



Mise en garde

La bibliothèque du Cégep de l'Abitibi-Témiscamingue et de l'Université du Québec en Abitibi-Témiscamingue (UQAT) a obtenu l'autorisation de la personne autrice de ce document afin de diffuser, dans un but non lucratif, une copie de son œuvre dans [Depositum](#), site d'archives numériques, gratuit et accessible à tous. L'autrice ou l'auteur conserve néanmoins ses droits de propriété intellectuelle, dont son droit d'auteur, sur cette œuvre.

Warning

The library of the Cégep de l'Abitibi-Témiscamingue and the Université du Québec en Abitibi-Témiscamingue (UQAT) obtained the permission of the author to use a copy of this document for nonprofit purposes in order to put it in the open archives [Depositum](#), which is free and accessible to all. The author retains ownership of the copyright on this document.

UNIVERSITÉ DU QUÉBEC EN ABITIBI-TÉMISCAMINGUE

CONCEPTION D'UN SYSTÈME INTELLIGENT BASÉ SUR IoT
ET IA

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INGÉNIERIE

PAR
KHALDI OUADJIH

Novembre 2025

Remerciements

Avant de débiter ce rapport, je tiens à exprimer ma sincère gratitude et ma profonde reconnaissance envers toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail. Ce mémoire représente l'aboutissement d'un long parcours, rendu possible grâce au soutien, à l'encouragement et à l'implication de nombreuses personnes que je souhaite ici remercier chaleureusement.

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de recherche **Pr Fouad Slaoui Hasnaoui** pour son encadrement rigoureux, sa disponibilité constante, ses conseils avisés et sa bienveillance tout au long de ce projet. Son expertise et sa confiance m'ont permis d'évoluer tant sur le plan scientifique que personnel. Son accompagnement a été un pilier essentiel dans la réussite de ce travail.

Mes remerciements s'adressent également à **ma famille**, et en particulier à **mes parents**, pour leur amour inconditionnel, leur patience, leur présence rassurante et leur soutien moral. Leur confiance en moi a été une source de motivation constante, et leur soutien m'a permis de surmonter les moments les plus exigeants de ce parcours.

Je remercie également mes **amis, collègues et camarades de promotion**, avec qui j'ai partagé de nombreux moments d'entraide, de réflexion, mais aussi de convivialité. Leur esprit de coopération, leur bonne humeur et leur solidarité ont grandement enrichi mon expérience personnelle et académique.

Je n'oublie pas non plus les **personnes rencontrées au cours de ce projet**, qu'il s'agisse de techniciens, d'ingénieurs, de personnels administratifs ou de professionnels du domaine, qui ont, chacun à leur manière, apporté leur aide, leurs conseils ou leur soutien.

Enfin, je remercie **toutes les personnes qui, directement ou indirectement**, ont contribué à la réalisation de ce mémoire. À chacun d'entre vous, je dis avec reconnaissance et humilité : **merci**.

Dédicace

Je dédie ce travail à mes parents,

pour tout l'amour, la patience et les encouragements qu'ils m'ont donnés depuis le début. Merci de toujours croire en moi, même quand moi je doutais.

À mes frères et sœurs,

pour leur soutien, leur présence et tous les moments partagés qui m'ont aidé à garder le moral.

À mes amis,

pour leur bonne humeur, leurs conseils, et leur motivation quand il le fallait. Vous avez rendu ce parcours plus léger.

À mes professeurs,

pour leur accompagnement, leur disponibilité et tout ce qu'ils m'ont transmis durant ce projet.

Et enfin, à vous qui lisez ce mémoire,

en espérant que ce travail saura vous apporter quelque chose d'utile ou d'inspirant.

Table des matières

Remerciements	I
Dédicace	II
Liste des figures	VII
Liste des tableaux	X
LISTE DES ABRÉVIATIONS	XI
Résume	XII
Abstract.....	XIV
État de l’art.....	1
Chapitre 1	6
Introduction générale	6
1.1. Contexte général.....	7
1.2. Problématique.....	8
1.3. Originalités	8
1.4. Les Objectifs	8
1.5. Les hypothèses	9
1.6. Méthodologie	9
Chapitre 2	10
Aperçu sur l’IoT, les Systèmes Embarqués, et la Détection des Anomalies	
Agricole	10
2.1. Introduction	10
2.2. L’Internet des Objets en Agriculture Intelligente	10
2.2.1. Définition et rôle de l’IoT en agriculture	10
2.2.2. Architecture d’un système IoT agricole	11
2.3. Les Systèmes Embarqués : Fondations de l’Agriculture Intelligente	11

2.3.1. Définition et caractéristiques générales	11
2.3.2. Pourquoi utiliser des systèmes embarqués en agriculture intelligente ..	12
2.4. Détection et Prédiction d'Anomalies : Un Outil Clé pour l'Optimisation Agricole	13
2.4.1. Concepts généraux.....	13
2.4.2. Techniques modernes de détection d'anomalies adaptées aux systèmes embarqués	13
2.5. Importance de l'IA embarquée pour la prédiction en agriculture	14
2.6. Défis et Perspectives	14
2.7. Conclusion.....	15
Chapitre 3	16
Conception et mise en œuvre du système embarqué.....	16
3.1. Introduction	16
3.2. Architecture proposée du système embarqué	17
3.2.1. Choix technologique et justification.....	17
3.3. Programmation description fonctionnelle du système	26
3.3.1. Cycle de fonctionnement	27
3.4. Conclusion.....	29
Chapitre 4	30
Mise en réseau et communication IoT pour une supervision intelligente.....	30
4.1. Introduction	30
4.2. Cadre théorique et concepts clés	31
4.2.1. Supervision et commande distribuée.....	31
4.2.2. Protocoles de communication.....	31
4.2.3. Plateformes de visualisation des données.....	32

4.2.4. Interfaces utilisateurs dans l'IoT	32
4.3. Architecture de communication du système IoT proposé	32
4.3.1. Schéma global de l'architecture	33
4.3.2. Communication bidirectionnelle	34
4.3.3. Synchronisation et priorités	34
4.3.4. Choix technologiques et justifications.....	35
4.4. Intégration avec la plateforme cloud (ThingSpeak)	35
4.4.1. Résultats de l'intégration en temps réel.....	37
4.4.2. Interprétation d'acquisition des données temps réel.....	38
4.4.3. Conclusion global et Portée des résultats	44
4.5. Développement de l'interface Web embarquée	45
4.5.1. Environnement de développement	45
4.5.2. Conception de l'interface : légèreté et réactivité	48
4.5.3. Conclusion	51
4.6. Supervision et commande locale avec LabVIEW	52
4.6.1. Programmation graphique et Conception du bloc diagramme	52
4.6.2. Développement de l'interface utilisateur (Front Panel)	57
4.6.3. Téléversement et visualisation des résultats	58
4.6.4. Conclusion	59
Chapitre 5	60
Intégration de l'IA pour la détection des anomalies	60
5.1. Introduction	60
5.2. Cadre Méthodologique Général	61
5.2.1. Base des données	61
5.2.2. Augmentation des données	61

5.2.3. Réglage des hyperparamètres	61
5.2.4. Évaluation des modèles	62
5.3. Détection de la qualité du sol à l'aide du model yolo v11	62
5.3.1. Spécificité du Dataset	63
5.3.2. Résultats et discussions	64
5.3.3. Interprétation des résultats.....	65
5.4. Détection des fuites des tuyaux à l'aide du model yolo v11	68
5.4.1. Spécificité du Dataset	68
5.4.2. Résultats et discussions	69
5.4.3. Interprétation des résultats.....	70
5.5. Détection de maladie des plantes l'aide de la version YOLOv12.....	73
5.5.1. Spécificité du Dataset	74
5.5.2. Résultats et discussions	75
5.5.3. Interprétation des résultats.....	76
5.6. Mise en œuvre des modèles entraînés	79
5.6.1. Déploiement du model	81
5.6.2. Affichage des résultats.....	84
5.7. Conclusion.....	87
Comparaison des résultats avec des travaux récente.....	88
Conclusion.....	89
Références	92

Liste des figures

Figure 1 : Architecture du esp32	18
Figure 2 : Esp32 réel	18
Figure 3 : Schéma de circuit du module DHT11 [40].....	20
Figure 4 : ESP32 interfacé avec DHT11	20
Figure 5 : Schéma de circuit du module capteur capacitif d’humidité du sol	21
Figure 6 : ESP32 interfacé avec capteur capacitif d’humidité du sol	22
Figure 7 : ESP32 interfacé avec capteur ST-045.....	23
Figure 8 : ESP32 interfacé avec capteur YF-S201	24
Figure 9 : Schéma global – Intégration de tous les capteurs.....	25
Figure 10 : Schéma global – partie actionneurs	26
Figure 11 : résultat réel de sortie en temps réel sur le moniteur série.....	29
Figure 12 : Architecture de communication proposé	33
Figure 13 : Résultat d’acquisition en ligne en temps réel	38
Figure 14 : Résultat d’acquisition en temps réel d’humidité du sol.....	39
Figure 15 : Résultat d’acquisition en temps réel de niveau d’eau.....	40
Figure 16 : Résultat d’acquisition en temps réel la pression de l’eau	41
Figure 17 : Résultat d’acquisition en temps réel de température ambiante	42
Figure 18 : Résultat d’acquisition en temps réel de l’humidité	43
Figure 19 : Résultat d’acquisition en temps réel de l’état de la pompe	44
Figure 20 : Liaison de l’esp32 avec une page web	45
Figure 21 : La bibliothèque ESPAsyncWebServer.....	46
Figure 22 : La bibliothèque ArduinoJSON.h	47

Figure 23 : Structure Fichiers Web et ESP32.....	47
Figure 24 : État général de la page web	50
Figure 25 : État général de la page web lors du fonctionnement du système	51
Figure 26 : Schéma de l'initialisation des paramètres de communication.....	53
Figure 27 : Schéma de la boucle while dans labview [42].....	54
Figure 28 : Programmation graphique dans LabVIEW	54
Figure 29 : Programmation graphique de sortie de commande dans LabVIEW ...	55
Figure 30 : Programmation graphique de l'arrêt sécurisé dans LabVIEW	56
Figure 31 : L'ensemble de l'architecture graphique de note système	56
Figure 32 : Le schéma final du digramme bloc fonctionnel	58
Figure 33 : État du système lors de l'exécution	59
Figure 34 . Architecture YOLOv11 [44].	62
Figure 35 : Quelques images de la base des données du qualité du sol	63
Figure 36 : Résultats d'entraînement du model	64
Figure 37 : Courbes d'apprentissage du model.....	64
Figure 39 : Les courbes des pertes	65
Figure 40 : Les courbes de Précision et de Rappel	66
Figure 41 : Les courbes mAP.....	67
Figure 42 : Quelques images de la base des données du des fuites des tuyaux	69
Figure 43 : Courbes d'apprentissage du model.....	70
Figure 44 : Les courbes des pertes	70
Figure 45 : Les courbes de Précision et de Rappel	71
Figure 46 : Les courbes mAP.....	72

Figure 47 : Comparaison de YOLOv12 par rapport aux versions précédentes	74
Figure 48 : Quelques images de la base des données de l'état des plantes.....	75
Figure 49 : Courbes d'apprentissage du model.....	76
Figure 50 : Les courbes de Précision et de Rappel	76
Figure 51 : Les courbes mAP.....	77
Figure 52 : Les courbes des pertes	78
Figure 53 : Démarche d'hébergement avec Roboflow	80
Figure 54 : Sélection du type de modèle à déployer sur Roboflow.	81
Figure 55 : Chargement des modèles	82
Figure 56 : Interface générée pour la détection en temps réel via la caméra.....	83
Figure 57 : L'API générée pour la détection à partir des captures fournies.....	83
Figure 58 : résultat de détection en direct en temps réel.....	85
Figure 59 : résultat obtenu instantanément par l'API.	86

Liste des tableaux

Tableau 1 : Choix technologiques	35
Tableau 2 : Métriques d'évaluation du YOLOv11	64
Tableau 3 : Résultats d'entraînement du model	69
Tableau 4 : Métriques d'évaluation YOLOv12.....	75
Tableau 5 : comparaison des résultats avec des travaux récentes	88

LISTE DES ABRÉVIATIONS

IoT	Internet of Things (Internet des Objets)
IA	Intelligence Artificielle
DL	Deep Learning
ML	Machine Learning
CV	Computer Vision (Vision par ordinateur)
API	Application Programming Interface
IDE	Integrated Development Environment
MCU	Microcontroller Unit (Unité de microcontrôle)
GPIO	General Purpose Input/Output
HMI	Human-Machine Interface
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
HTML	HyperText Markup Language
API Key	Clé de l'interface de programmation (clé d'accès)
TCP/IP	Transmission Control Protocol / Internet Protocol
HTTP	Hypertext Transfer Protocol
YOLO	You Only Look Once (algorithme de détection d'objet)
mAP	mean Average Precision
IoU	Intersection over Union
F1-score	Indice F1 : moyenne harmonique entre précision et rappel

Résumé

Dans un contexte où l'agriculture est confrontée à des défis croissants en matière d'optimisation, de surveillance en temps réel et de prise de décision autonome, ce travail de recherche propose le développement et le déploiement d'un système embarqué intelligent destiné à l'agriculture de précision. L'originalité de cette approche réside dans l'intégration de plusieurs sous-systèmes complémentaires, formant une solution globale, flexible et optimisée. Ce système embarqué permet l'automatisation en s'appuyant sur un réseau de capteurs environnementaux assurant un suivi continu des paramètres essentiels.

L'aspect novateur de notre approche repose sur l'utilisation de modèles d'apprentissage profond, légers et personnalisés, accessibles via une interface en ligne ou directement depuis un téléphone, sans nécessiter de puissance de calcul locale. Ces modèles ont été améliorés de manière progressive afin d'en augmenter la précision et la fiabilité. Nos résultats expérimentaux ont clairement montré que le système est capable de détecter avec une grande sensibilité des anomalies souvent difficiles à percevoir, comme de petites fuites dans les tuyaux, des variations discrètes de la qualité du sol ou les tout premiers signes de maladies sur les plantes. La détection peut s'effectuer aussi bien en temps réel sur le terrain, grâce à des dispositifs mobiles, qu'à distance via un site web, offrant ainsi une grande flexibilité d'usage. De plus, l'intégration d'une API performante rend l'expérience encore plus fluide et personnalisable : il suffit d'envoyer une image, et le système fournit instantanément une analyse précise et détaillée. Cette combinaison entre

accessibilité, réactivité et intelligence permet une surveillance continue et efficace, ouvrant la voie à une agriculture moderne, connectée et adaptée aux besoins réels du terrain.

Par ailleurs, les modèles de vision par ordinateur ont été entraînés sur des jeux de données réels, et leur performance a été évaluée à l'aide de métriques standards telles que la précision, le rappel et le F1-score. Nos résultats expérimentaux ont démontré une supériorité notable par rapport aux approches existantes, confirmant l'efficacité, la robustesse et la fiabilité du système proposé. Cette recherche met en lumière le potentiel des technologies embarquées combinées à l'intelligence artificielle pour répondre aux enjeux de l'agriculture moderne de manière innovante, évolutive et autonome. Ce projet illustre ainsi la faisabilité d'une solution complète alliant systèmes embarqués, IoT et IA pour une agriculture intelligente, autonome et accessible.

Mots-clés_ Agriculture intelligente, systèmes embarqués, IoT, vision par ordinateur, apprentissage profond léger, détection d'anomalies, surveillance en temps réel, automatisation, interface web et mobile.

Abstract

In a context where agriculture is facing increasing challenges in terms of optimization, real-time monitoring, and autonomous decision-making, this research proposes the development and deployment of an intelligent embedded system designed for small-scale precision agriculture. The originality of this approach lies in the integration of multiple complementary subsystems, forming a global, flexible, and optimized solution. This embedded system enables automation by relying on a network of environmental sensors that ensure continuous monitoring of essential parameters.

The innovative aspect of our approach is based on the use of lightweight and customized deep learning models, accessible via an online interface or directly from a mobile phone, without the need for local computational power. These models have been progressively refined to improve their accuracy and reliability. Our experimental results clearly demonstrate that the system can detect subtle anomalies that are often difficult to perceive, such as small leaks in pipes, slight variations in soil quality, or the very early signs of plant diseases. Detection can be performed both in real-time on-site using mobile devices, and remotely via a web platform, offering a high degree of flexibility. Furthermore, the integration of a high-performance API makes the experience even more seamless and customizable: users simply upload an image, and the system instantly delivers a precise and detailed analysis. This combination of accessibility, responsiveness, and

intelligence enables continuous and efficient monitoring, paving the way for modern, connected agriculture that adapts to real field needs.

Moreover, the computer vision models were trained on real-world datasets, and their performance was evaluated using standard metrics such as precision, recall, and F1-score. Our experiments showed a notable superiority over existing approaches, confirming the effectiveness, robustness, and reliability of the proposed system. This research highlights the potential of embedded technologies combined with artificial intelligence to address the challenges of modern agriculture in an innovative, scalable, and autonomous way. This project thus demonstrates the feasibility of a complete solution that combines embedded systems, IoT, and AI for intelligent, autonomous, and accessible agriculture.

Keywords_ Smart agriculture, embedded systems, IoT, computer vision, lightweight deep learning, anomaly detection, real-time monitoring, automation, web and mobile interface.

État de l'art

Plusieurs recherches récentes se sont intéressées à l'intégration des technologies intelligentes dans le domaine agricole, avec des approches variées que nous présentons ci-dessous.

Systèmes d'irrigation automatisés

L'optimisation de l'irrigation constitue un levier crucial pour une agriculture durable, notamment dans les zones soumises au stress hydrique. De nombreux travaux ont exploré l'automatisation de cette tâche à travers des systèmes intelligents, accessibles financièrement et énergétiquement sobres.

Rafael C. Borges et al [1] ont posé les fondations d'un système d'irrigation à faible coût, alimenté par énergie solaire et contrôlé via une plateforme IoT. Ce système simple mais fonctionnel permet de piloter à distance l'arrosage selon les conditions environnementales. Cette approche a démontré qu'un couplage minimal entre énergie renouvelable et technologie connectée permettait déjà un saut qualitatif pour les petites exploitations.

Ce principe a été étendu par Pablo Catota et al [2], qui ont enrichi le système d'un contrôle précis des intrants agricoles (eau, engrais, pesticides), avec des mouvements tridimensionnels mécaniques optimisant la répartition sur le terrain. En parallèle, Bouali Et-taibi et al [3] ont intégré la dimension big data et la prévision météorologique, démontrant la valeur d'un pilotage adaptatif de l'irrigation en fonction de données agrégées à l'échelle régionale.

La montée en complexité s'est poursuivie avec F. Idris et al [4], qui ont introduit la régulation automatique de la conductivité électrique (EC) dans des systèmes de fertigation, accompagnée de notifications en cas d'anomalie. Ce système intelligent, piloté par un ESP32, illustre la transition vers des dispositifs capables d'auto-régulation.

Enfin, Amritpal Kaur et al [5] ont exploité des algorithmes d'apprentissage automatique pour prédire les besoins hydriques et déclencher automatiquement l'irrigation. Le modèle atteint une précision de 93,5 %, bouclant une première boucle d'évolution vers des systèmes agricoles auto-apprenants et prédictifs.

Connectivité en zones rurales et solutions sans Internet

La connectivité reste un défi fondamental dans les zones agricoles reculées. L'absence d'infrastructure réseau pousse à innover vers des systèmes hors-ligne ou à longue portée.

Gembali Dinesh et al [6] ont intégré la technologie LoRa, qui permet la communication de capteurs à plusieurs kilomètres sans accès Internet. Ce protocole assure une transmission fiable, avec une faible consommation énergétique, et convient parfaitement aux exploitations rurales déconnectées.

Dans le même esprit, Kaustav Moni Hazarika et al [7] ont développé un système d'irrigation contrôlable via SMS, contournant ainsi la nécessité de plateformes web. Cette solution GSM offre un contrôle distant universel, viable même dans les environnements les plus isolés.

Ces deux approches montrent que la résilience technologique est aussi une composante clé de l'agriculture intelligente.

Interfaces de supervision et commande à distance

Avec la multiplication des dispositifs connectés, le besoin d'interfaces accessibles et ergonomiques s'est intensifié. Des solutions variées ont été proposées pour permettre une commande à distance simplifiée.

Lucky Yosua et Rino [8] ont utilisé Telegram comme canal de commande pour une application domotique. Cette interface peut être facilement transposée aux systèmes agricoles, grâce à sa simplicité et son accessibilité sur mobile.

Pavol Kuchár et al [9] ont intégré la vision par ordinateur dans la gestion éthique des nuisibles : un système embarqué différencie animaux domestiques et nuisibles pour permettre des captures sélectives et non létales, illustrant un croisement entre traitement embarqué et action cloud.

Dans le domaine de la surveillance santé, Norsuzila Ya'acob et al [10] ont démontré que l'ESP32 permettait une acquisition continue de signaux vitaux, avec un affichage en temps réel sur application mobile. Cette architecture confirme la robustesse et la flexibilité du microcontrôleur dans divers contextes.

Nahomy José et al [11], dans une perspective agricole, ont couplé ESP32, Blynk et MATLAB pour la surveillance des cultures de café, validant leur système par des mesures réelles. Ensemble, ces travaux montrent l'importance croissante de l'interface utilisateur dans la chaîne de valeur technologique agricole.

Surveillance intelligente des cultures via IoT et machine learning

L'Internet des objets devient encore plus puissant lorsqu'il est couplé à l'intelligence artificielle. Ces approches permettent une prise de décision contextuelle.

Rekha J. Patil et al [12] ont été parmi les premiers à fusionner IoT et machine learning pour recommander les emplacements optimaux de plantation, à partir des données du sol. Cette démarche marque un tournant dans la gestion culturale basée sur la donnée.

Ghulam Mohyuddin et al [13] ont intégré le machine learning à l'agriculture pour favoriser des décisions intelligentes basées sur les données. Leur solution analyse les paramètres environnementaux afin de prédire les cultures optimales, détecter précocement les maladies, améliorer la gestion de l'irrigation et optimiser l'usage des ressources.

