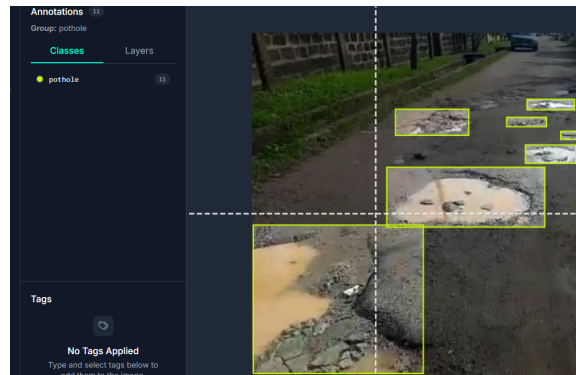


FIGURE 10 – Division des données

3.2.3 Division des données

Nous avons scindé nos données en trois sous-ensembles distincts. Le premier, appelé ensemble d’entraînement, a été utilisé pour ajuster les paramètres du modèle. Le deuxième, appelé ensemble de validation, nous a permis de suivre les performances du modèle pendant l’entraînement et d’ajuster les hyperparamètres afin d’éviter le surapprentissage. Enfin, le troisième, appelé ensemble de test, a servi à évaluer les performances finales du modèle sur des données jamais vues, garantissant ainsi que

le modèle YOLOv8 est capable de généraliser efficacement à de nouvelles images de routes contenant des nids-de-poule

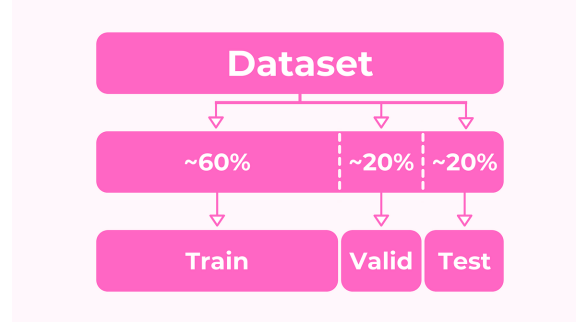


FIGURE 11 – Division des données

3.2.4 Entraînement du modèle

Après avoir préparé et annoté nos données, nous avons commencé l'entraînement du modèle YOLOv8. Cette étape est cruciale, car c'est à ce moment que le modèle apprend à reconnaître les nids-de-poule dans les images

3.2.4.1 Configuration de l'environnement d'entraînement

Nous avons utilisé un ordinateur avec un GPU pour accélérer le processus d'apprentissage. Nous avons installé toutes les bibliothèques nécessaires, y compris PyTorch et Ultralytics YOLOv8, afin de pouvoir lancer l'entraînement efficacement.

3.2.4.2 Définition des paramètres d'entraînement

Nous avons configuré plusieurs paramètres importants :

a) Nombre d'époques

Nous avons choisi d'entraîner le modèle sur 50 époques, c'est-à-dire que le modèle parcourt l'ensemble de notre dataset 50 fois. Ce nombre nous a permis de donner suffisamment d'opportunités au modèle pour apprendre les caractéristiques des nids-de-poule, tout en évitant un sur-apprentissage.

b) Taille des images

Nous avons fixé la taille des images à 640x640 pixels. Cela nous a permis de conserver assez de détails pour détecter correctement les nids-de-poule tout en maintenant une vitesse d'entraînement raisonnable.

c) Batch size

le nombre d'images traitées simultanément par le modèle.

3.2.4.3 Lancement de l'entraînement

Nous avons exécuté l'entraînement en utilisant l'ensemble d'images d'entraînement. Le modèle a progressivement appris à reconnaître les caractéristiques des nids-de-poule à partir des annotations que nous avons faites.

3.2.4.4 Suivi et évaluation intermédiaire

Pendant l'entraînement, nous avons surveillé plusieurs indicateurs :

a) La précision

nous avons vérifié combien de fois le modèle prédit correctement la présence d'un nid-de-poule dans les images. Cela nous a permis de savoir si le modèle devenait de plus en plus fiable au fil des époques.

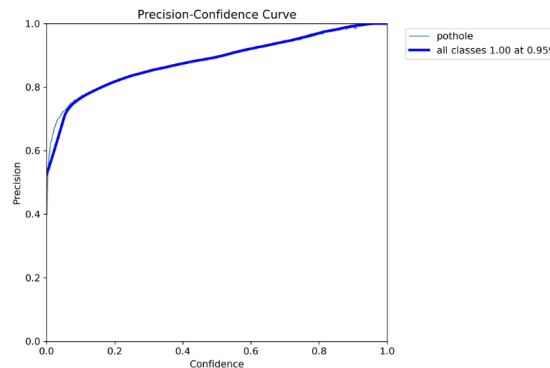


FIGURE 12 – La précision

b) La perte (loss)

nous avons surveillé la valeur de la perte, qui mesure l'écart entre les prédictions du modèle et les annotations réelles. Une diminution progressive de la perte indiquait que le modèle apprenait efficacement.

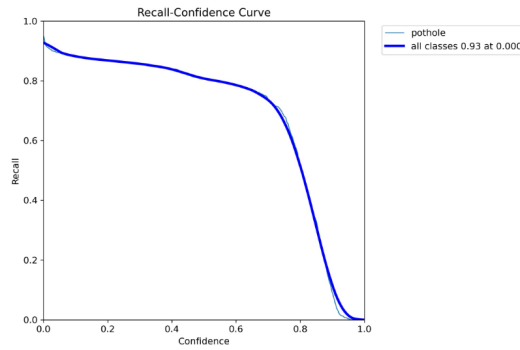


FIGURE 13 – La perte (loss)

c) images de test

nous avons testé régulièrement le modèle sur des images qu'il n'avait jamais vues afin de vérifier sa capacité à détecter correctement les nids-de-poule dans des situations nouvelles.



FIGURE 14 – images de test

3.2.5 Matrice de confusion

Pour évaluer la performance de notre modèle YOLOv8, nous avons utilisé une matrice de confusion. Cette matrice nous permet de comparer les prédictions du modèle avec les véritables étiquettes (ground truth).

- **Vrais positifs (VP)** : représentent les nids-de-poule correctement détectés.
- **Faux positifs (FP)** : correspondent à des détections erronées, où le modèle a identifié un objet qui n'était pas un nid-de-poule.
- **Faux négatifs (FN)** : représentent les nids-de-poule que le modèle n'a pas détectés.

En analysant la matrice de confusion, nous avons pu identifier les types d'erreurs les plus fréquentes et déterminer les points à améliorer dans notre dataset ou l'entraînement du modèle.

3.2.6 Analyse de précision et de perte

Pendant l'entraînement du modèle YOLOv8, nous avons suivi deux indicateurs principaux pour évaluer ses performances. La précision nous a permis de savoir combien de fois le modèle prédisait correctement la présence d'un nid-de-poule parmi toutes ses prédictions. En parallèle, nous avons sur-

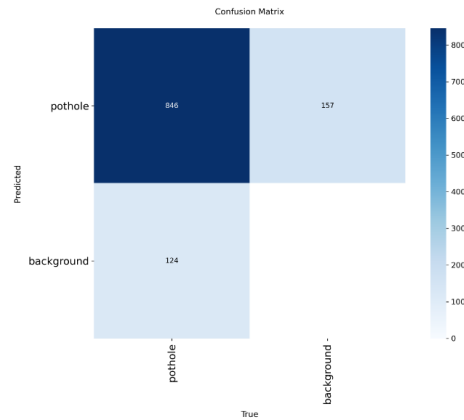


FIGURE 15 – Matrice de confusion

veillé la perte (loss), qui mesure l'écart entre les prédictions du modèle et les annotations réelles. Une diminution régulière de cette perte au fil des époques indiquait que le modèle apprenait efficacement. Cette analyse nous a également permis d'ajuster certains paramètres d'entraînement, comme le taux d'apprentissage ou la taille des images, afin d'améliorer les résultats et de garantir un apprentissage stable et fiable.

3.2.7 Évaluation du modèle

Pour valider les performances de YOLOv8, nous avons testé le modèle sur l'ensemble test, c'est-à-dire sur des images qu'il n'avait jamais vues pendant l'entraînement. Nous avons comparé les prédictions aux annotations réelles et calculé différents indicateurs, tels que la précision, le rappel et le score F1. En parallèle, nous avons visualisé les résultats directement sur les images, avec les boîtes englobantes et les scores de confiance, afin de vérifier si le modèle détectait correctement les nids-de-poule. Cette évaluation nous a permis de confirmer que le modèle était capable de fonctionner efficacement dans des conditions variées et de préparer son intégration dans le prototype embarqué.

3.2.8 Configuration de la Raspberry Pi 4

3.2.8.1 Installation du système d'exploitation

Nous avons commencé par installer le système d'exploitation **Raspberry Pi OS** sur une carte microSD à l'aide du logiciel **Raspberry Pi Imager**. Après le démarrage de la carte, nous avons effectué les configurations de base : langue, réseau Wi-Fi, et mise à jour du système. Cette étape est essentielle pour garantir la stabilité et la compatibilité avec les bibliothèques nécessaires à notre projet.