Nuwan Jaliyagoda et al [14] ont transformé les tunnels agricoles en laboratoires connectés, prouvant que même les environnements fermés peuvent être exploités pour l'expérimentation intelligente.

D'un autre point, Shola Usharani et al [15] ont mis au point un système de détection d'intrusions animales sur les cultures, utilisant des capteurs couplés à des modules d'alerte à distance. Teddy Gunawan et al [16] ont démontré qu'un arrosage piloté par mobile permettait de maintenir des conditions optimales pour la culture du piment.

Ces contributions forment une base solide pour des systèmes de surveillance agricoles intelligents, exploitables à grande échelle et personnalisables selon les cultures.

Vision par ordinateur et détection d'anomalies par IA

La reconnaissance visuelle automatisée a connu une croissance significative, en particulier grâce aux réseaux de neurones convolutifs et aux modèles comme YOLO ou ResNet.

Mon Timothy Isaac M. Carcellar et al [17] ont prouvé l'efficacité de YOLOv5 sur Raspberry Pi pour détecter les clignements, montrant la faisabilité d'un traitement embarqué en temps réel.

Subramani et al [18] ont appliqué un CNN à la classification visuelle des sols, atteignant 97 % de précision. Kumar et al [19] ont proposé une méthode hybride combinant chaotic spider monkey optimization et représentation Bag-of-Features.

Agarwal et al [20] ont raffiné l'approche visuelle avec l'optimisation HGSO, axée sur la génération de descripteurs visuels. Latha et Kumaresan [21] ont comparé CNN et LSTM pour prédire les nutriments du sol, montrant la supériorité des réseaux profonds.

Saki et al [22] ont mené une revue systématique sur les Transformers, utilisés pour la fusion de données agricoles, tandis que Kavitha et al [23] ont utilisé ResNet pour extraire les caractéristiques visuelles du sol.

Enfin, Yue Shen et al [47] ont proposé une version optimisée de YOLOv8, intégrant des convolutions groupées, un bloc SE (Squeeze-and-Excitation) et l'optimiseur AdamW, pour détecter les feuilles de tomate malades. Entraîné sur 1500 images, le

modèle a été déployé sur un robot de pulvérisation, assurant une application ciblée et durable des pesticides.

Robotique agricole et systèmes embarqués autonomes

La robotisation complète du processus agricole devient progressivement une réalité. Ces systèmes réunissent vision, décision et action dans une seule entité embarquée.

Vijay Mahore et al [24] ont instrumenté un tracteur avec des capteurs connectés à un Raspberry Pi, mesurant la traction, le glissement et la profondeur de travail.

Lamia Mezai et al [25] ont conçu un robot mobile de désinfection à commande distante, applicable aux environnements critiques comme les hôpitaux ou les serres agricoles. Sowmya C. H. et al [26] ont adapté les techniques de deep learning pour la conduite autonome sur routes locales, posant les bases du guidage autonome d'outils agricoles.

Tarun et Esther [27] ont démontré que EfficientDet-Lite, embarqué sur Raspberry Pi, permet une détection rapide des panneaux de signalisation. Ces solutions confirment la faisabilité de systèmes embarqués légers, mobiles, et performants.

Synthèse et positionnement de notre approche

Liakos et al [28] ont identifié quatre pôles majeurs dans l'IA appliquée à l'agriculture : la gestion des cultures, du sol, de l'eau et du bétail. Cependant, de nombreux travaux existants abordent ces pôles de manière séparée : IoT sans IA, IA sans déploiement embarqué, interfaces sans supervision locale.

Chapitre 1

Introduction générale

Au fil des dernières années, l'agriculture a connu une transformation progressive mais significative, marquée par l'intégration croissante des technologies numériques, de l'Internet des objets (IoT) et de l'intelligence artificielle (IA). Face aux défis climatiques, à la pression démographique et à la nécessité d'optimiser l'utilisation des ressources naturelles, les pratiques agricoles traditionnelles ne suffisent plus à répondre aux exigences modernes. De ce constat est née l'agriculture intelligente, une approche qui combine automatisation, surveillance en temps réel et prise de décision autonome pour accroître la productivité tout en minimisant les pertes.

L'agriculture intelligente à petite échelle s'impose comme une solution durable, en particulier dans les zones rurales ou les exploitations de taille modeste. Elle permet aux agriculteurs de bénéficier de technologies avancées sans exiger de lourds investissements en infrastructure ou en matériel. Le cœur de cette révolution repose sur des systèmes embarqués connectés à un réseau de capteurs, capables de collecter des données sur l'environnement (température, humidité, qualité du sol, niveau d'eau, etc.) et de les traiter localement ou à distance à l'aide d'algorithmes d'apprentissage automatique ou profond. Ces données servent ensuite à déclencher automatiquement des actionneurs comme des pompes, des ventilateurs ou des systèmes d'irrigation, selon les besoins détectés.

Par ailleurs, la détection d'anomalies (AD) dans les données environnementales agricoles est devenue un axe de recherche fondamental. Elle consiste à identifier des comportements ou valeurs inattendus qui peuvent signaler des problèmes potentiels, tels qu'une fuite d'eau, une dégradation du sol ou un début de maladie des plantes. L'AD permet donc d'anticiper et de corriger les défaillances avant qu'elles ne deviennent critiques. Grâce à l'IA embarquée, cette surveillance peut se faire en temps

réel, avec des alertes envoyées automatiquement à l'utilisateur via une interface mobile ou web, offrant une grande flexibilité d'utilisation.

1.1. Contexte général

Traditionnellement, les petits exploitants agricoles devaient compter sur leur observation manuelle pour surveiller l'état des cultures et du sol. Cette méthode, bien qu'efficace dans certains cas, demeure chronophage, peu précise et sujette à l'erreur humaine. Aujourd'hui, grâce à l'évolution des capteurs numériques et des microcontrôleurs à faible consommation énergétique (comme l'ESP32, Raspberry pi ...), il est désormais possible de déployer des dispositifs embarqués capables de capter les moindres variations environnementales avec une grande précision et à moindre coût.

Cette technologie ouvre de nouvelles perspectives pour les systèmes d'irrigation intelligente, la gestion automatisée de la ventilation des serres, ou encore l'analyse de l'état de santé des plantes. Les modèles de vision par ordinateur peuvent par exemple détecter l'apparition de taches suspectes sur les feuilles ou des décolorations indiquant un stress hydrique ou une infection. Ces modèles, optimisés pour les plateformes embarquées, utilisent des architectures légères, permettant une exécution fluide sans nécessiter de ressources informatiques importantes.

L'objectif de cette recherche est de concevoir et déployer un système embarqué intelligent dédié à l'agriculture de précision à petite échelle, combinant des capteurs environnementaux, une interface web de contrôle, des modèles d'apprentissage profond optimisés, et un accès distant via Wi-Fi ou réseau mobile. À travers ce système, il devient possible de surveiller les conditions de culture en continu, de détecter les anomalies à un stade précoce, et d'intervenir automatiquement ou à distance selon le scénario défini.

Ce travail s'inscrit ainsi dans une dynamique de modernisation des pratiques agricoles, en s'appuyant sur des technologies accessibles, légères et fiables. Il contribue à rendre l'agriculture plus résiliente, autonome et durable, tout en répondant aux enjeux actuels de sécurité alimentaire et d'optimisation des ressources.

1.2. Problématique

Parmi les principales difficultés rencontrées, on peut citer :

- Difficulté d'inspecter et d'observer une anomalie dans les cas complexes.
- Prise de temps avec un coût élevé pour l'inspection et détection des anomalies
- La gestion qui exige une présence continue
- Gaspillages et pertes récurrents dus à un suivi insuffisant

1.3. Originalités

- Intégration complète et combinée des trois technologies clés (IoT + IA + système embarqué connecté).
- Contrôle locale et à distance
- L'optimisation et la mise en œuvre des modèles IA
- L'exploitation de la dernière version de YOLOv12.
- Adaptation d'un logiciel industriel (LabVIEW) à un microcontrôleur pour notre projet
- Étude et développement d'un système intelligent connecté

1.4. Les Objectifs

➤ Objectif principal :

Ce travail de recherche vise à concevoir et implémenter un système embarqué intelligent combinant l'IoT et l'intelligence artificielle légère, personnalisée et optimisée. L'objectif est de permettre une surveillance en temps réel local et à distance, une détection d'anomalies précise et une prise de décision autonome dans l'agriculture de précision dans un modèle réduit.

➤ Objectifs spécifiques :

- Analyse et traitement des données
- Développement d'une architecture IoT
- Conception, entraînement et personnalisation des modèles d'IA
- Affichage et contrôle locale et à distance en temps réel
- Détection et localisation en temps réel des anomalies
- Validation et test du système dans un environnement réel

1.5. Les hypothèses

Ce projet de recherche repose sur plusieurs hypothèses de travail.

- Les données utilisées sont des données réelles prises par camera.
- La communication réseau repose exclusivement sur le Wi-Fi
- L'IA se limite à la détection visuelle d'anomalies.
- L'interface de surveillance et de commande est accessible à distance via un navigateur web.
- Le contrôleur dispose des ressources suffisantes (mémoire, vitesse, connectivité) pour gérer un système embarqué autonome.

1.6. Méthodologie

La réalisation de ce projet s'est appuyée sur une approche en plusieurs étapes structurées :

- ✓ La première phase est une revue de la littérature.
- ✓ Analyse des besoins et spécifications
- ✓ Conception du système embarqué
- ✓ Mise en place de la communication IoT
- ✓ Intégration de l'intelligence artificielle (IA)
- ✓ Test, validation et démonstration

Chapitre 2

Aperçu sur l’Internet des Objets, les Systèmes Embarqués, et la Détection et Prédiction des Anomalies Agricole

2.1. Introduction

L'essor technologique de ces dernières années a profondément transformé l'agriculture, un secteur historiquement perçu comme traditionnel. L'intégration de l'Internet des Objets (IoT), des systèmes embarqués et de l'intelligence artificielle (IA) a permis de développer ce que l'on appelle aujourd'hui l'agriculture intelligente ou agriculture de précision. Cette mutation technologique vise à accroître les rendements agricoles tout en optimisant l'utilisation des ressources naturelles et en minimisant l'impact environnemental. Ce chapitre présente d'abord les concepts de base liés à l'IoT agricole et aux systèmes embarqués, avant d'approfondir les approches modernes de détection et prédiction des anomalies, essentielles pour maintenir et améliorer les performances des systèmes agricoles intelligents.

2.2. L’Internet des Objets en Agriculture Intelligente

2.2.1. Définition et rôle de l’IoT en agriculture

L’Internet des Objets (IoT) désigne l’ensemble des dispositifs interconnectés capables de collecter, transmettre et traiter des données sans intervention humaine directe. Appliqué à l’agriculture, l’IoT permet de connecter capteurs, actionneurs, drones, robots agricoles, systèmes d’irrigation et équipements météorologiques afin de surveiller et de contrôler les différentes variables du milieu agricole en temps réel [29]. Grâce à l’IoT, il est aujourd’hui possible de mesurer de manière continue l’humidité du

sol, la température, la luminosité, la composition chimique du sol, la vitesse du vent et même l'état de santé des plantes à l'aide de caméras multispectrales.

L'utilisation de ces technologies vise plusieurs objectifs majeurs : optimiser la consommation d'eau, réduire l'usage d'engrais et de pesticides, détecter rapidement les maladies végétales, et améliorer la prise de décision des agriculteurs. Cette approche fondée sur des données en temps réel constitue une rupture avec l'agriculture traditionnelle, largement basée sur l'expérience empirique et les interventions manuelles [30] .

2.2.2. Architecture d'un système IoT agricole

Un système IoT dédié à l'agriculture repose sur une architecture classique en couches, comprenant :

- La couche perception, composée des capteurs et actionneurs installés dans les champs ou les serres.
- La couche réseau, qui assure la transmission des données vers des systèmes de traitement, via Wi-Fi, LoRaWAN, GSM ou satellites.
- La couche de traitement, qui peut inclure des microcontrôleurs embarqués, des serveurs locaux ou des plateformes cloud où les données sont analysées.
- La couche application, qui propose des interfaces de visualisation, d'analyse et de contrôle pour l'utilisateur final [31] .

2.3. Les Systèmes Embarqués : Fondation de l'agriculture intelligente

2.3.1. Définition et caractéristiques générales

Un système embarqué est un système informatique autonome, dédié à exécuter une tâche spécifique souvent en temps réel, avec des contraintes strictes de mémoire, de

puissance de calcul et de consommation énergétique [32] . Contrairement aux ordinateurs de bureau ou aux serveurs, les systèmes embarqués doivent être compacts, robustes, fiables et souvent capables de fonctionner dans des environnements difficiles (température élevée, humidité, vibrations, etc.).

En agriculture intelligente, les systèmes embarqués sont au cœur des dispositifs IoT. Ils permettent de collecter les données des capteurs, de les traiter localement (edge computing), de prendre des décisions automatisées (par exemple, déclencher l'irrigation), et d'envoyer les résultats vers des serveurs distants pour un stockage ou un traitement plus poussé.

2.3.2. Pourquoi utiliser des systèmes embarqués en agriculture intelligente

L'usage des systèmes embarqués dans l'agriculture présente plusieurs avantages majeurs :

- **Autonomie et robustesse** : les dispositifs doivent pouvoir fonctionner de manière autonome pendant de longues périodes, souvent dans des conditions environnementales difficiles.
- **Réduction des coûts** : par rapport à l'utilisation de PC industriels ou d'installations informatiques lourdes, les systèmes embarqués tels que les microcontrôleurs ESP32 ou les modules Raspberry Pi permettent de développer des solutions à faible coût.
- **Optimisation de la connectivité** : dans des zones rurales éloignées où l'accès aux réseaux est limité, il est crucial d'utiliser des protocoles de communication légers et efficaces, ce qui est rendu possible par les systèmes embarqués modernes.
- **Traitement local** : pour réduire la latence et économiser la bande passante, il est essentiel de traiter certaines données directement sur place. Les systèmes

embarqués récents permettent d'exécuter des modèles d'IA optimisés directement à proximité des capteurs, sans passer systématiquement par le cloud.

Ainsi, dans notre projet de recherche, le choix d'implanter l'intelligence artificielle sur des systèmes embarqués spécifiques est motivé par la recherche de performance, d'autonomie énergétique, et de pertinence opérationnelle en environnement agricole.

2.4. Détection et Prédiction d'Anomalies : Un Outil Clé pour l'Optimisation Agricole

2.4.1. Concepts généraux

La détection d'anomalies consiste à identifier des observations ou comportements différents du fonctionnement normal d'un système [33]. En agriculture intelligente, les anomalies peuvent être liées à un stress hydrique, une attaque parasitaire, une défaillance d'un capteur ou d'un actionneur, ou encore à une mauvaise qualité des données collectées.

La prédiction d'anomalies vise à anticiper ces événements avant qu'ils ne surviennent, afin de permettre une maintenance prédictive, réduire les pertes et améliorer l'efficacité opérationnelle [34].

2.4.2. Techniques modernes de détection d'anomalies adaptées aux systèmes embarqués

Pour des systèmes embarqués à ressources limitées, il est crucial de choisir des algorithmes légers et efficaces. Parmi les méthodes les plus adaptées, on retrouve :

- Isolation Forest : algorithme basé sur l'idée que les anomalies sont plus facilement isolables que les points normaux [35].
- Autoencodeurs légers (TinyML) : réseaux neuronaux compacts capables de détecter des anomalies en comparant l'entrée et la sortie après reconstruction [36].

- Local Outlier Factor (LOF) : approche basée sur la densité locale des points pour identifier ceux qui sont isolés ou atypiques.
- One-Class SVM : apprentissage sur des données normales pour définir une frontière autour du comportement attendu.

2.5. Importance de l'IA embarquée pour la prédiction en agriculture

Le déploiement de modèles de Machine Learning (ML) ou Deep Learning (DL) dans l'agriculture intelligente ne peut se faire qu'à travers une approche optimisée : le Tiny Machine Learning (TinyML). TinyML consiste à déployer des modèles d'IA compacts directement sur des microcontrôleurs embarqués (ESP32, ARM Cortex-M4, etc.) avec des capacités de calcul limitées [37].

En travaillant directement au niveau de la périphérie (edge computing), cette approche permet :

- De garantir une latence extrêmement faible,
- D'économiser de la bande passante réseau,
- De préserver la vie privée des données sensibles (locales aux exploitations),
- D'assurer une autonomie énergétique plus longue.

Ceci est fondamental pour notre projet, car il s'agit d'assurer des performances élevées tout en garantissant la viabilité pratique sur le terrain.

2.6. Défis et Perspectives

Malgré les avancées notables, plusieurs défis restent à surmonter pour généraliser l'utilisation de l'IoT et de l'IA embarquée dans l'agriculture :

- Gestion de l'énergie : prolonger la durée de vie des batteries dans des environnements sans accès électrique permanent.
- Fiabilité des capteurs : maintenir une précision suffisante malgré des conditions climatiques extrêmes.
- Sécurité et cybersécurité : protéger les dispositifs et les données contre les intrusions ou les altérations malveillantes.
- Évolutivité : rendre les systèmes modulaires et adaptables aux changements d'échelle ou de besoins.

La recherche actuelle se concentre donc non seulement sur l'amélioration des performances matérielles, mais aussi sur l'optimisation des algorithmes pour s'adapter à ces contraintes [38].

2.7. Conclusion

Le présent chapitre a présenté un aperçu des principales technologies clés sur lesquelles repose l'agriculture intelligente moderne. L'Internet des Objets agricole, combiné aux systèmes embarqués performants et à l'intelligence artificielle légère, offre des perspectives d'amélioration considérables pour la productivité agricole durable. Ces technologies, bien que prometteuses, nécessitent encore des travaux de recherche pour lever les défis liés à la connectivité, l'énergie, la précision et la sécurité. Notre projet s'inscrit pleinement dans cette dynamique, en explorant les possibilités offertes par des systèmes embarqués optimisés pour la détection précoce des anomalies agricoles.

Chapitre 3

Conception et mise en œuvre du système embarqué

3.1. Introduction

L'intégration des technologies embarquées dans le domaine agricole constitue aujourd'hui une réponse pertinente aux défis de la productivité, de la durabilité et de l'optimisation des ressources. À l'heure où les changements climatiques et la pression sur les ressources naturelles imposent une gestion plus fine des intrants agricoles (eau, engrais, énergie), le recours à des systèmes intelligents et autonomes devient indispensable. Le concept d'agriculture intelligente (ou *Smart Agriculture*) repose notamment sur l'emploi de systèmes embarqués capables de mesurer en temps réel les paramètres clés de l'environnement agricole et de prendre des décisions locales ou centralisées pour agir sur le terrain.

Dans ce contexte, ce chapitre présente en détail la conception, la mise en œuvre et les choix techniques d'un système embarqué original, basé sur la carte ESP32 et un ensemble de cinq capteurs environnementaux, dans le cadre d'une solution de surveillance et de contrôle intelligent de conditions agricoles. L'objectif est de montrer comment un tel système peut, à coût réduit, collecter des données précises, les transmettre de façon fiable et piloter des actionneurs en fonction de règles prédéfinies ou de modèles d'intelligence artificielle.

Ce travail repose notamment sur :

- Le choix stratégique de l'ESP32, un microcontrôleur double cœur avec connectivité Wi-Fi intégrée, parfaitement adapté aux projets IoT pour une faible consommation ;

- L'intégration cohérente de cinq capteurs complémentaires, permettant une vision multidimensionnelle de l'état environnemental ;
- Une conception orientée vers la simplicité de déploiement et l'extensibilité future (ajout d'autres capteurs, transmission cloud, contrôle à distance, intelligence embarquée).

3.2. Architecture proposée du système embarqué pour l'agriculture intelligente

L'objectif principal de cette section est de présenter en détail l'architecture matérielle et logicielle du système embarqué conçu pour améliorer la gestion d'une petite exploitation agricole. Cette architecture tire parti de l'ESP32 comme cœur du système, accompagné de cinq capteurs essentiels à la surveillance environnementale et à la prise de décision automatique à travers des actionneurs intelligents.

3.2.1. Choix technologique et justification

a) Microcontrôleur : ESP32

Le microcontrôleur ESP32 a été choisi pour son excellent rapport performance-coût, sa double connectivité Wi-Fi et Bluetooth, sa faible consommation énergétique, et ses nombreux GPIOs compatibles avec différents types de capteurs (analogiques et numériques). Il est parfaitement adapté aux applications IoT dans un contexte agricole grâce à :

- Un processeur dual-core
- Une mémoire RAM suffisante
- Une compatibilité avec le développement via IDE et MicroPython
- Une intégration possible avec des services cloud ou locaux

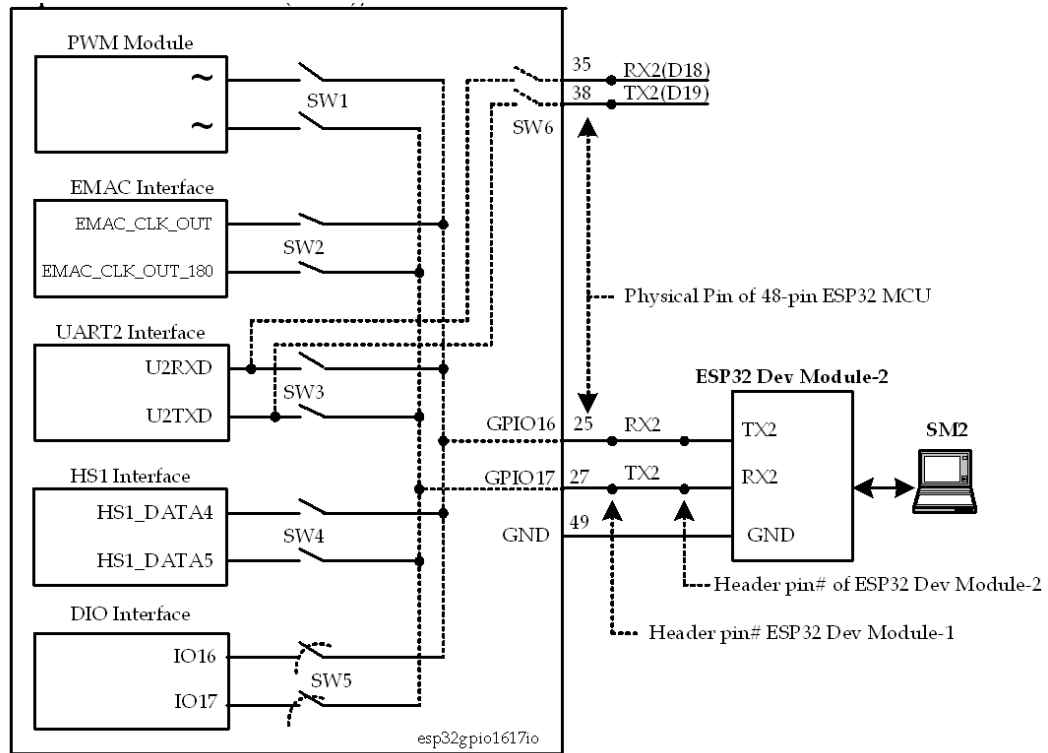


Figure 1 : Architecture du esp32 [39].



Figure 2 : Esp32 réel

b) Capteurs sélectionnés

Le système comprend plusieurs types de capteurs représentatifs des variables les plus critiques pour l'agriculture intelligente :

- **DHT11 – Le capteur météo de notre système**

Le DHT11 est un capteur très pratique qui mesure la température et l'humidité de l'air. Dans un projet agricole automatisé, il joue un rôle essentiel qui aide à surveiller l'état ambiante, une utilisation fiable, et permet à notre système de s'adapter automatiquement aux conditions pour assurer un bon développement des cultures.

Connexion :

- **VCC** → VCC de l'ESP32
- **GND** → GND de l'ESP32
- **DATA** → GPIO 4 de l'ESP32
- **Résistance** : Une résistance de **10 k Ω** entre DATA et VCC est recommandée pour assurer une communication stable.

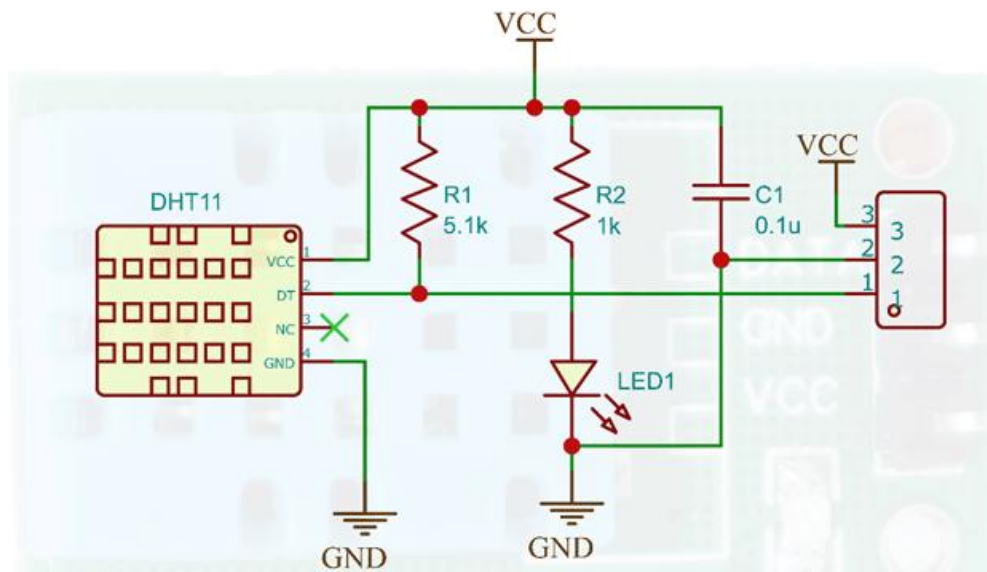


Figure 3 : Schéma de circuit du module DHT11 [40].

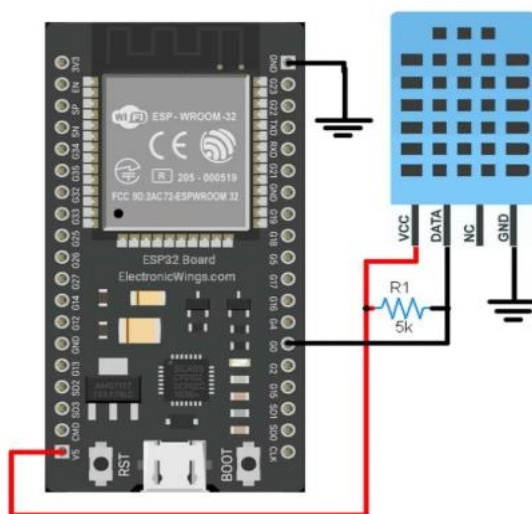


Figure 4 : ESP32 interfacé avec DHT11

- **Capteur capacitif d'humidité du sol**

Ce capteur permet de connaître l'humidité du sol. Il fonctionne sans contact direct avec l'eau, ce qui le rend plus durable et plus stable dans le temps. Grâce à lui, notre

système sait quand la terre est sèche ou humide, ce qui permet d'arroser juste ce qu'il faut, au bon moment.

Connexion :

- **VCC** → VCC de l'ESP32
- **GND** → GND de l'ESP32
- **AOUT** → GPIO 36 (entrée analogique) de l'ESP32

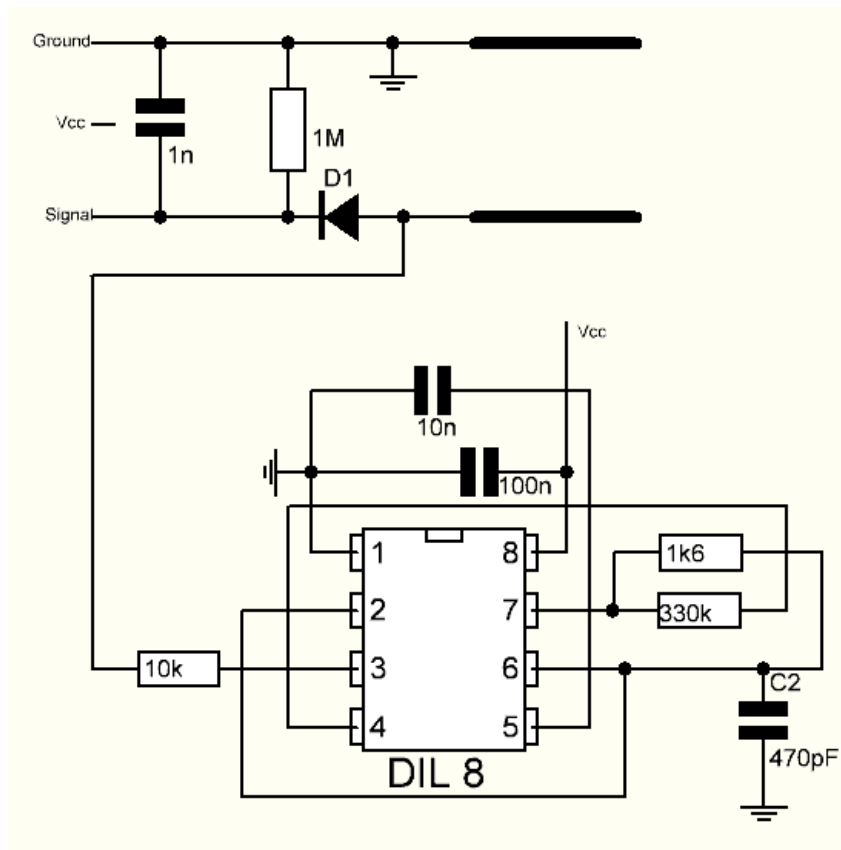


Figure 5 : Schéma de circuit du module capteur capacitif d'humidité du sol [41]

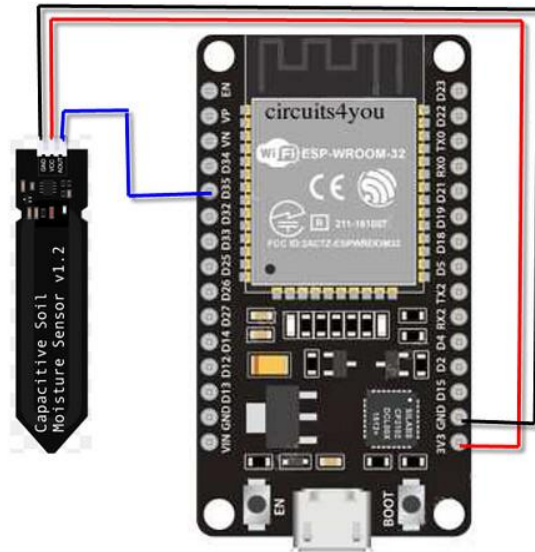


Figure 6 : ESP32 interfacé avec capteur capacitif d'humidité du sol

- **Capteur de niveau d'eau ST-045**

Le capteur de niveau d'eau joue un rôle stratégique dans la gestion hydrique de notre système intelligent. Placé dans le réservoir, il surveille le niveau d'eau disponible, prêt à alerter le système au moindre signe de baisse critique. Intuitif, simple et fiable, pour une irrigation intelligente.

Connexion :

- **VCC** → VCC de l'ESP32
- **GND** → GND de l'ESP32
- **S (Signal)** → GPIO 34 (entrée analogique) de l'ESP32

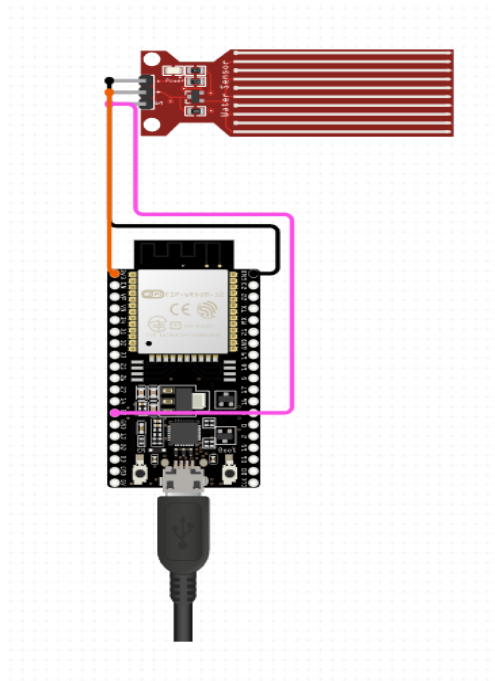


Figure 7 : ESP32 interfacé avec capteur ST-045

- **Capteur de débit d'eau (YF-S201)**

Capteur débitmètre intelligent, placé au cœur du système pour la gestion de flux de notre système d'irrigation. Mesure précisément la quantité d'eau circulant dans les conduites, et il permet un arrosage ciblé, rationné et respectueux de l'environnement

Connexion :

- **VCC** → VCC de l'ESP32
- **GND** → GND de l'ESP32
- **Signal** → GPIO 27 de l'ESP32

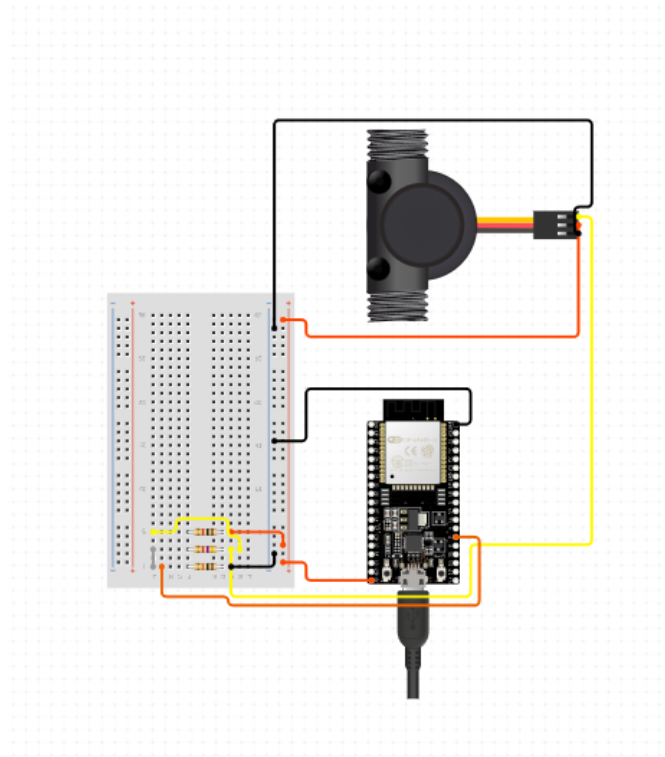


Figure 8 : ESP32 interfacé avec capteur YF-S201

- **Schéma global – Intégration de tous les capteurs avec l'ESP32**

Le schéma global présenté ci-dessous illustre de manière claire et détaillée l'interconnexion entre l'ESP32 et l'ensemble des capteurs utilisés dans notre système embarqué. Il reflète la complémentarité parfaite entre chaque composant, où chaque capteur joue un rôle bien défini dans la surveillance intelligente de l'environnement. Cette architecture a été pensée pour être optimisée pour des performances en temps réel, même dans des conditions de terrain. L'ensemble du câblage et de la disposition a été réalisé à l'aide du simulateur Circuit.io, garantissant précision et lisibilité du montage.

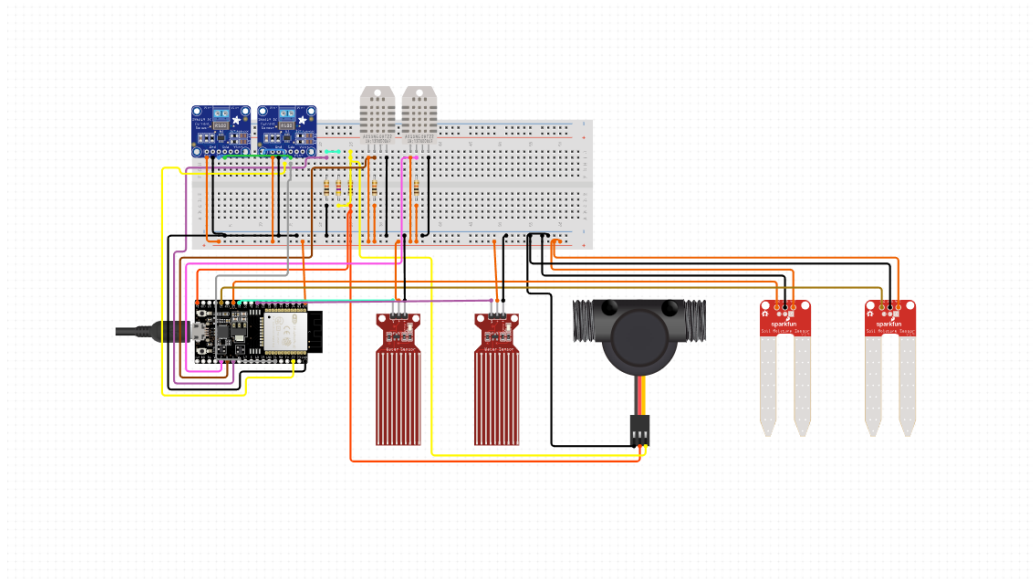


Figure 9 : Schéma global – Intégration de tous les capteurs

c) Actionneur utilisé

Dans notre système embarqué, les actionneurs représentent l'élément actif de l'intelligence embarquée. Grâce à la pompe à eau, l'irrigation devient automatique et précise, déclenchée uniquement lorsque le sol en a réellement besoin.

Ce processus est piloté par l'ESP32, à partir des données reçues des capteurs.

Pour commander la pompe en toute sécurité, nous utilisons un module relais, qui fait le lien entre la partie électronique et les équipements de puissance. L'ensemble est alimenté par une batterie rechargeable, garantissant l'autonomie et la continuité du système. Cette combinaison intelligente offre une solution autonome et adaptée aux besoins réels du terrain.

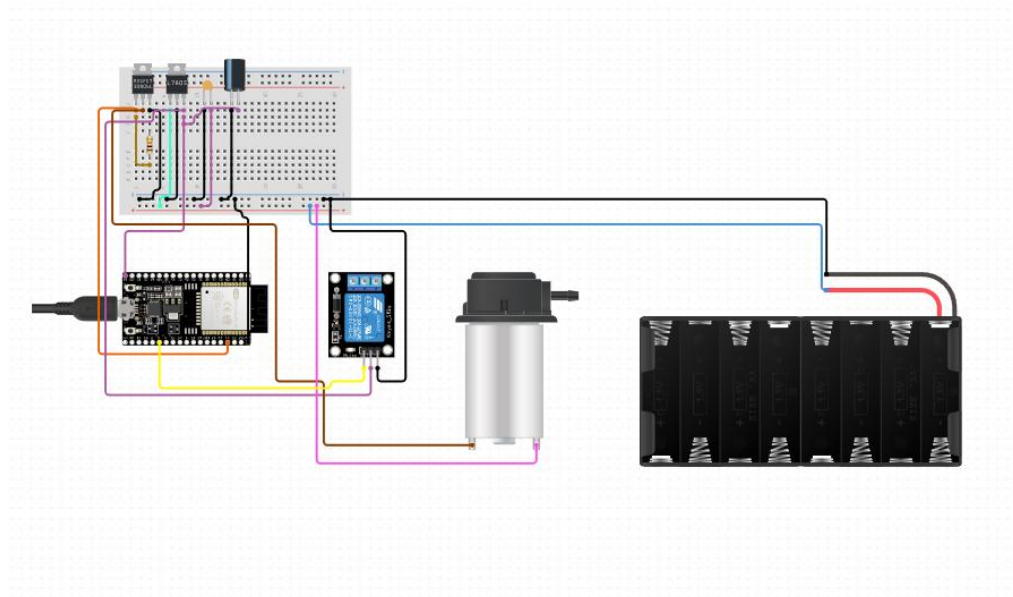


Figure 10 : Schéma global – partie actionneurs

3.3. Programmation description fonctionnelle du système

Après la conception matérielle soignée de l'architecture embarquée, l'étape suivante a consisté à déployer une programmation optimisée entre capteurs, microcontrôleur et actionneurs.

Cette programmation a été réalisée à l'aide du langage C++ au sein de l'environnement IDE, connu pour sa flexibilité avec les cartes ESP32.

L'objectif principal du code était d'assurer un cycle de lecture-réaction autonome, tout en garantissant une faible consommation d'énergie et une rapidité de traitement adaptée aux contraintes du terrain.

À travers une logique embarquée soigneusement pensée, l'ESP32 exécute une série d'opérations cycliques qui permettent de surveiller en temps réel les paramètres

critiques, de prendre des décisions intelligentes localement, et d'agir sans supervision humaine.

Les résultats sont visualisables en direct via le moniteur série, fournissant un retour instantané sur l'état des capteurs, les décisions prises et les éventuelles activations des actionneurs.

Dans cette section, nous présentons en détail :

- La structure générale du programme utilisé (algorithme embarqué),
- Une description fonctionnelle étape par étape,
- Ainsi que les résultats observables sur le moniteur série, illustrant la réactivité et l'autonomie du système.

Cette combinaison entre codage efficace, logique embarquée intelligente et retour d'information en temps réel met en lumière l'efficacité du système proposé, en tant que solution embarquée pour une agriculture intelligente.

3.3.1. Cycle de fonctionnement

✓ Initialisation du système (Power-Up)

Dès la mise sous tension, l'ESP32 effectue une séquence d'initialisation.

Tous les capteurs connectés sont détectés et configurés automatiquement.

Cette phase garantit que chaque module est opérationnel avant le début des mesures.

✓ Lecture périodique des capteurs

À des intervalles réguliers (10 secondes), l'ESP32 déclenche la lecture de l'ensemble des capteurs.

Les mesures collectées offrent une vision complète de l'environnement surveillé, en temps réel.

✓ **Traitement local des données**

L'ESP32 analyse les valeurs reçues à l'aide d'une logique de traitement embarquée.

Ce traitement s'appuie sur des règles conditionnelles

L'objectif est de déterminer si une action doit être déclenchée ou non.

✓ **Décision et activation des actionneurs**

Si une action est nécessaire, l'ESP32 commande les actionneurs via un module relais.

Les composants sont pilotés avec précision et en toute sécurité.

✓ **Communication des données**

Les résultats et états du système sont affichés en local via le port série.

Optionnellement, les données peuvent être envoyées par Wi-Fi, permettant une consultation à distance via une interface ou une application mobile.

✓ **Boucle continue et autonome**

Le système répète ces étapes de façon continue, sans intervention manuelle.

Cela permet une gestion autonome et intelligente de notre environnement.

```

36   flowPulseCount++;
37 }
38
39 void setup() {
40   Serial.begin(115200);
41   WiFi.begin(ssid, password);
42   while (WiFi.status() != WL_CONNECTED) {
43     delay(1000);
44     Serial.print(".");
45   }
46   Serial.println("\nConnecté au WiFi");
47
48   pinMode(flowPin, INPUT_PULLUP);
49   pinMode(relayPin, OUTPUT);
50   digitalWrite(relayPin, HIGH); // Pompe désactivée par défaut
51
52   dht.begin();
53
54   attachInterrupt(digitalPinToInterrupt(flowPin), countFlowPulse, RISING);
55

```

Output Serial Monitor x

Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')

13:51:21.293 -> Humidité du sol : 38%
13:51:21.293 -> Niveau d'eau : 80%
13:51:21.293 -> Température : 24.70 °C
13:51:21.293 -> Humidité air : 60.30%
13:51:21.293 -> Débit d'eau : 53.00 %
13:51:21.337 -> Pompe : ON
13:51:21.337 -> -----
13:51:23.293 -> Humidité du sol : 38%
13:51:23.293 -> Niveau d'eau : 80%
13:51:23.336 -> Température : 24.70 °C
13:51:23.336 -> Humidité air : 60.30%
13:51:23.336 -> Débit d'eau : 53.00 %

Figure 11 : résultat réel de sortie en temps réel sur le moniteur série

3.4. Conclusion

Ce chapitre a présenté le système embarqué développé pour notre solution d’agriculture intelligente, en mettant en valeur l’ESP32 et l’ensemble des capteurs et actionneurs utilisés. À travers la description des capteurs, des actionneurs, et du cycle de fonctionnement automatisé, nous avons démontré la capacité du système à collecter, traiter, et agir en temps réel, tout en offrant une connectivité moderne grâce au Wi-Fi intégré. L'utilisation du langage C++ dans l’environnement Arduino IDE a permis d’exploiter la puissance de l’ESP32 tout en garantissant une logique embarquée claire et efficace.