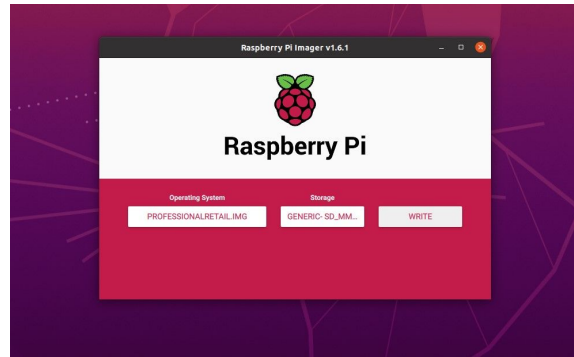


FIGURE 16 – Installation du système d’exploitation

3.2.8.2 Configuration du réseau et des accès

Nous avons configuré la connexion Wi-Fi et activé l’accès SSH afin de pouvoir contrôler la Raspberry Pi 4 à distance depuis notre ordinateur. Cela nous a permis d’exécuter les scripts Python et de surveiller le fonctionnement du système sans avoir besoin d’un écran ou d’un clavier connecté directement à la carte.

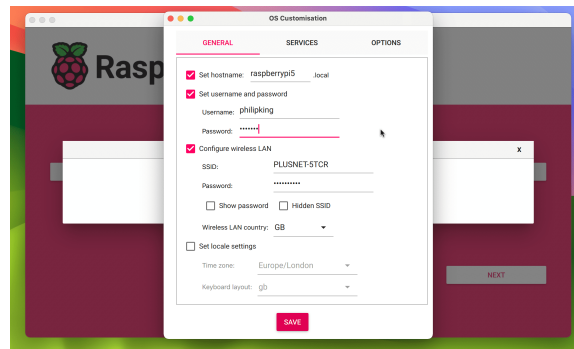


FIGURE 17 – Configuration du réseau et des accès

3.2.8.3 Installation des dépendances logicielles

Après la configuration réseau, nous avons installé les bibliothèques nécessaires à l’exécution du modèle YOLOv8. Parmi elles, Python 3 pour le développement du projet, OpenCV pour le traitement d’images et la capture vidéo, et Ultralytics YOLOv8 pour le chargement et l’exécution du modèle d’intelligence artificielle. Ces outils ont permis à la Raspberry Pi 4 d’assurer le traitement des images et la détection des nids-de-poule en temps réel.

3.2.8.4 Intégration du modèle YOLOv8 dans la carte Raspberry Pi 4

Après la configuration complète de la Raspberry Pi 4, nous avons procédé à l’intégration du modèle YOLOv8 dans le système embarqué. Cette étape visait à permettre au robot de détecter les nids-de-poule en temps réel et de réagir automatiquement selon la situation détectée.

3.2.8.5 Validation et test sur le prototype

Enfin, des tests ont été réalisés sur le prototype afin de vérifier la réactivité du système. Lors des essais, le robot a correctement détecté les nids-de-poule simulés et ajusté sa trajectoire pour les éviter. Ce résultat confirme la bonne intégration entre la détection intelligente et le contrôle physique du robot, démontrant le bon fonctionnement de l'ensemble du système embarqué.

Conclusion Générale

L'intelligence artificielle (IA) occupe aujourd'hui une place essentielle dans le développement des systèmes intelligents et autonomes. Grâce aux progrès rapides du Deep Learning et de la vision par ordinateur, il est désormais possible de concevoir des machines capables de percevoir, d'analyser et d'interagir avec leur environnement de manière efficace. Ces avancées ont ouvert la voie à de nombreuses applications, notamment dans les domaines de la sécurité routière, de la robotique et des systèmes d'aide à la conduite (ADAS).

Dans ce contexte, notre projet s'inscrit dans la volonté d'exploiter les capacités de l'IA pour détecter les nids-de-poule sur la route, un problème réel qui affecte la sécurité des usagers et la durabilité des véhicules. L'idée consiste à concevoir un robot intelligent équipé d'une caméra et d'un modèle de détection basé sur YOLOv8, capable d'identifier les nids-de-poule en temps réel et de réagir automatiquement en ajustant sa trajectoire pour les éviter.

Le projet a été réalisé au sein du Centre de Recherche en Microélectronique et Nanotechnologie (CRMN), où nous avons pu combiner des compétences en programmation embarquée, traitement d'images et apprentissage automatique. La démarche adoptée comprend plusieurs étapes : la collecte et le traitement des données, l'entraînement du modèle YOLOv8, puis l'intégration du système dans une carte Raspberry Pi pour piloter un robot mobile.

Ce travail nous a permis d'approfondir notre compréhension des technologies d'intelligence artificielle appliquées à la vision par ordinateur, tout en développant une solution innovante et concrète pour la détection automatique des dégradations routières.