Chapitre 4

Mise en réseau et communication IoT pour une supervision intelligente

4.1. Introduction

Dans un système intelligent basé sur l'Internet des Objets (IoT), la simple collecte de données ne suffit pas, il est aussi essentiel de pouvoir superviser les mesures en temps réel et d'interagir à distance ou localement avec les actionneurs. Ce besoin de contrôle dynamique et distribué repose sur la mise en réseau de divers éléments, permettant non seulement la transmission fiable des données environnementales, mais aussi la commande efficace des équipements automatisés.

Ce chapitre présente l'ensemble des stratégies de communication mises en œuvre dans ce travail, en lien direct avec la couche matérielle développée au chapitre précédent. L'originalité de cette approche réside dans l'intégration simultanée de trois moyens complémentaires :

- Une plateforme cloud (ThingSpeak) pour la visualisation des données à distance,
- Une interface web embarquée sur le microcontrôleur pour un contrôle personnalisé à distance,
- Un poste de supervision local, basé sur un système hybride entre notre système embarqué et un logiciel industriel (LabVIEW), pour la commande et le suivi sur terrain.

Ainsi, ce chapitre expose en détail l'architecture réseau adoptée, les technologies utilisées, les méthodes d'intégration, et les mécanismes de synchronisation entre les différents modes de commande. Une analyse comparative avec d'autres approches similaires issues de la littérature permet de situer l'apport spécifique de ce système.

4.2. Cadre théorique et concepts clés

La mise en réseau dans un système IoT repose sur l'interconnexion d'objets physiques via des canaux de communication sans fil, permettant l'échange de données, le traitement distribué et la commande à distance. Dans le contexte de ce projet, plusieurs notions fondamentales sont à maîtriser pour comprendre l'architecture mise en place.

4.2.1. Supervision et commande distribuée

Un système distribué en IoT implique que la supervision et le contrôle puissent être exercés depuis différents points d'accès : localement, depuis un poste relié physiquement ou via le même réseau, ou à distance, depuis n'importe quel terminal connecté à Internet. La supervision locale offre une grande réactivité tandis que le contrôle distant permet une accessibilité continue, même en l'absence d'opérateurs sur le terrain.

4.2.2. Protocoles de communication

Les communications dans un système IoT s'appuient généralement sur :

- HTTP/HTTPS : pour la transmission de données vers les services web (comme ThingSpeak).
- TCP/IP (niveau système) : utilisé par les applications pour assurer la fiabilité des communications.
- Communication série : entre microcontrôleur et PC local, pour l'échange de données via des logiciels comme LabVIEW.
- Wi-Fi local : réseau partagé entre le microcontrôleur, le poste de supervision local, et les interfaces distantes.

4.2.3. Plateformes de visualisation des données

Parmi les plateformes existantes, ThingSpeak est une solution cloud utilisée dans les projets IoT. Elle permet de recevoir, stocker, visualiser et analyser les données issues de capteurs à l'aide de graphiques temps réel.

4.2.4. Interfaces utilisateurs dans l'IoT

Une interface utilisateur dans un système IoT sert à simplifier l'interaction avec le système. Elle peut prendre la forme :

- D'un dashboard local, tel qu'un panel LabVIEW,
- D'une page web dynamique hébergée sur le microcontrôleur,
- D'une interface de consultation dans le cloud.

Dans ce projet, ces trois formes ont été implémentées conjointement, Permettant une plus flexibilité d'utilisation.

4.3. Architecture de communication du système IoT proposé

L'architecture de communication de ce système repose sur une structure hybride, combinant des échanges de données en local et à distance. L'objectif est de garantir une supervision continue, un contrôle fiable des actionneurs, ainsi qu'un enregistrement permanent des mesures issues des capteurs.

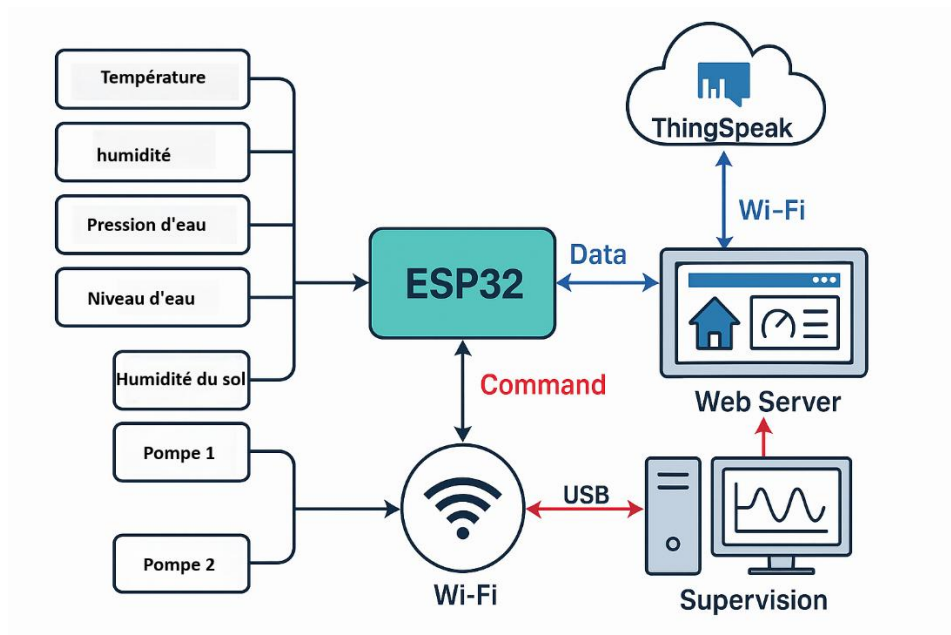
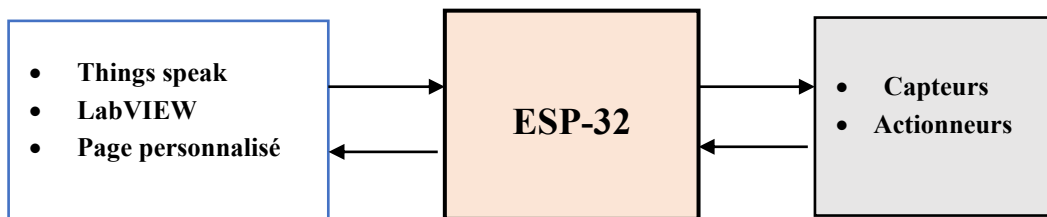


Figure 12 : Architecture de communication proposé

4.3.1. Schéma global de l'architecture



L'architecture adoptée s'articule autour de trois couches fonctionnelles :

- **Couche capteurs/actionneurs (niveau embarqué)** : Constituée d'un microcontrôleur ESP32 connecté à plusieurs capteurs environnementaux.
- **Couche de communication réseau** : L'ESP32 agit comme un nœud intelligent capable de :

- ✓ Transmettre les données vers un service distant (ThingSpeak),
- ✓ Héberger une page web accessible à distance sur le réseau local,
- ✓ Échanger localement avec un poste PC à travers une interface série pour contrôle par LabVIEW.
- **Couche de supervision (locale et distante) :**
 - ✓ Locale : Application LabVIEW pour la visualisation temps réel, le déclenchement d'actionneurs et le suivi des capteurs.
 - ✓ Distant : Interface web personnalisée permettant de consulter les mesures et d'activer manuellement les actionneurs à partir d'un navigateur.
 - ✓ Cloud : Plateforme ThingSpeak assurant la traçabilité et l'analyse graphique des données en le temps.

4.3.2. Communication bidirectionnelle

L'un des aspects innovants du système est sa capacité à recevoir et transmettre des données à double sens :

- Du système vers l'utilisateur : Données issues des capteurs envoyées vers la plateforme cloud et affichées sur les interfaces (web, LabVIEW).
- De l'utilisateur vers le système : Commandes de l'utilisateur (via LabVIEW ou page web) envoyées à l'ESP32 pour activer ou désactiver les actionneurs.

Cette bidirectionnalité permet de passer du simple monitoring à une interaction intelligente avec l'environnement supervisé.

4.3.3. Synchronisation et priorités

Pour éviter les conflits entre plusieurs modes de commande (local via LabVIEW, distant via web, ou automatique), une gestion de priorité logicielle a été mise en œuvre. En résumé :

- Le mode automatique, est prioritaire en l'absence de commande manuelle.

- Le mode local (LabVIEW) peut temporairement suspendre le mode automatique si activé.
- Le mode distant (page web) permet une commande ponctuelle, qui repasse ensuite au mode automatique.

Cette logique garantit un comportement cohérent du système, quel que soit le canal utilisé.

4.3.4. Choix technologiques et justifications

Supervision	Choix retenu	Justification
Supervision locale	LabVIEW	Interface graphique rapide, fiable
Supervision distante	Interface web personnalisée	Accessible depuis n'importe quel navigateur
Cloud IoT	ThingSpeak	Intégration simple, visualisation fluide
Réseau	Wi-Fi local sécurisé	Communication rapide et économique

Tableau 1 : Choix technologiques

4.4. Intégration avec la plateforme cloud (ThingSpeak)

Avant d'exploiter les données dans un tableau de bord dynamique, une série d'étapes techniques ont été réalisées pour assurer la communication entre l'ESP32 et la plateforme cloud ThingSpeak. Cette phase a nécessité à la fois une configuration côté cloud et une programmation embarquée en C++.

1) Programmation de l'ESP32 en C++

L'ESP32 a été programmé en langage C++ pour :

- ✓ Lire les données en temps réel à partir des capteurs
- ✓ Se connecter à un réseau Wi-Fi.
- ✓ Envoyer périodiquement les données vers la plateforme via des requêtes.

Cette programmation assure l'automatisation de la collecte et la transmission des données selon une fréquence définie.

2) Création d'un Channel

Une fois connecté, un Channel (canal) a été créé. Celui-ci joue le rôle d'espace de stockage et d'affichage des données. Chaque canal contient plusieurs champs configurables

3) Récupération de la clé d'écriture (Write API Key)

Chaque canal dispose d'une clé API d'écriture sécurisée (Write API Key). Cette clé est insérée dans le code embarqué de l'ESP32 pour authentifier chaque tentative d'envoi de données vers le serveur ThingSpeak.

4) Test de communication et affichage initial

Une fois toutes les configurations effectuées :

- ✓ Le microcontrôleur a été mis sous tension.
- ✓ La communication Wi-Fi a été vérifiée via le moniteur série.
- ✓ Les premières valeurs ont été envoyées à distance vers ThingSpeak.
- ✓ Le canal a été actualisé automatiquement à chaque envoi

4.4.1. Résultats de l'intégration en temps réel

Dans cette section, nous présentons les résultats issus de la phase d'intégration en temps réel du système connecté. Les données affichées proviennent directement des capteurs installés et sont transmises de manière continue vers notre plateforme. Les graphiques suivants illustrent l'évolution des différentes variables clés au cours d'une période de précise, permettant d'observer la réactivité du système et la cohérence des informations mesurées.

Les résultats constituent une base essentielle pour évaluer le bon fonctionnement du système embarqué, la fiabilité des capteurs et l'efficacité des décisions automatisées prises par les actionneurs.

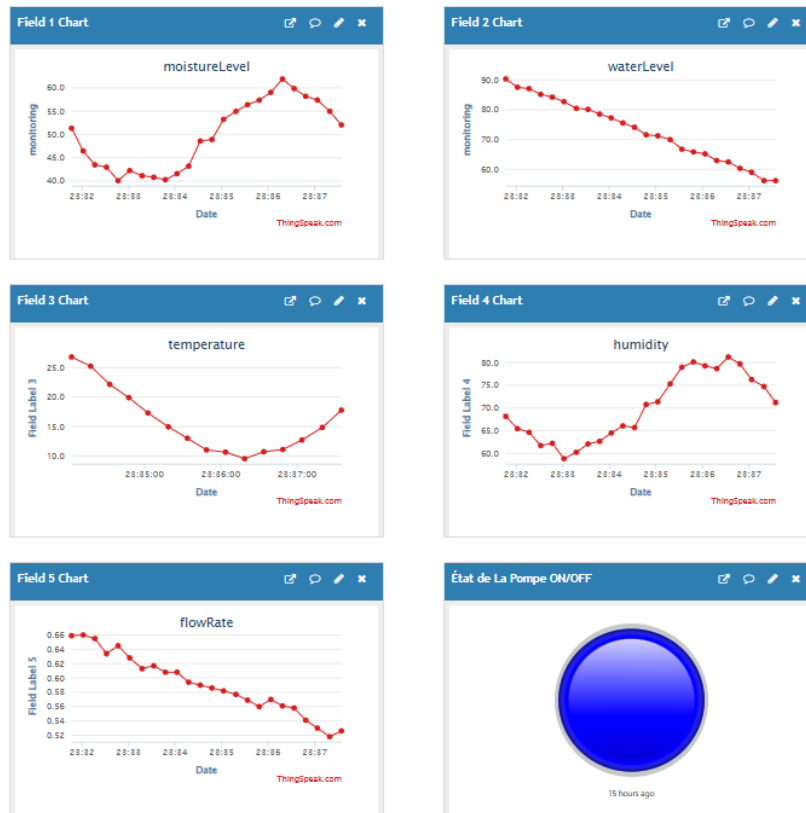


Figure 13 : Résultat d'acquisition en ligne en temps réel

4.4.2. Interprétation d'acquisition des données temps réel

1) Humidité du sol (moistureLevel - Field

L'humidité du sol varie de ~42 % à ~60 %, avec une tendance montante suivie d'une légère chute.

Le capteur répond de façon dynamique aux variations. Cette situation indique que les mesures sont cohérentes et reflètent bien les changements d'humidité. Cela témoigne d'un bon étalonnage et d'un système de captation réactif.

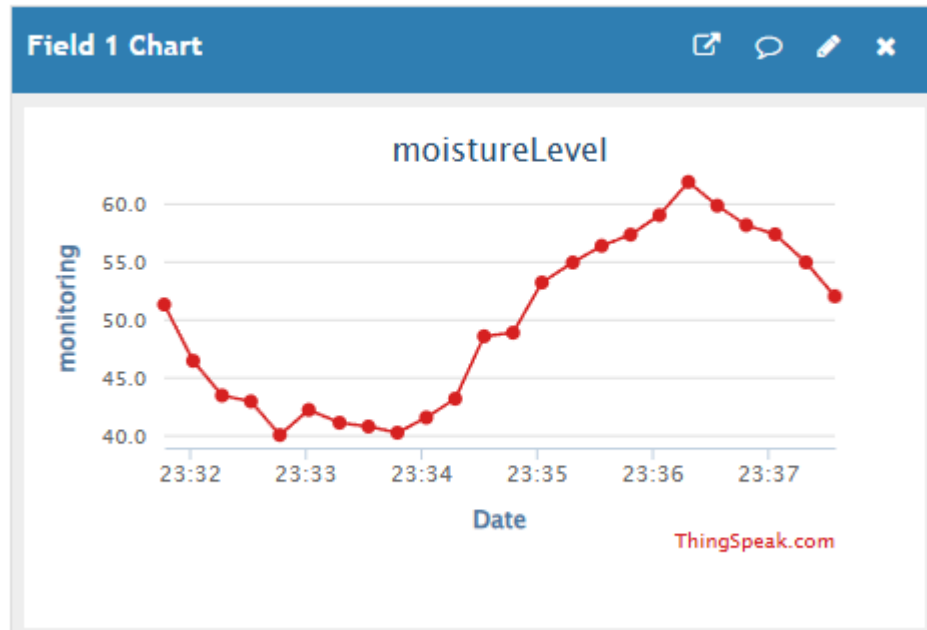


Figure 14 : Résultat d'acquisition en temps réel d'humidité du sol

2) Niveau d'eau (field2)

Plage observée : ~80 % à 50 %, On observe une diminution progressive, ce qui reflète une consommation régulière d'eau par le système. Cela montre que le système fonctionne et que le niveau d'eau baisse logiquement (processus d'irrigation en cours).

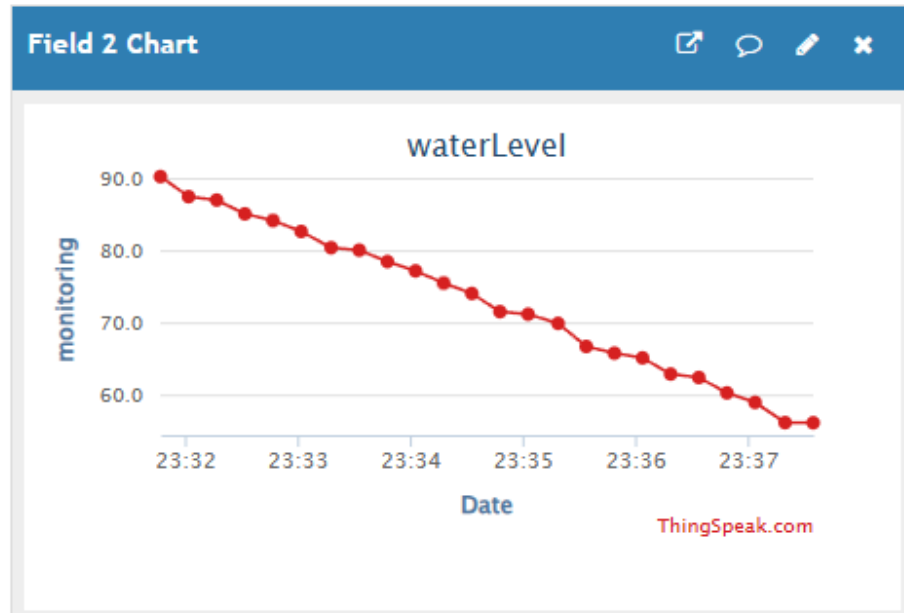


Figure 15 : Résultat d'acquisition en temps réel de niveau d'eau

3) Pression d'eau (flowRate- field5)

Pression descendante de ~ 0.66 à ~ 0.52 bar, Cette chute graduelle est cohérente avec la baisse du niveau d'eau, ce qui valide le lien entre pression et disponibilité. Aucune coupure brutale n'est détectée, donc le système d'alimentation est opérationnel et stable.

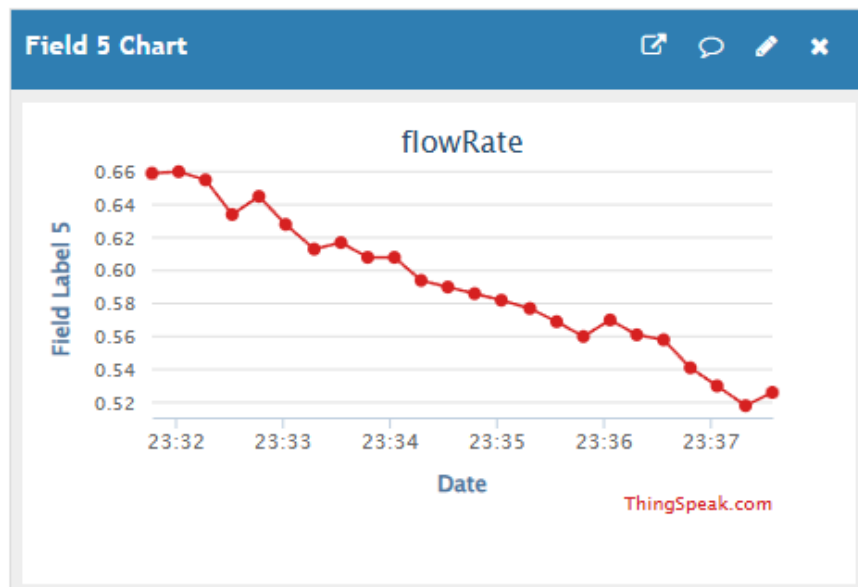


Figure 16 : Résultat d'acquisition en temps réel la pression de l'eau

4) Température ambiante (field3)

Variation entre $\sim 15^{\circ}\text{C}$ et $\sim 28^{\circ}\text{C}$, le système montre une courbe équilibrée, ce qui indique une bonne stabilité thermique et l'absence de pics anormaux

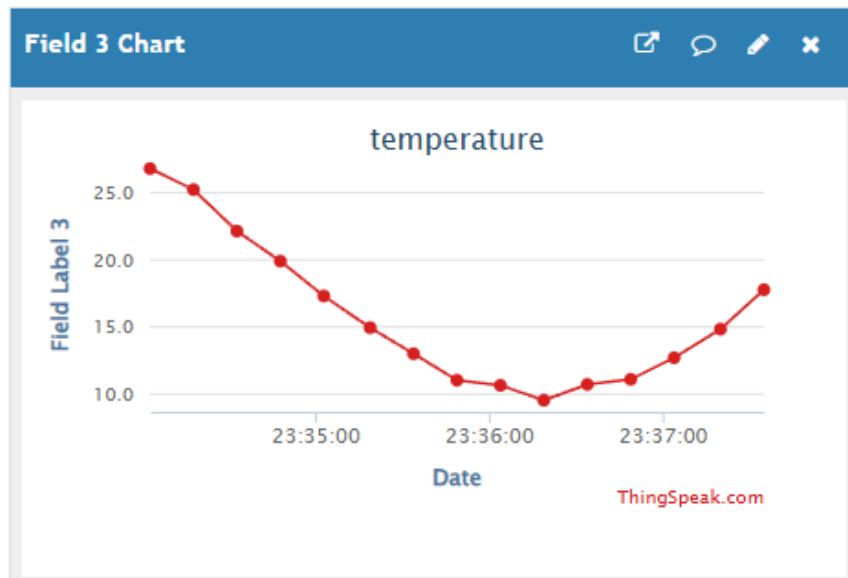


Figure 17 : Résultat d'acquisition en temps réel de température ambiante

5) Humidité ambiante (humidity - Field 4)

Variation entre ~62 % et ~78 %, en corrélation avec la température, Les valeurs restent dans une plage cohérente, sans instabilité.

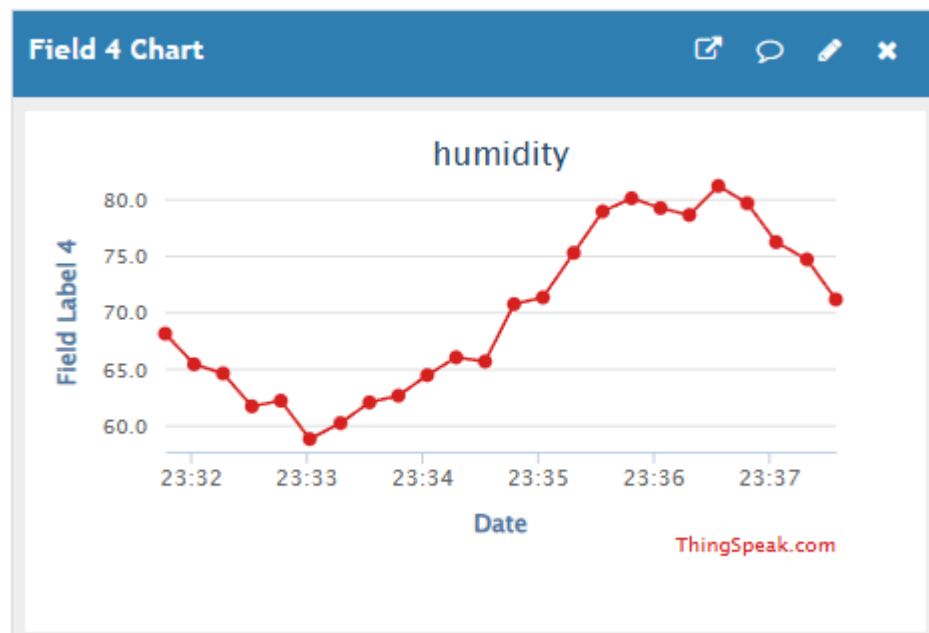


Figure 18 : Résultat d'acquisition en temps réel de l'humidité

6) État de la pompe (ON/OFF)

L'interface affiche une pompe active (bouton Bleu), Le système agit automatiquement en fonction des données collectées. L'activation automatique confirme que les seuils définis sont atteints et que la commande fonctionne en temps réel.

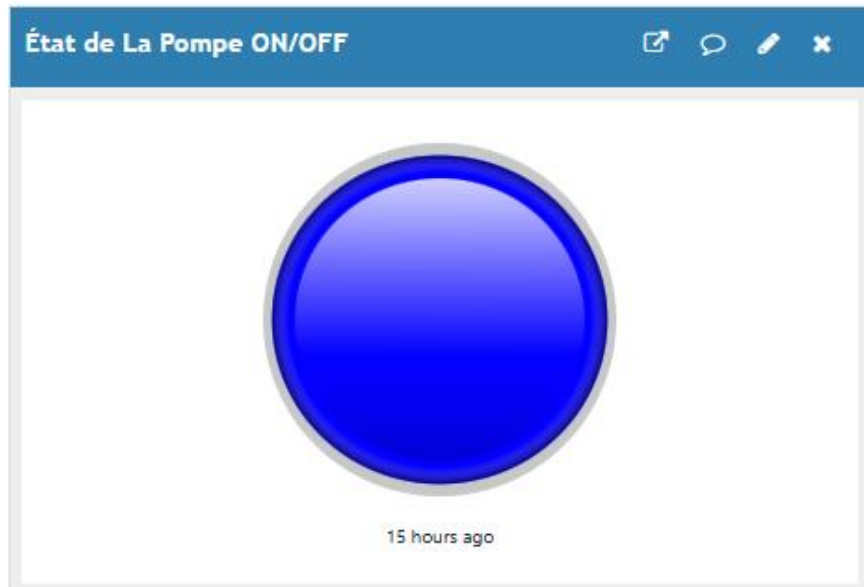


Figure 19 : Résultat d’acquisition en temps réel de l’état de la pompe

4.4.3. Conclusion globale et Portée des résultats

Les courbes et l’indicateur montrent un système opérationnel, réactif et autonome. Chaque capteur délivre des valeurs dynamiques, cohérentes et bien corrélées, prouvant que :

- ✓ Les mesures sont valides.
- ✓ La logique de régulation et d’automatisation est fonctionnelle.
- ✓ Le système est apte à assurer une supervision fiable et continue.

4.5. Développement de l'interface Web embarquée

La conception de l'interface Web embarquée a représenté une étape essentielle du projet, combinant savoir-faire en électronique, développement embarqué et design Web. Ceci constitue une passerelle intelligente entre l'utilisateur et le système intelligent, et a été développé pour allier son efficacité

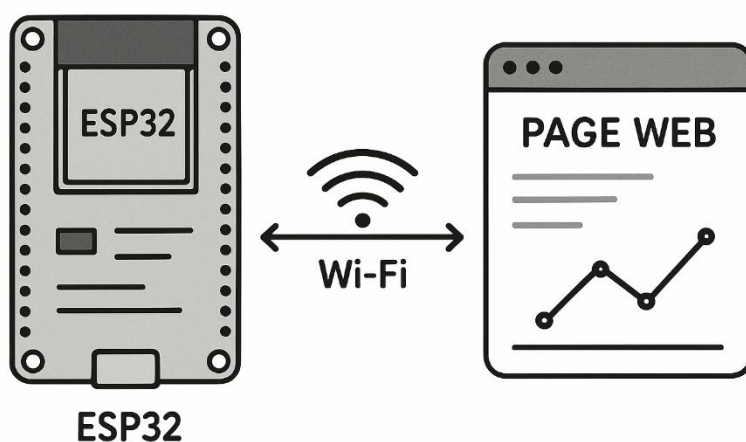


Figure 20 : Liaison de l'esp32 avec une page web

4.5.1. Environnement de développement

Le développement a été réalisé dans l'environnement Arduino IDE, couplé à l'utilisation de bibliothèques spécifiques pour la gestion réseau :

- **ESPAsyncWebServer.h** : Bibliothèque permettant la création d'un serveur Web non-bloquant et performant, idéal pour les systèmes embarqués.

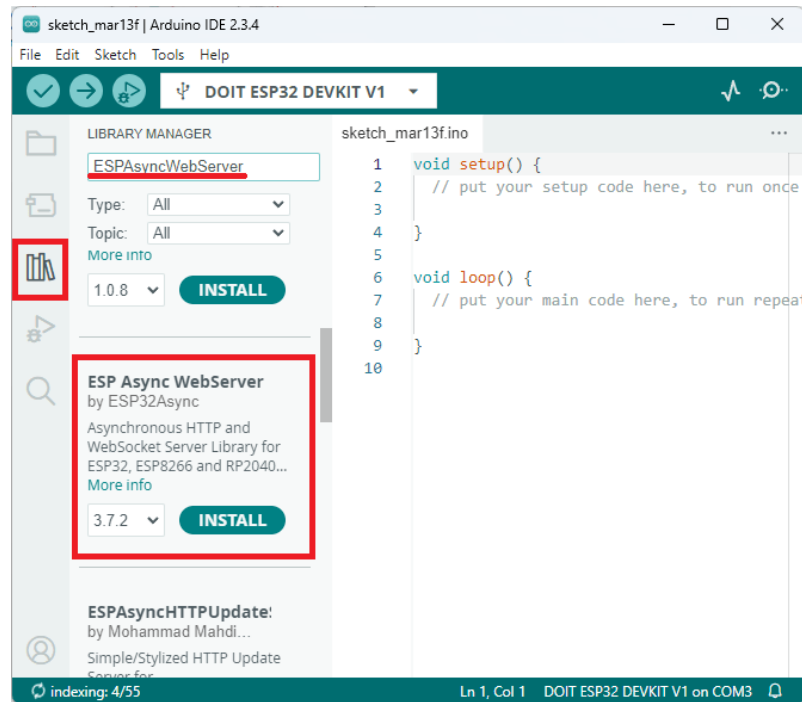


Figure 21 : La bibliothèque ESPAsyncWebServer

- **WiFi.h** : Pour établir et gérer la connexion au réseau local via le module Wi-Fi intégré de l'ESP32.
- **ArduinoJSON.h** : Pour formater les réponses de données en JSON et les rendre lisibles côté client.

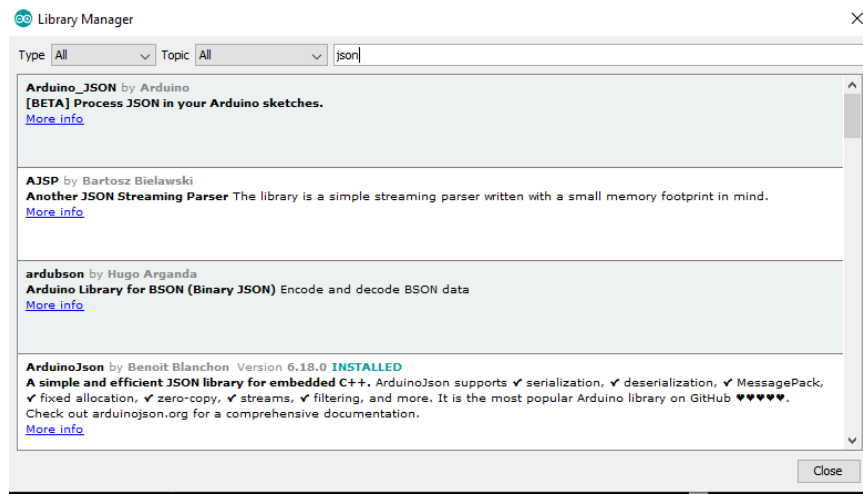


Figure 22 : La bibliothèque ArduinoJSON.h

- **HTML, CSS et JavaScript embarqués** directement dans le code C++, pour ne pas dépendre de fichiers externes.

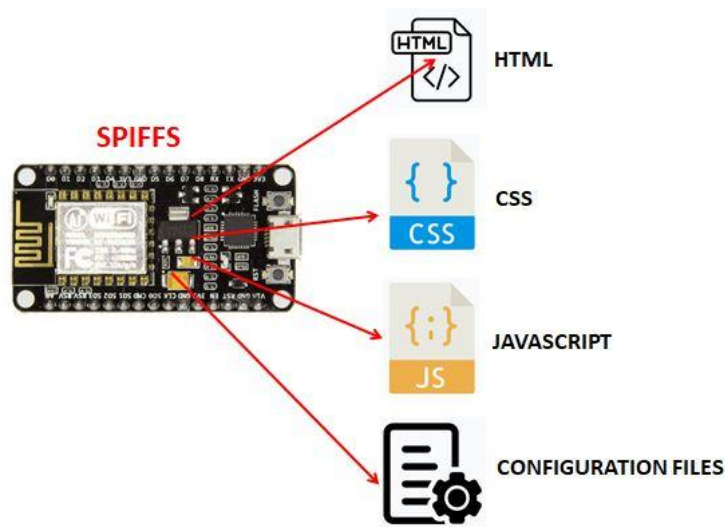


Figure 23 : Structure Fichiers Web et ESP32

4.5.2. Conception de l'interface : légèreté et réactivité

Le design de la page Web embarquée repose sur les principes suivants :

- ✓ Simplicité visuelle : chaque élément (bouton, texte, indicateur) est positionné de manière intuitive pour permettre une interaction directe sans ambiguïté.
- ✓ Réactivité immédiate : l'utilisation de JavaScript asynchrone (AJAX) permet de rafraîchir les données des capteurs toutes les quelques secondes sans recharger la page, garantissant une expérience fluide et moderne.
- ✓ Accessibilité multiplateforme : la page est accessible depuis n'importe quel navigateur, ce qui en fait un outil pratique, universel et économique.

1) Fonctionnement interne : cycle de requêtes et réponses

Le serveur Web traite les interactions selon un modèle de requêtes HTTP, que ce soit pour :

- Lire les données des capteurs (GET /data)
- Changer le mode de fonctionnement (POST /mode)
- Allumer ou éteindre la pompe (POST /pump/on, POST /pump/off)

2) Interface Web

Zone de Contrôle – Commande de la Pompe

Mode de fonctionnement : La mention « Mode : Automatique » informe l'utilisateur que le système gère actuellement la pompe en autonomie, selon les seuils et conditions environnementales.

- Bouton “Passer en Manuel” : Ce bouton permet de désactiver la gestion automatique et de prendre la main manuellement sur l'activation de la pompe. Ce basculement est instantané, offrant une grande souplesse d'utilisation.

- Bouton “Éteindre la Pompe” : Lorsque l'utilisateur est en mode manuel, ce bouton lui permet de désactiver immédiatement la pompe, assurant une réactivité et une précision dans la gestion de l'eau.

Ces boutons sont mis en valeur visuellement grâce à un design vert vif, synonyme d'action positive, de contrôle et de sécurité.

3) Zone d'Acquisition – Données des Capteurs

La seconde partie de la page offre une vue synthétique mais détaillée des informations captées en temps réel dans l'environnement :

- ✓ Humidité du Sol
- ✓ Niveau d'Eau
- ✓ Température et Humidité de l'Air
- ✓ Débit d'Eau

L'image ci-dessous donne un aperçu de l'organisation globale des éléments constituant l'interface web.

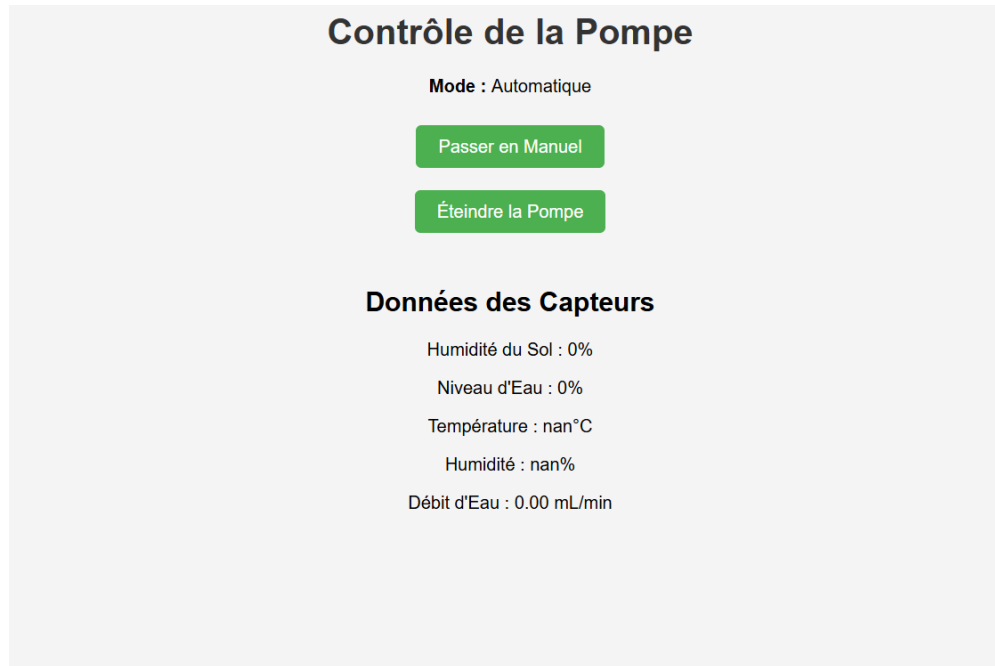


Figure 24 : État général de la page web

4) Bilan et apport fonctionnel

Cette interface a été conçue pour allier simplicité, performance et autonomie. Elle constitue un point fort du système, car elle :

- Permet une supervision en temps réel depuis n'importe quel terminal connecté.
- Offre une commande souple et rapide sur les actionneurs.
- Centralise les informations pour une prise de décision immédiate.
- Renforce l'autonomie du système sans infrastructure cloud ni dépendances externes.

L'image ci-dessous présente l'état général de la page web développée au moment du fonctionnement actif du système, permettant la visualisation et le contrôle des paramètres.



Figure 25 : État général de la page web lors du fonctionnement du système

4.5.3. Conclusion

L'implémentation d'une page Web embarquée sur l'ESP32 constitue une avancée significative dans le cadre de ce projet

Grâce à cette solution, notre système peut être utilisé à distance, sans avoir besoin d'une installation compliquée ou de matériel supplémentaire. On peut facilement contrôler les actionneurs et suivre les données des capteurs en temps réel, simplement en ouvrant une page Web claire et facile à utiliser.

Ce développement rend notre projet plus autonome, il montre aussi que l'ESP32 est une excellente solution, il peut lire les capteurs, contrôler des éléments, traiter les données et les partager directement sur le Web.

4.6. Supervision et commande locale avec LabVIEW

Dans le cadre de ce projet de conception d'un système embarqué dédié à l'agriculture intelligente, nous avons intégré une supervision locale à l'aide du logiciel industriel LabVIEW (Laboratory Virtual Instrument Engineering Workbench), un environnement de développement graphique reconnu, développé par National Instruments. Ce choix technologique s'est imposé grâce à sa capacité à offrir une programmation graphique, tout en répondant aux exigences de robustesse des environnements industriels.

L'approche retenue repose sur un système hybride, combinant la puissance de traitement local de notre microcontrôleur avec la flexibilité de l'interface graphique LabVIEW. Le microcontrôleur assure la collecte et la transmission des données issues des capteurs, tandis que LabVIEW prend en charge la visualisation en temps réel, la supervision locale et le pilotage des actionneurs.

LabVIEW s'appuie sur une architecture robuste, souvent déployée sous Linux, tout comme notre microcontrôleur qui fonctionne lui aussi sur un environnement Linux embarqué. En combinant ces deux environnements, nous avons pu mettre en place une solution hybride efficace, réunissant la supervision industrielle avec LabVIEW et le traitement embarqué en temps réel sur le microcontrôleur.

4.6.1. Programmation graphique et Conception du bloc diagramme

La mise en œuvre du système dans LabVIEW commence par la conception du bloc diagramme, qui représente la logique interne du programme.

Pour concevoir un système cohérent et fonctionnel, plusieurs étapes clés doivent être respectées :

- **Initialisation et configuration du port série**

Le microcontrôleur, qui centralise les données provenant des capteurs envoie ces valeurs par le port série. Une structure d'initialisation configure les paramètres de communication (port COM, débit de transmission, bits de contrôle, etc.), assurant une liaison robuste et constante.

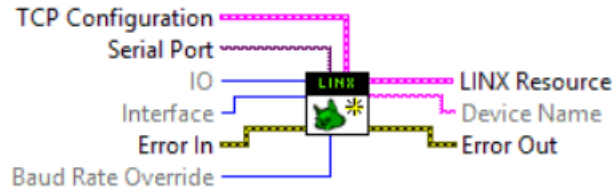


Figure 26 : Schéma graphique de l'initialisation des paramètres de communication

- **Boucle d'acquisition continue**

Une boucle while gère l'acquisition continue, avec des délais ajustables pour éviter la surcharge du processeur. À chaque itération, une lecture de chaîne série est effectuée.

Cette méthode assure une synchronisation fiable entre les données du microcontrôleur et l'affichage sous LabVIEW.

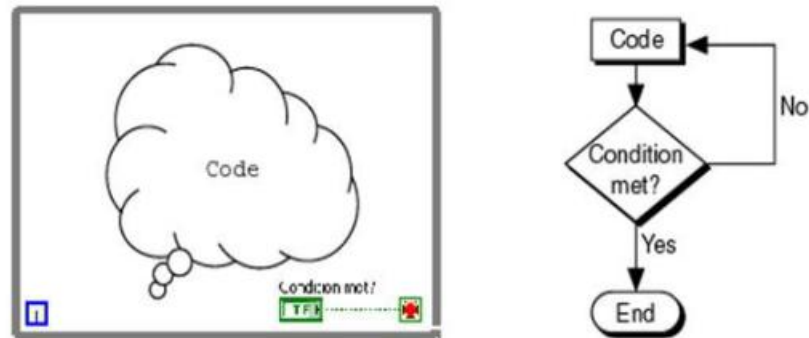


Figure 27 : Schéma de la boucle while dans labview [42].

- **Mise à jour dynamique des indicateurs**

Chaque variable extraite est directement liée aux composants visuels du Front Panel (jauges, champs numériques, graphiques). Le système assure donc une mise à jour en temps réel de l'état environnemental surveillé.

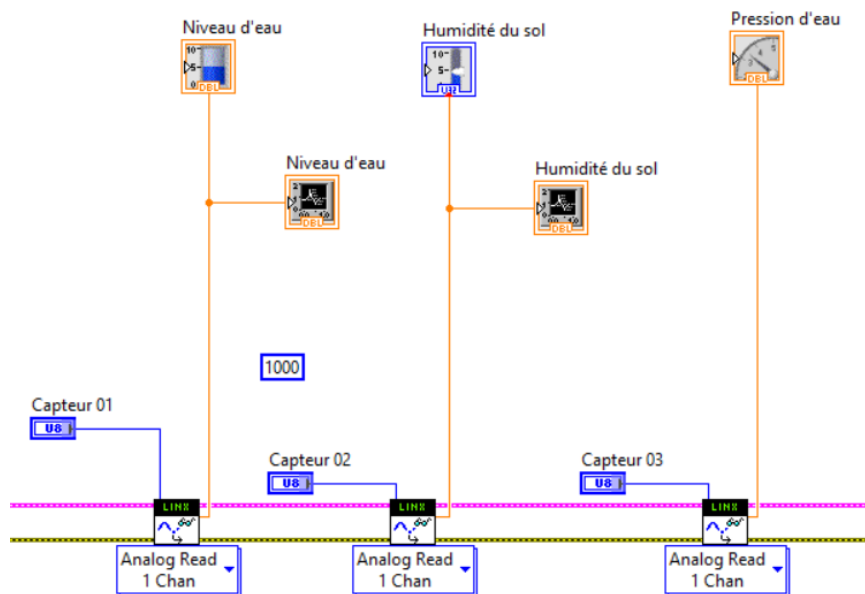


Figure 28 : Programmation graphique des composant et des indicateurs dans LabVIEW

- **Commande de la pompe d'irrigation**

Un module de commande, intégré dans le diagramme, permet à l'utilisateur d'envoyer un signal de commande (ON/OFF) vers le microcontrôleur via un simple clic sur l'interface LabVIEW. Cette instruction est traduite en un caractère ou une séquence reconnue par le microcontrôleur, qui active ou désactive la pompe en conséquence.

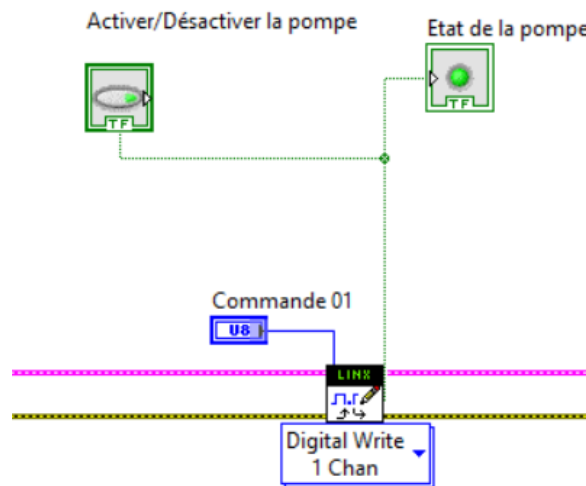


Figure 29 : Programmation graphique de sortie de commande dans LabVIEW

- **Gestion des erreurs et arrêt sécurisé**

Des blocs spécifiques gèrent les erreurs potentielles (perte de communication, mauvaise lecture, etc.) et garantissent un arrêt contrôlé du système via un bouton "STOP", préservant la sécurité du matériel et des données.

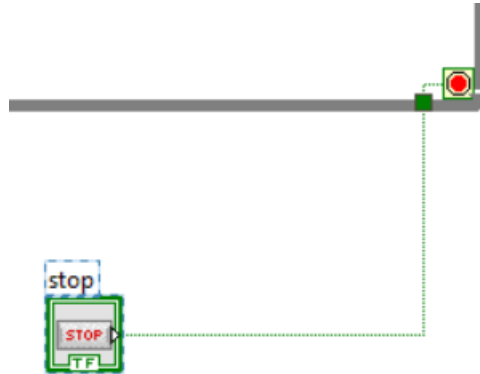


Figure 30 : Programmation graphique de l'arrêt sécurisé dans LabVIEW

Au terme de cette conception, le bloc diagramme final reflète toute la structure fonctionnelle du système,

L'illustration suivante présente cette architecture graphique dans son ensemble, démontrant l'organisation et l'efficacité du système développé.

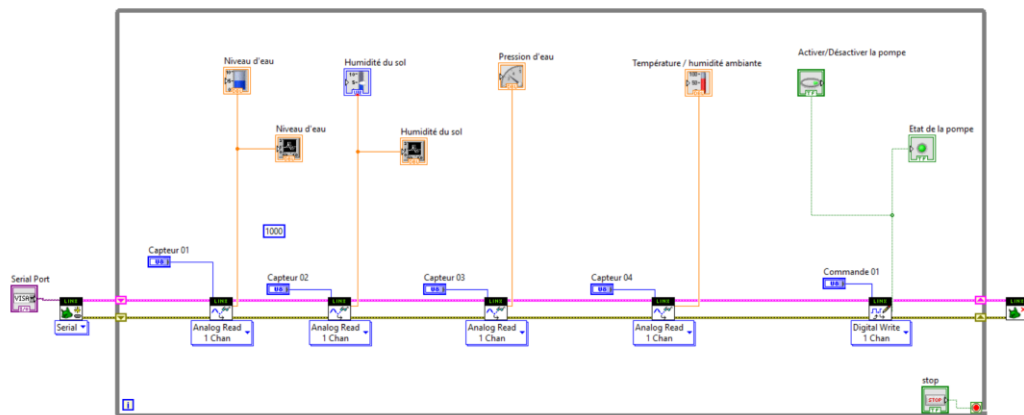


Figure 31 : L'ensemble de l'architecture graphique de notre système

4.6.2. Développement de l'interface utilisateur (Front Panel)

Le Front Panel de LabVIEW représente une interface de supervision active et joue un rôle crucial dans l'interaction avec le système. Conçu de manière conviviale, il reflète les principes fondamentaux des interfaces de notre système

- **Affichage en temps réel des mesures**

Le panneau intègre des modules de visualisation correspondant au système connecté au microcontrôleur :

- ✓ Affichage vertical avec curseur pour le niveau d'eau
- ✓ Affichage vertical avec curseur pour l'humidité du sol
- ✓ Indicateur circulaire de type manomètre pour la pression d'eau
- ✓ Thermomètre graphique pour la température ambiante
- ✓ Indicateur LED pour visualiser l'état de la pompe (activée ou désactivée)
- ✓ Graphiques temps réel pour le suivi des capteurs de niveau d'eau et d'humidité du sol
- ✓ Bouton graphique poussoir pour activer/désactiver manuellement la pompe
- ✓ Champs numériques pour afficher les valeurs de chaque capteur en temps réel

Chaque composant est mis à jour en continu grâce aux valeurs reçues, offrant une visualisation instantanée de l'état de notre système.

- **Commande manuelle de la pompe d'irrigation**

Le bouton de commande permet de piloter directement la pompe d'irrigation à travers l'interface. Ce bouton agit comme un déclencheur d'événement dans le bloc diagramme, envoyant une instruction série au microcontrôleur.

L'image ci-dessous montre le Front Panel final développé sous LabVIEW

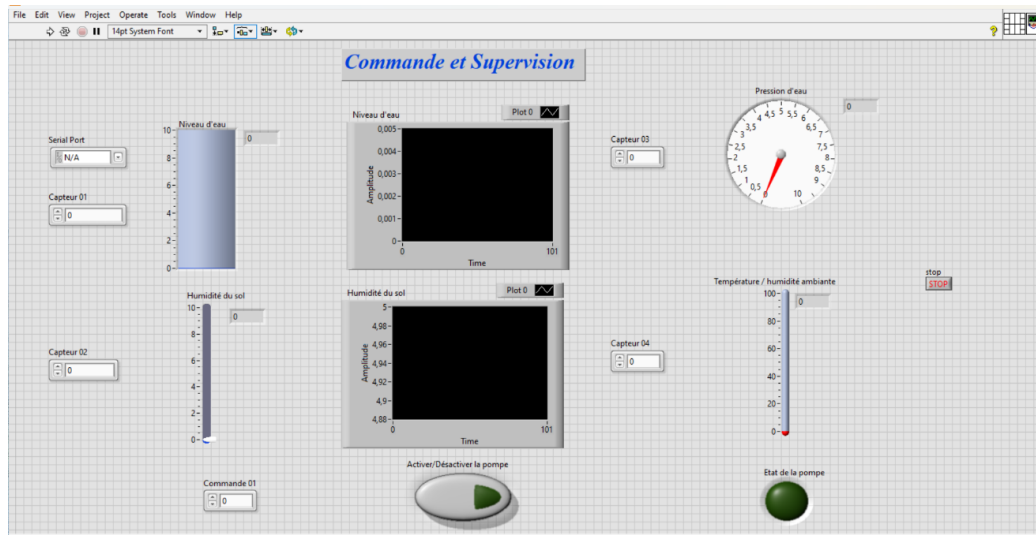


Figure 32 : Le schéma final du digramme bloc fonctionnel

4.6.3. Téléversement et visualisation des résultats

Une fois la conception du bloc diagramme terminée et l'interface utilisateur (Front Panel) entièrement configurée, la dernière étape consiste à téléverser le code sur le microcontrôleur et à établir la communication série avec LabVIEW.

Dès que la connexion est active, les données des capteurs commencent à s'afficher en temps réel sur le Front Panel :

- L'humidité du sol,
- Le niveau d'eau,
- La pression de l'eau,
- La température et l'humidité ambiantes.

En parallèle, les commandes logiques intégrées dans le bloc diagramme permettent d'activer automatiquement ou manuellement la pompe, en fonction des conditions mesurées.

Cette phase de test en conditions réelles valide l'ensemble du système :

- Le bon fonctionnement de la communication entre le microcontrôleur et LabVIEW,
- La réactivité de l'affichage des données,
- La fiabilité du contrôle des actionneurs.

L'image ci-dessous illustre une partie du fonctionnement du système en temps réel, tel qu'observé sur l'interface LabVIEW.

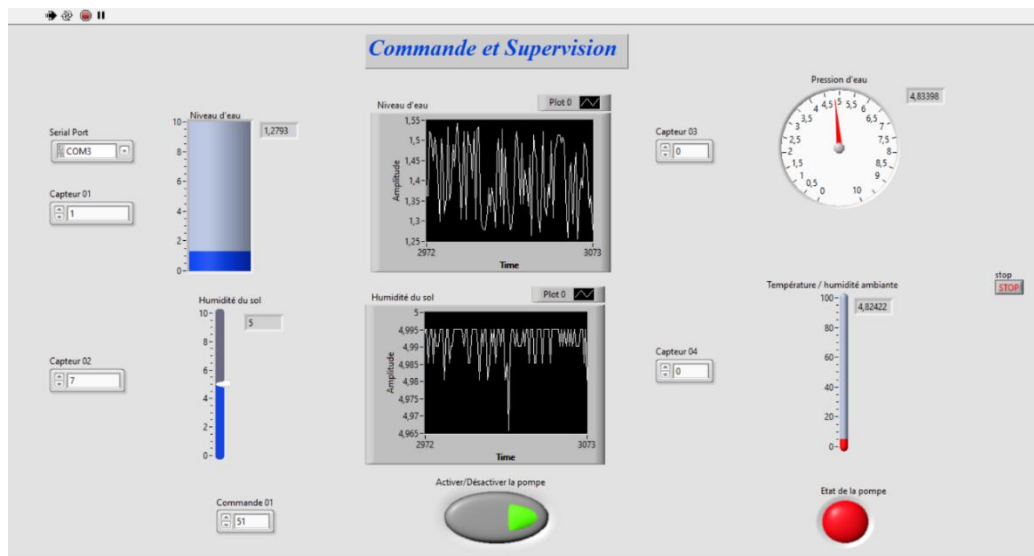


Figure 33 : État du système lors de l'exécution

4.6.4. Conclusion

L'intégration hybride entre LabVIEW et le microcontrôleur a permis de profiter de point fort de chacun : la puissance de traitement et de visualisation de LabVIEW d'un côté, et la réactivité d'un système embarqué de l'autre.

Ce travail illustre parfaitement comment une solution industrielle peut être adaptée un système embarqué léger avec une commande et supervision professionnel et fiable.

Chapitre 5

Intégration de l'IA et de l'apprentissage profond pour la détection des anomalies

5.1. Introduction

Ce chapitre présente l'intégration de modèles d'apprentissage profond basés sur la vision par ordinateur pour la détection automatique d'anomalies en temps réel. Les modèles développés permettent d'identifier avec précision des fissures dans les tuyaux, des maladies sur les plantes, ainsi que la qualité du sol pour l'irrigation automatisé par IA.

Les modèles ont été conçus, entraînés et testés dans un environnement combinant Google Colab pour le traitement et l'entraînement, et Roboflow pour l'annotation, l'optimisation et l'hébergement. Une fois les performances validées, ils ont été intégrés dans la plateforme Roboflow de détection en ligne et en temps accessible via un site personnalisé, ainsi qu'à travers une API indépendante.

Le travail réside dans l'utilisation de modèles à la fois légers et performants, avec une précision élevée. À cela s'ajoute une double intégration technique : d'une part, une détection visuelle en temps réel accessible depuis n'importe quel emplacement via une interface en ligne, et d'autre part, Une API dédié accessible en ligne permettant d'analyser instantanément les résultats de détection d'anomalies sur une image ou une vidéo fournie.

5.2. Cadre Méthodologique Général

Dans le cadre de ce travail, trois modèles de détection ont été conçus et déployés pour résoudre des problématiques spécifiques en agriculture intelligente. Bien que chaque modèle traite une tâche différente, ils partagent une méthodologie commune, notamment en matière de préparation des données, d'environnement de développement et de critères d'évaluation. Cette section présente donc les étapes techniques globales appliquées à l'ensemble des modèles, avant de détailler individuellement chaque cas.

5.2.1. Base des données

Le jeu de données utilisé provient de la plateforme Roboflow, qui offre une interface efficace pour le téléchargement, l'annotation et l'exportation de jeux de données. Chaque image a été annotée manuellement avec une boîte englobante précise pour localiser la classe. Le dataset a été divisé en trois sous-ensembles : 70% pour l'entraînement, 20% pour la validation, et 10% pour les tests.

5.2.2. Augmentation des données

Pour améliorer la diversité du jeu de données et augmenter la capacité de généralisation du modèle, des techniques d'augmentation d'image ont été appliquées à l'aide de (la bibliothèque python keras ImageDataGenerator). Celles-ci incluent la rotation, le retournement horizontal, les modifications de luminosité, l'ajout de bruit et le zoom. Grâce à ces transformations, le nombre total d'images a été augmenté d'environ 30%, permettant au modèle d'apprendre à partir de variations visuelles plus riches.

5.2.3. Réglage des hyperparamètres

Les hyperparamètres des modèles ont été ajustés afin d'optimiser ses performances, avec les réglages suivants : un taux d'apprentissage de 0.001, un nombre d'époques fixé à 100, une taille de lot de 16, des images redimensionnées en 640x640 pixels, et l'utilisation de l'optimiseur SGD.

Ces réglages ont été affinés après plusieurs essais, en analysant les courbes de perte, de précision et de rappel à chaque itération.

5.2.4. Évaluation des modèles

L'évaluation des modèles a été réalisée à l'aide de plusieurs métriques standards de détection : précision, rappel, F1-score, mAP@50 et mAP@50-95.

5.3. Détection de la qualité du sol à l'aide du model yolo v11

L'algorithme YOLO continue de progresser avec l'arrivée de YOLOv11, une version améliorée qui marque une nouvelle étape dans la détection d'objets en temps réel. Cette version reprend les points forts des versions précédentes tout en apportant des nouveautés qui la rendent encore plus performante et polyvalente.

YOLOv11 se distingue par sa grande flexibilité, capable de gérer non seulement la détection d'objets, mais aussi d'autres tâches de vision par ordinateur comme l'estimation de pose ou la segmentation d'instances. Grâce à cette conception plus complète, le modèle devient utilisable dans une variété de domaines. Son architecture a été pensée pour offrir un bon compromis entre puissance et simplicité d'utilisation, permettant de relever des défis complexes avec efficacité et précision [43].

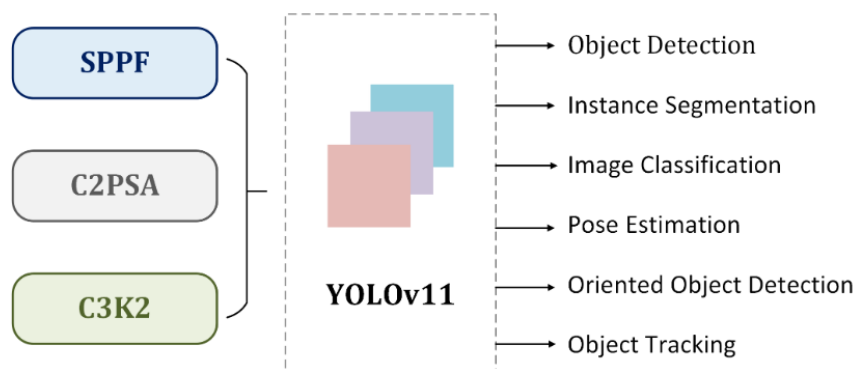


Figure 34 . Architecture YOLOv11 [44].

L'objectif de cette première application est de détecter visuellement l'état d'humidité du sol à partir d'images, dans un contexte agricole. L'état du sol est catégorisé en trois classes : sol sec, sol modérément humide et sol très humide. Cette information est essentielle pour automatiser la prise de décision en matière d'irrigation.

5.3.1. Spécificité du Dataset

Les données utilisées pour cette tâche proviennent de captures terrain et ont été organisées à l'aide de la plateforme Roboflow. Chaque image a été annotée selon la nature visible du sol, permettant de distinguer les trois états d'humidité. Les annotations ont été effectuées en format YOLO, c'est-à-dire par des boîtes englobantes associées à une classe. Les images représentent différentes conditions de luminosité, d'angle et de texture du sol, assurant ainsi une diversité suffisante pour l'entraînement du modèle.



Figure 35 : Quelques images de la base des données du qualité du sol

5.3.2. Résultats et discussions

Après entraînement du modèle YOLOv11 et intervention au niveau des hyperparamètres, les résultats ont montré une très bonne capacité de détection pour les trois classes. Les courbes de loss ont convergé de manière stable, et les métriques finales sont les suivantes:

Précision globale	95.5%
Rappel global	98.4%
mAP@50	98.1%
mAP@50-95	88.2%

Tableau 2 : Métriques d'évaluation du YOLOv11

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	61	61	0.955	0.984	0.981	0.882
tanah aluvial	21	21	0.998	0.952	0.983	0.903
tanah humus	25	25	0.956	1	0.977	0.872
tanah liat	15	15	0.91	1	0.982	0.87

Figure 36 : Résultats d'entraînement du model

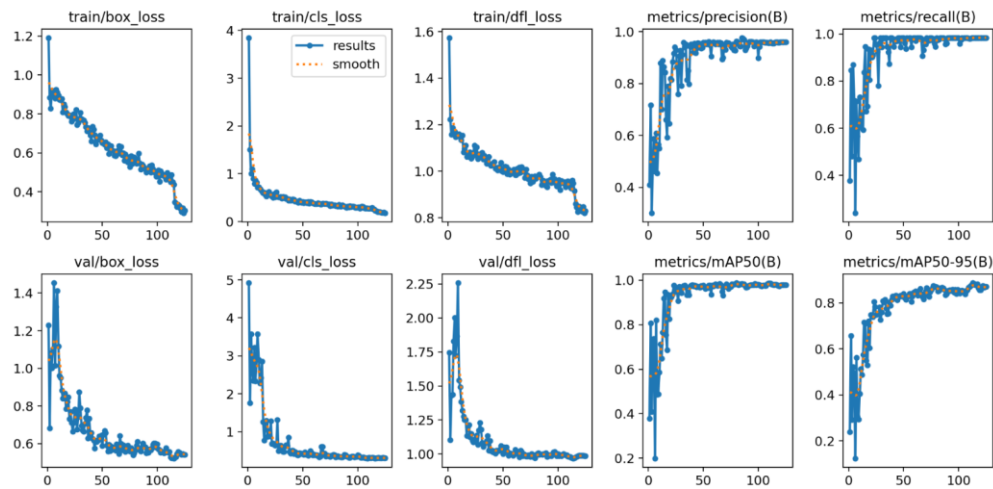


Figure 37 : Courbes d'apprentissage du mode

5.3.3. Interprétation des résultats

1) Courbe des pertes :

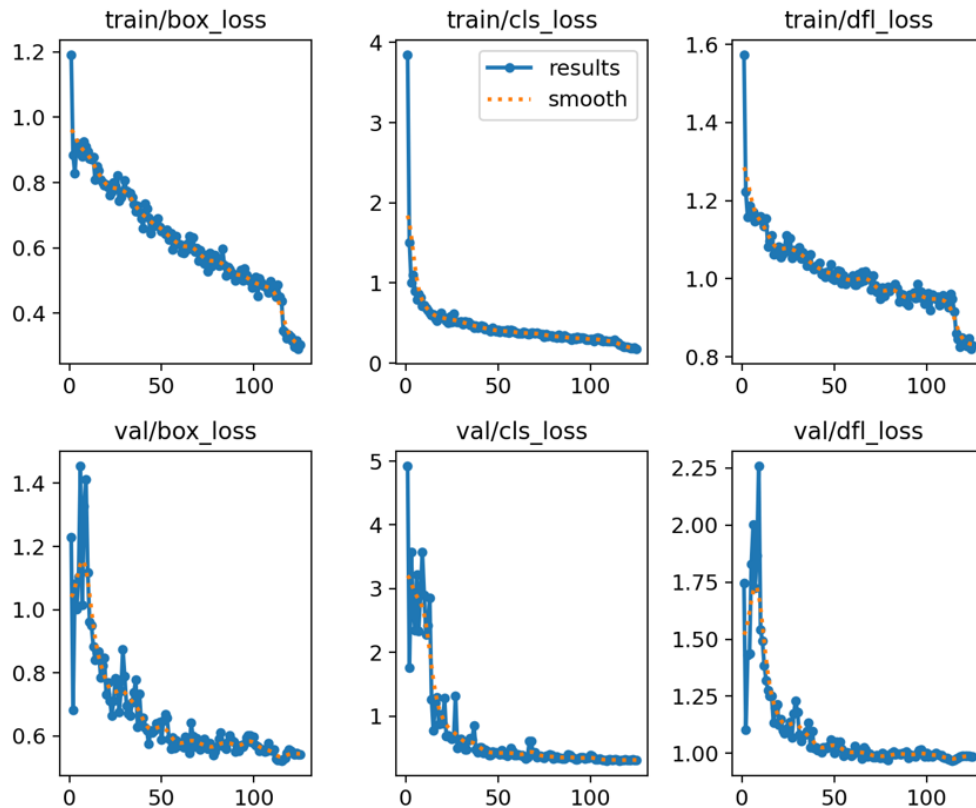


Figure 38 : Les courbes des pertes

Les courbes de perte représentent l'évolution de l'erreur du modèle pendant l'entraînement et la validation.

On observe que toutes les courbes d'entraînement (train/..._loss) suivent une tendance décroissante continue, signe que le modèle apprend efficacement au fil des itérations.

La box_loss diminue régulièrement, ce qui montre une amélioration progressive de la capacité du modèle à bien localiser les objets.

La cls_loss chute très rapidement, ce qui indique que le modèle comprend rapidement comment distinguer les différentes classes de sol (sec, modéré, humide).

La `dfl_loss`, qui mesure la précision de localisation fine des boîtes, montre aussi une amélioration constante.

Du côté des courbes de validation (`val/..._loss`), on note qu'elles suivent des trajectoires très similaires aux courbes d'entraînement, avec une diminution progressive.

Conclusion :

Le modèle généralise bien, sans surapprentissage. L'entraînement est stable, cohérent, et les pertes convergent correctement, ce qui est un indicateur positif pour la fiabilité du modèle.

2) Courbes de Précision et de Rappel

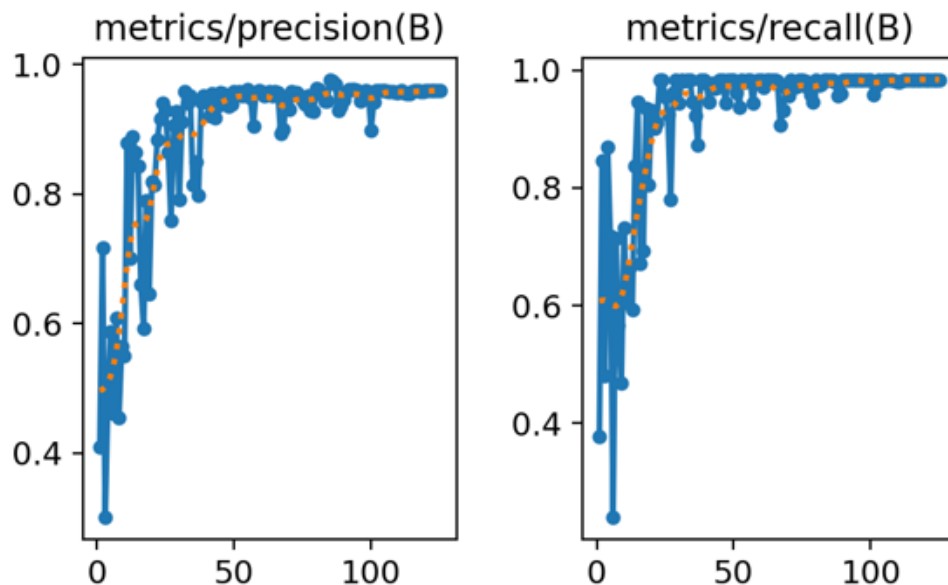


Figure 39 : Les courbes de Précision et de Rappel

Ces deux courbes permettent d'évaluer la qualité des prédictions :

La précision (precision) mesure la part des détections qui sont correctes. Elle monte rapidement vers 0.96, ce qui signifie que le modèle fait très peu de fausses détections.

Le rappel (recall) évalue combien d'objets réels sont bien détectés. Il atteint presque 0.99, indiquant que presque tous les objets présents sont captés par le modèle.

Ces courbes montrent une progression rapide et une stabilisation haute, ce qui est un signe fort d'un modèle performant et équilibré.

Conclusion :

Le modèle présente une excellente capacité à détecter et classifier correctement sans confondre.

3) Courbes mAP

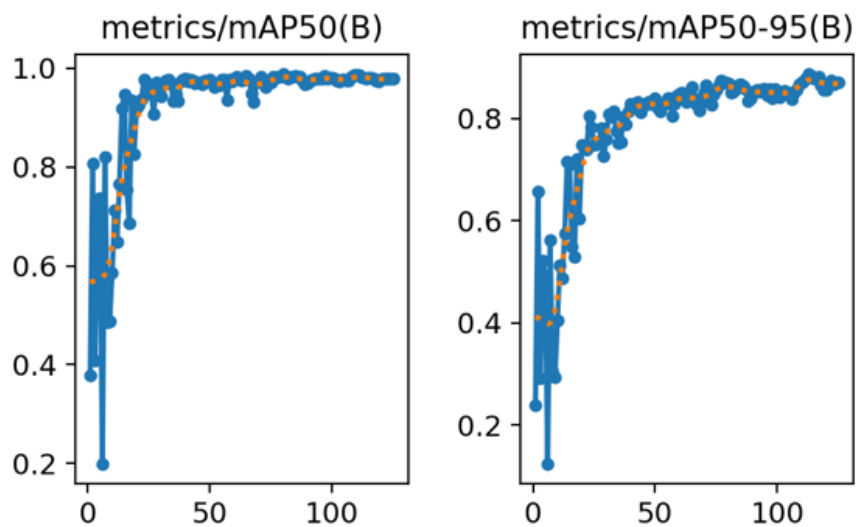


Figure 40 : Les courbes mAP

La mAP (mean Average Precision) est l'une des métriques les plus complètes pour juger la performance d'un modèle de détection.

- La $mAP@50$ atteint très vite 98%, ce qui montre que le modèle est très bon pour détecter les objets avec une tolérance d'erreur de 50% sur la position.
- La $mAP@50-95$, plus stricte car elle combine plusieurs seuils d'exigence (de 0.5 à 0.95 en IoU), s'élève à environ 88%, ce qui est très élevé dans ce contexte. Cela montre que même pour des précisions plus précises, le modèle maintient de très bonnes performances.

Conclusion :

Le modèle est robuste et fiable, même quand on lui demande une très grande précision dans la localisation des objets.

RÉSUMÉ GÉNÉRAL

L'ensemble des courbes montrent une excellente stabilité, une forte précision, et une bonne généralisation. Cela prouve que le modèle YOLOv11 est bien entraîné et adapté à la tâche de détection de la qualité du sol.

5.4. Détection des fuites des tuyaux à l'aide du model yolo v11

L'objectif de cette deuxième application est de détecter automatiquement les anomalies visibles au niveau des tuyaux à partir d'images. Le système distingue trois situations : fuite d'eau, fissure sur le tuyau, et débordement d'eau. Cette classification visuelle permet d'identifier rapidement les signes de problème dans les tuyaux d'eau et de faciliter les interventions de réparation.

5.4.1. Spécificité du Dataset

Les données utilisées pour cette tâche proviennent de situations réelles partagées dans des jeux de données publics disponibles sur la plateforme Roboflow. Ces images illustrent différentes anomalies visibles autour des tuyaux. Chaque image a été annotée manuellement en fonction de la nature du problème observé, en distinguant trois classes : fuite d'eau, fissure sur le tuyau, et débordement d'eau. Les annotations ont été

réalisées au format YOLO, à l'aide de boîtes englobantes associées à une étiquette de classe. Les images présentent une diversité en termes de luminosité, d'angles de prise, ce qui permet d'améliorer la robustesse du modèle lors de l'entraînement.

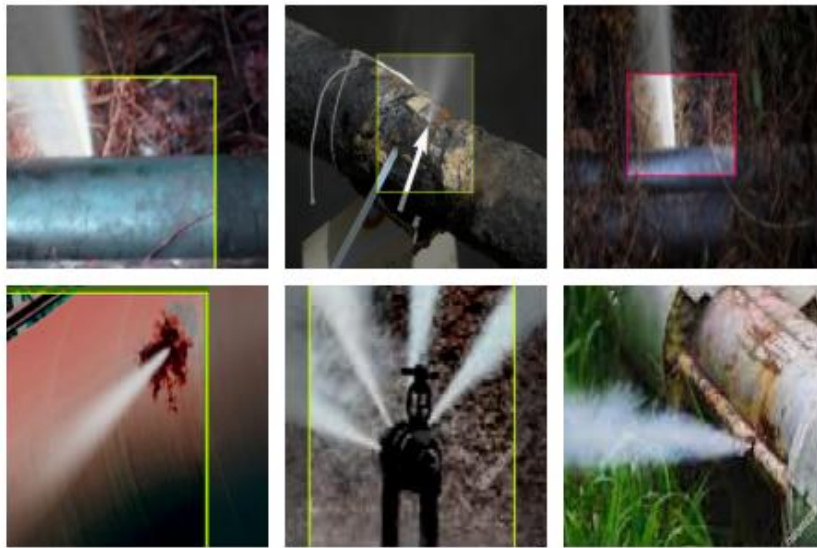


Figure 41 : Quelques images de la base des données du des fuites des tuyaux

5.4.2. Résultats et discussions

Après entraînement du modèle YOLOv11 et intervention au niveau des hyperparamètres, les résultats ont montré une très bonne capacité de détection pour les trois classes. Les courbes de loss ont convergé de manière stable, et les métriques finales sont les suivantes:

Précision globale	98.5%
Rappel global	84.2%
mAP@50	89.9%
mAP@50-95	80.2%

Tableau 3 : Résultats d'entrainement du model

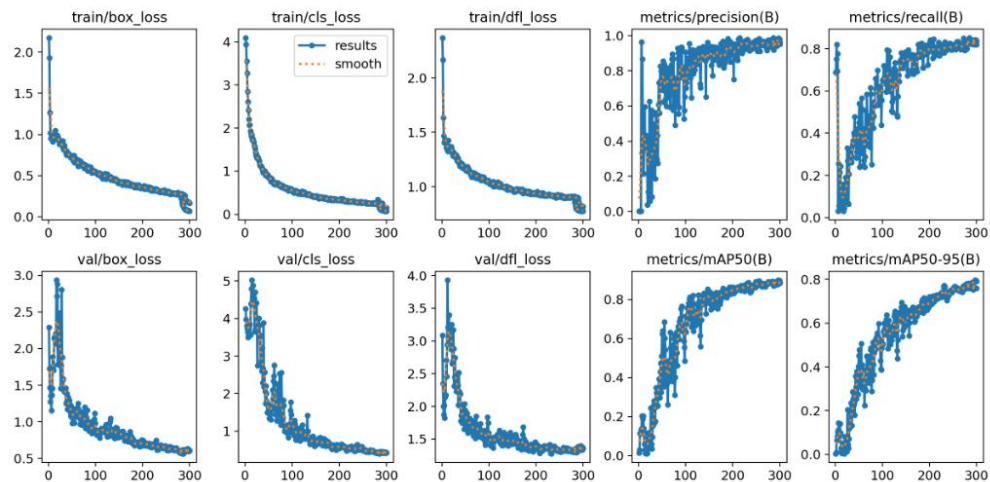


Figure 42 : Courbes d'apprentissage du model

5.4.3. Interprétation des résultats

Courbe des pertes :

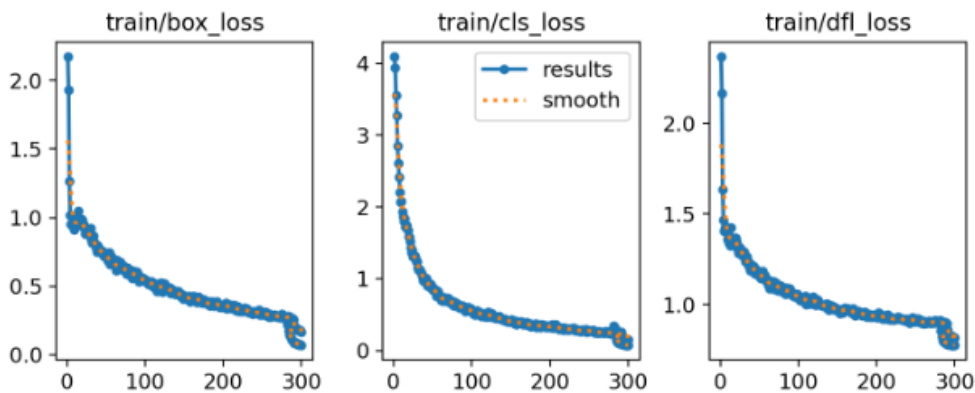


Figure 43 : Les courbes des pertes

Ces courbes montrent comment l'erreur du modèle diminue au fil des itérations, à la fois sur les données d'entraînement et de validation. On observe que toutes les courbes d'entraînement (train/..._loss) suivent une tendance décroissante continue, signe que le modèle apprend efficacement au fil des itérations.

La `box_loss`, La perte de localisation des objets diminue régulièrement, ce qui prouve que le modèle apprend à repérer correctement les anomalies dans les images (fissures, fuites, etc.).

La `cls_loss`, Forte baisse initiale de la perte de classification, indiquant que le modèle apprend vite à distinguer les trois classes

La `dfl_loss`, La perte de précision des contours (DFL) diminue de manière progressive, ce qui montre une bonne amélioration de la qualité des boîtes prédictives.

Conclusion :

Le fait que les courbes d'entraînement et de validation évoluent de façon parallèle est un bon indicateur : le modèle apprend bien sans surapprendre (pas d'overfitting).

Courbes de Précision et de Rappel

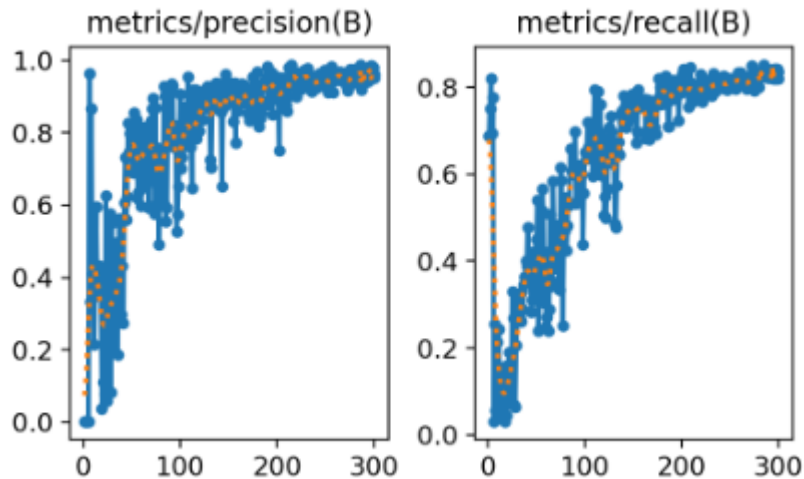


Figure 44 : Les courbes de Précision et de Rappel

Ces courbes permettent de mesurer la qualité des prédictions du modèle.

Précision (precision) : Monte vers 0.98, ce qui signifie que la majorité des objets détectés sont corrects, avec peu de fausses alertes.

Rappel (recall) : Atteint progressivement 0.85, ce qui montre que la plupart des anomalies présentes sont bien détectées.

Légères fluctuations visibles mais globalement les deux courbes convergent vers de bons niveaux de performance, équilibrés et stables.

Courbes mAP

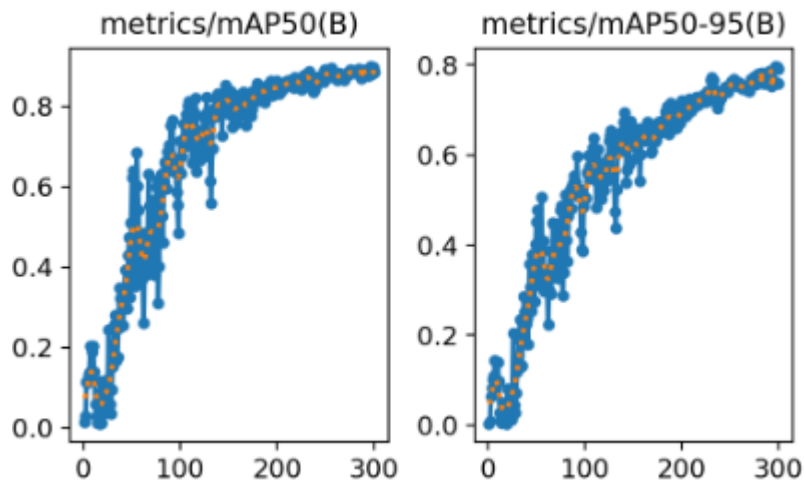


Figure 45 : Les courbes mAP

Les courbes mAP indiquent la performance globale du modèle à différents niveaux d'exigence.

mAP@50 atteint 0.89 : très bon score, preuve que le modèle localise correctement les objets dans la majorité des cas.

mAP@50-95 (plus précis) monte jusqu'à 0.80 : le modèle reste précis même sous des critères rigoureux, ce qui confirme sa robustesse.

Ces deux courbes montrent une progression constante, avec une bonne stabilisation en fin d'entraînement.

Résumé général

L'ensemble des courbes confirme que le modèle est : Bien entraîné (les pertes baissent clairement), Stable (courbes val/train bien alignées) et Efficace (précision, rappel et mAP très satisfaisants)

Ces résultats suggèrent que notre modèle est capable de détecter efficacement les fuites, fissures et le débordement d'eau autour des tuyaux, avec une très bonne fiabilité.

5.5. Détection de maladie des plantes l'aide de la nouvelle version YOLOv12

YOLOv12 représente une avancée majeure dans le domaine de la détection d'objets en temps réel. Cette version Sortie en mars 2025, introduit des améliorations architecturales significatives, visant à accroître la vitesse, la précision et l'efficacité, tout en étant adaptée aux appareils à ressources limitées tels que les mobiles et les systèmes embarqués. Son développement répond à la complexité croissante des tâches de vision par ordinateur et à la nécessité de rendre l'intelligence artificielle plus accessible sur le terrain. Bien que YOLOv11 ait déjà offert des progrès notables en termes de précision et de flexibilité, il restait difficile de maintenir un débit élevé sous contrainte matérielle. YOLOv12 surmonte ces limites grâce à des optimisations ciblées qui améliorent considérablement les performances globales du modèle. Il s'impose ainsi comme une solution puissante et efficace, particulièrement adaptée à des applications concrètes telles que la détection automatisée des maladies de plantes dans des contextes agricoles intelligents [44].

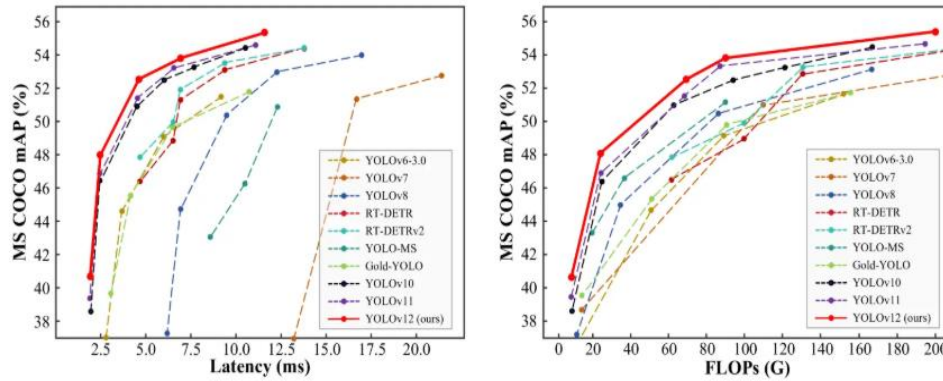


Figure 46 : Comparaison de YOLOv12 par rapport aux versions précédentes [45].

L'objectif de cette application est de détecter visuellement l'état de santé et maladies des plantes à partir d'images de base de données.

5.5.1. Spécificité du Dataset

Les données utilisées dans cette tâche proviennent de prises réelles sur le terrain, puis ont été organisées à l'aide de la plateforme Roboflow. à partir de jeux de données partagés sur cette dernière. Chaque image a été annotée en indiquant visuellement si la plante est saine ou malade, grâce à des boîtes englobantes associées à une classe, selon le format YOLO. Les photos ont été prises dans des conditions variées de lumière, d'angle de vue et d'aspect du sol, ce qui permet d'avoir un ensemble de données diversifié et adapté à l'entraînement du modèle.

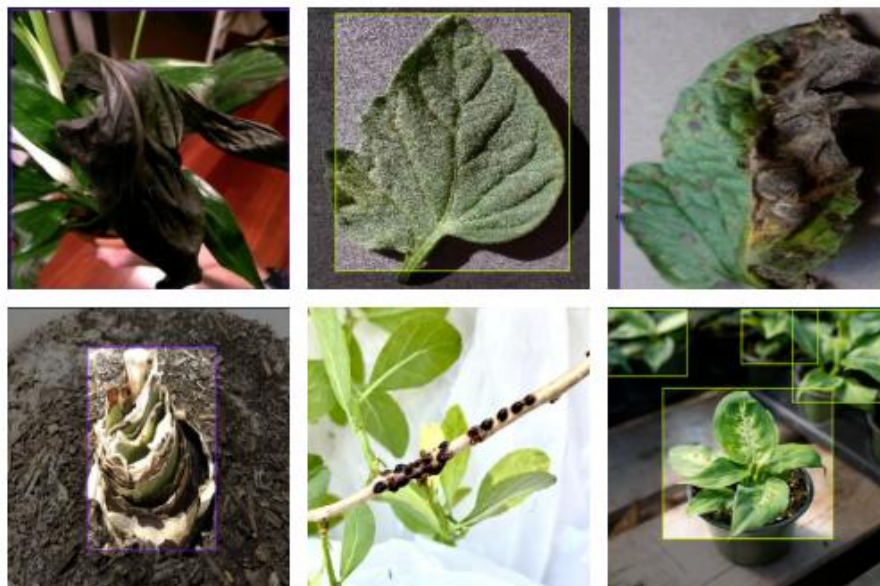


Figure 47 : Quelques images de la base des données de l'état des plantes

5.5.2. Résultats et discussions

Après entraînement du modèle YOLOv12 et intervention au niveau des hyperparamètres, les résultats ont montré une très bonne capacité de détection. Les courbes de loss ont convergé de manière stable, et les métriques finales sont les suivantes :

Précision globale	90.6%
Rappel global	88%
mAP@50	93.4%
mAP@50-95	78.2%

Tableau 4 : Métriques d'évaluation YOLOv12.

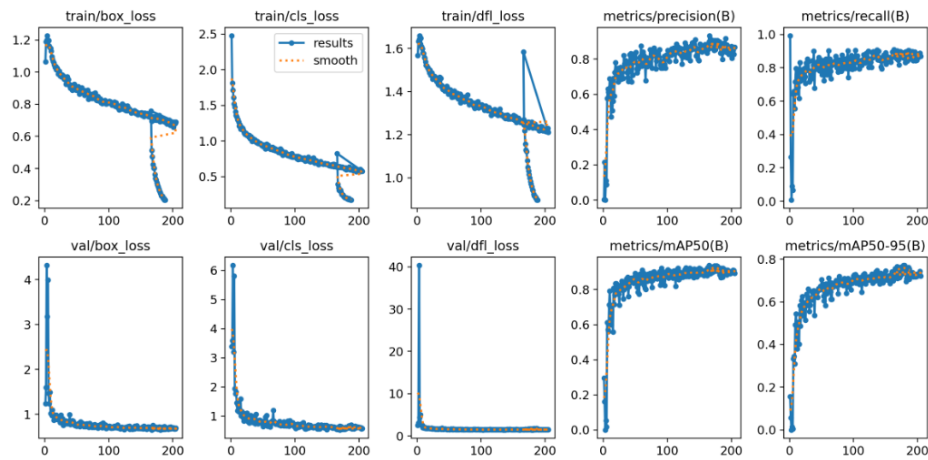


Figure 48 : Courbes d'apprentissage du model

5.5.3. Interprétation des résultats

Courbes de Précision et de Rappel

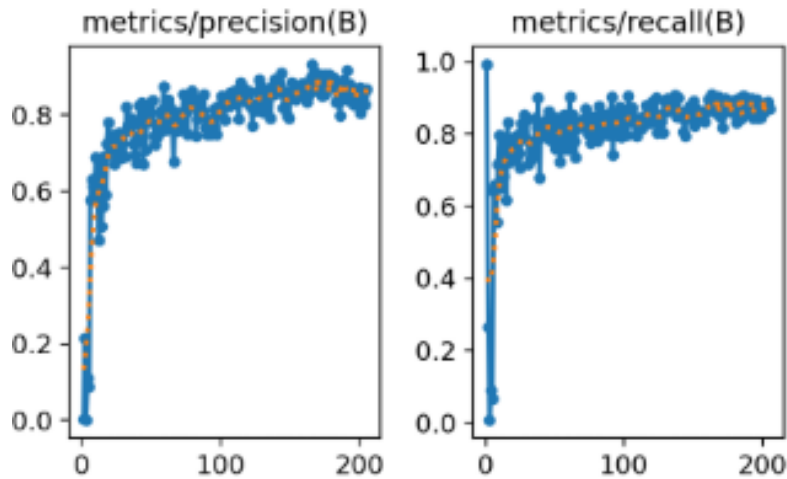


Figure 49 : Les courbes de Précision et de Rappel

Précision : La courbe progresse rapidement et atteint 0.90, ce qui signifie que la grande majorité des objets détectés sont corrects, avec très peu de fausses détections.

Rappel : Atteint progressivement 0.88. Cela indique que le modèle parvient à détecter presque tous les objets pertinents dans les images.

Courbes mAP

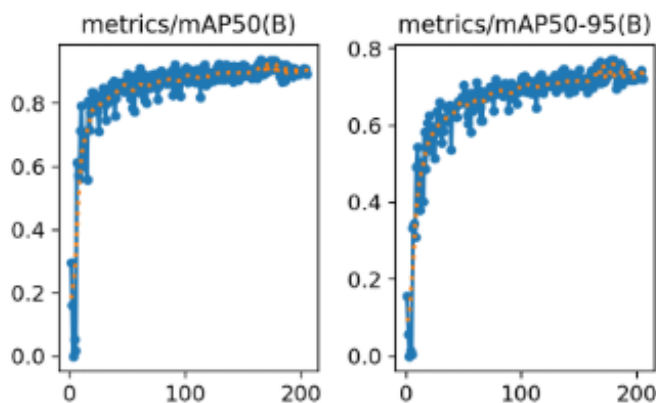


Figure 50 : Les courbes mAP

mAP@50 : Atteint 0.93, ce qui signifie que le modèle localise et classe très bien les objets dans la majorité des cas, même avec une tolérance modérée.

mAP@50-95 : Monte jusqu'à 0.78, ce qui est très bon pour un modèle de détection. Cela montre que le modèle reste précis même sous des critères compliqués.

Ces deux courbes montrent une progression constante et une bonne stabilisation en fin d'entraînement, confirmant la robustesse du modèle.

Courbe des pertes :

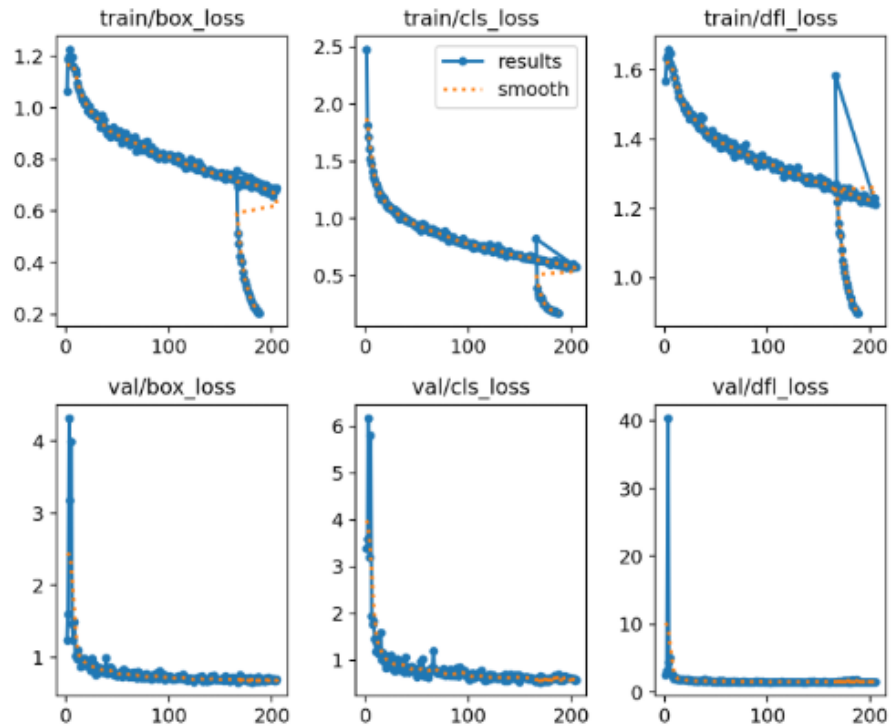


Figure 51 : Les courbes des pertes

Ces courbes montrent une diminution régulière de l'erreur du modèle tout au long de l'apprentissage, aussi bien sur les données d'entraînement que sur les données de validation.

train/box_loss & val/box_loss : La perte de localisation diminue progressivement, passant de ~1.2 à environ 0.25 pour l'entraînement, et de ~4 à ~0.8 pour la validation. Cela indique que le modèle apprend à repérer efficacement les objets ciblés dans les images.

train/cls_loss & val/cls_loss : Forte chute initiale, ce qui signifie que le modèle apprend rapidement à distinguer les différentes classes. La stabilisation à bas niveau montre une classification maîtrisée.

train/df1_loss & val/df1_loss : La perte relative à la précision des contours (Distance Focal Loss) suit une courbe typique : pic initial très élevé (surtout en validation), suivi d'une chute rapide et d'une stabilisation autour de ~ 1.0 . Cela confirme que le modèle s'améliore dans la définition des boîtes englobantes.

- **Explication des variations et des pics à la fin l'entraînement du model**

Les courbes de perte et les métriques issues de l'entraînement du modèle YOLOv12 révèlent une convergence progressive et stable, avec des performances finales élevées : une précision d'environ 90 %, un rappel supérieur à 87 %, un mAP@50 proche de 89 % et un mAP@50–95 atteignant 78 %.

Bien que quelques variation locales et pics ponctuels soient visibles, notamment à la fin d'entraînement, ceux-ci sont causées par la qualité variable des données, des lots d'images (batches) plus complexes, ou encore par certaines techniques moderne utilisées dans l'entraînement du nouveau model yolov12

Ces irrégularités n'ont eu aucun impact négatif sur les performances globales. Aucun signe critique de surapprentissage ou d'instabilité (comme des pertes qui augmentent, une divergence entre entraînement et validation, ou une stagnation des métriques), aucun signe de cela n'a été observé. Le modèle est donc jugé correct, fiable et précis.

5.6. Mise en œuvre des modèles entraînés

Après avoir entraîné nos modèles YOLOv11 et YOLOv12 avec de bons résultats, l'étape suivante a été de les rendre utilisables dans des situations réelles. L'idée était d'exploiter les poids obtenus pendant l'entraînement pour permettre une détection automatique des anomalies en temps réel, de façon simple et rapide.

Plutôt que d'intégrer ces modèles directement dans notre ESP32 ou un Raspberry Pi – ce qui demande souvent une optimisation à faire avec une configuration technique, nous avons choisi de les intégrer avec la plateforme Roboflow, une solution cloud

dédiée à la vision par ordinateur. Cette solution nous a permis de bénéficier d'un hébergement intelligent avec une prise en charge complète de l'inférence à distance

Deux solutions pratiques et puissantes ont été mis en place :

- Une interface en ligne accessible depuis n'importe quel appareil qui permet d'utiliser les poids du modèle et d'ouvrir directement la caméra de l'utilisateur et de lancer la détection en temps réel.
- Une API hébergée sur une plateforme web, dans laquelle on intègre directement les poids du modèle. Elle permet de téléverser une image ou une vidéo, puis d'obtenir instantanément les résultats de détection. Cette API peut aussi être utilisée pour automatiser le processus à partir d'un site web ou d'un autre système.

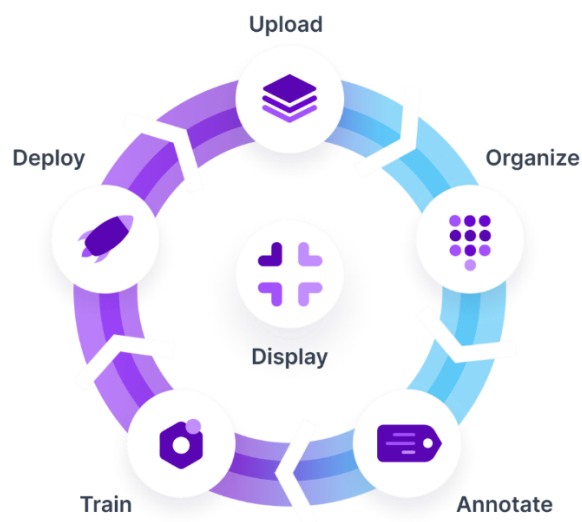


Figure 52 : Démarche d'hébergement avec Roboflow

5.6.1. Déploiement du model

Lors du déploiement, Roboflow affiche une interface pour demander quel type de modèle il souhaite utiliser.

Tout d'abord il faut préciser si le modèle doit seulement détecter des objets, les compter, ou encore les classer. Cette étape permet aussi d'indiquer la fonction principale du modèle et la qualité de ses prédictions attendue.

Dans notre cas, l'option "Détecter et classer" a été choisie, car elle correspond à notre objectif.



Figure 53 : Sélection du type de modèle à déployer sur Roboflow.

Après le choix des modèles et la configuration de certains paramètres comme le seuil de confiance et le mode d'affichage, les modèles sont automatiquement chargés et deviennent prêts à être utilisés pour la détection en ligne.

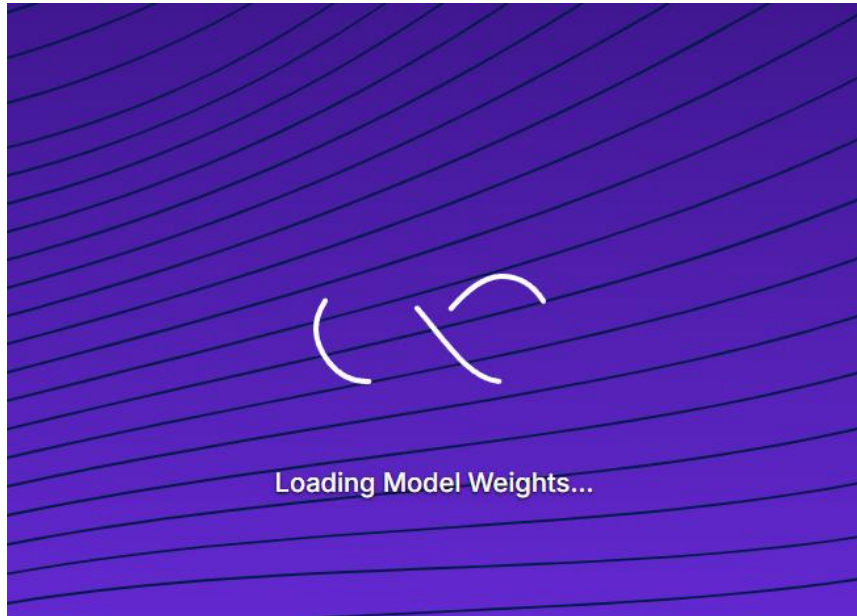


Figure 54 : Chargement des modèles

Une fois le modèle chargé, on aura plusieurs options pratiques, dont la génération automatique d'une interface de test. Cette interface permet à l'utilisateur soit d'activer la caméra pour effectuer une détection en temps réel, soit de générer une API personnalisée pour intégrer le modèle dans une autre application ou plateforme.

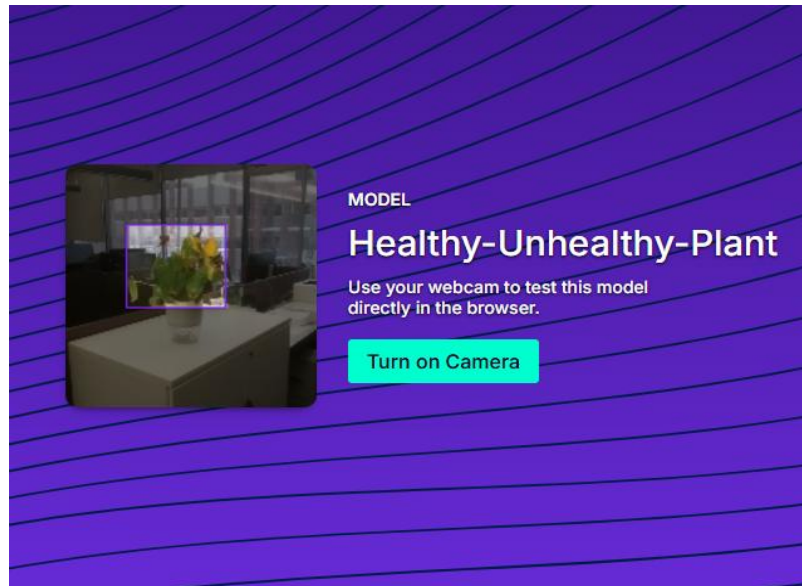


Figure 55 : Interface de test générée pour la détection en temps réel via la caméra.

Upload Method: ☒ Upload ☐ URL

Select File:

Filter Classes: Min Confidence: % Max Overlap: %

Separate names with commas

Inference Result: ☒ Image ☐ JSON

Labels: ☐ Off ☒ On

Stroke Width: ☐ 1px ☒ 2px ☐ 5px ☐ 10px

Figure 56 : L'API générée pour la détection à partir des captures ou des vidéos fournies.

5.6.2. Affichage des résultats

Après l'intégration des modèles et leur déploiement dans l'interface en ligne, plusieurs tests ont été effectués afin d'évaluer la qualité des détections en temps réel. Ces tests ont été réalisés à partir d'images variées et en activant la caméra depuis différents appareils. Voici un résumé des résultats observés :

- ✓ Le modèle détecte correctement les objets cibles dans la majorité des cas testé.
- ✓ Les boîtes englobantes sont généralement bien positionnées autour des zones ciblées
- ✓ La classification des objets détectés est fiable.
- ✓ Le score de confiance est souvent supérieur à 90 %.
- ✓ La détection est rapide, avec un délai de réponse de 1 à 2 secondes après la capture.
- ✓ L'interface fonctionne sans ralentissement, même sur des appareils mobiles.
- ✓ Le modèle reste stable même en cas de variation de luminosité ou d'angle de prise de vue.
- ✓ Aucune mauvaise classification majeure n'a été observée sur les exemples simples, ce qui confirme une bonne généralisation du modèle sur de nouvelles données.
- ✓ Lors des tests API, les résultats reçus sont clairs, bien structurés et identiques à ceux obtenus visuellement, ce qui confirme une bonne cohérence entre les deux méthodes de détection.

Les résultats ci-dessus sont illustrés par les exemples suivants, capturés directement depuis l'interface lors des tests en temps réel :



Figure 57 : Résultat de détection en direct en temps réel

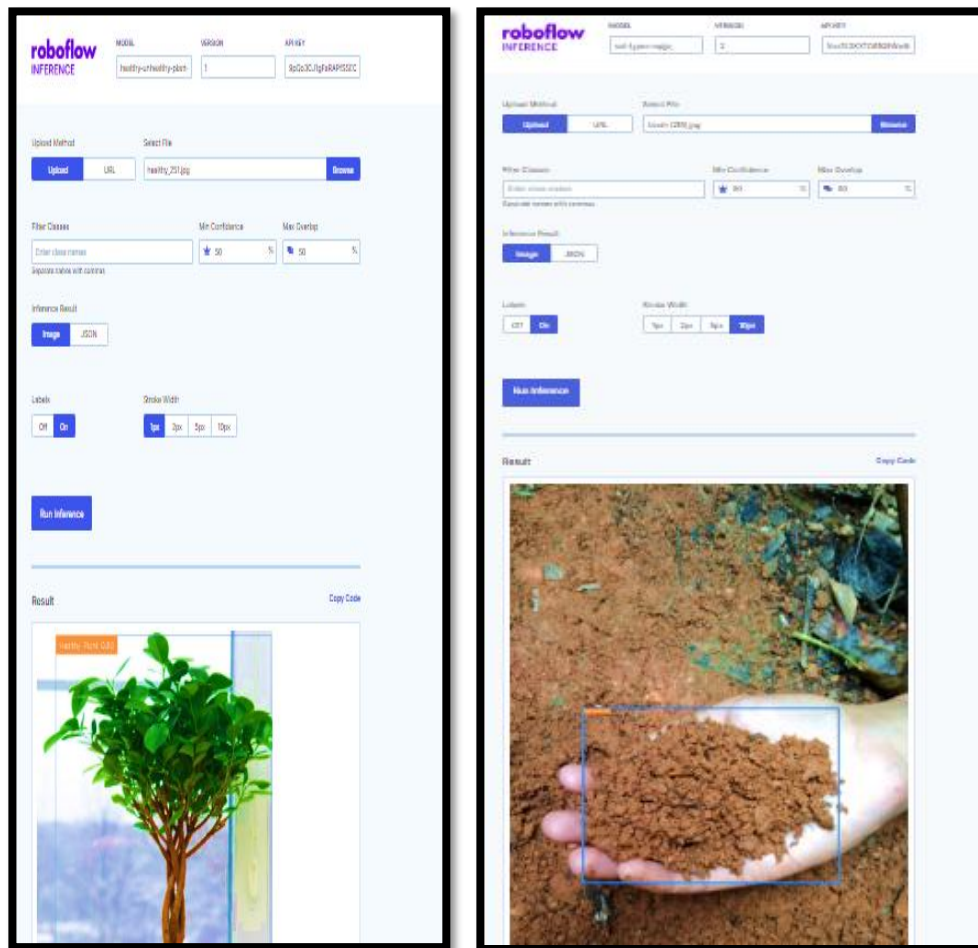


Figure 58 : Résultat obtenu instantanément par l'API.

En résumé, les résultats obtenus à travers l'interface en ligne et l'API Roboflow confirment que la détection d'anomalies peut être réalisée de manière efficace et rapide, Grâce à l'inférence déportée sur la plateforme Roboflow, le traitement est effectué en ligne, ce qui réduit les besoins matériels tout en assurant une flexibilité d'utilisation, même à distance.

Ce choix représente donc une solution légère et accessible, parfaitement adaptée à des tests rapides, des prototypes intelligents ou des déploiements dans des contextes à faibles ressources

5.7. Conclusion

Ce chapitre a mis en valeur le potentiel de l'intelligence artificielle et de l'apprentissage profond dans la détection automatique des anomalies visuelles, en s'appuyant sur les deux versions récentes du modèle YOLO : YOLOv11 et YOLOv12. Les tests effectués ont donné des résultats très satisfaisants, avec des boîtes de détection précises, des classifications fiables et des scores de confiance élevés. L'approche choisie se distingue par sa simplicité de mise en œuvre, car l'inférence se fait entièrement en ligne, sans recourir à des cartes électroniques ou à des dispositifs matériels complexes. Ainsi, l'utilisation de modèles récents couplés à des outils comme Roboflow permet de développer des solutions puissantes, accessibles et facilement déployables.

Comparaison des résultats avec des travaux récente

Le tableau suivant présente une comparaison entre le système proposé dans ce mémoire et plusieurs approches récentes dans le domaine des systèmes embarqué et les systèmes intelligent. Contrairement à la majorité des travaux étudiés qui se limitent à une seule couche (soit IoT, soit IA, soit interfaçage), ce mémoire propose une intégration complète et fonctionnelle de toutes les couches essentielles, avec une supervision hybride, un déploiement en temps réel, et des résultats de précision élevés.

Références	Application	Technologies utilisées	Précision IA	Déploiement	Originalité / Limites
Kaur et al. [5]	Irrigation intelligente	IoT + Machine Learning	93.5 %	Cloud	Bonne précision, pas d'IA visuelle ni supervision locale
Subramani et al. [22]	Classification du sol	CNN	97 %	Non embarqué	Précis mais juste une simulation
Gunawan et al. [15]	Suivi des plantes	IoT + Application mobile	-	Application mobile	Aucune intégration IA
Kumar et al. [23]	Classification du sol	Bag of Features	97 %	Non déployé	Pas de vision par ordinateur
Hazarika et al. [13]	Irrigation par module GSM	MODULE GSM	-	GSM	Très simple, sans retour visuel ni IA
Shen, Y et al [47]	Maladies sur feuilles de tomates	YOLOv8 modifié	85.7 %	Non embarqué	Bonne précision
Ce Mémoire	Système intelligent dédié à l'agriculture	Deep learning YOLOv11/v12 + IoT + Embedded + API + Web + LabVIEW	Jusqu'à 98.5 %	Triple couche (local + web + cloud)	Système complet, IA temps réel, supervision hybride

Tableau 5 : comparaison des résultats avec des travaux récentes

Conclusion

L'évolution des technologies numériques a ouvert la voie à une nouvelle génération de solutions agricoles, où les systèmes embarqués, l'Internet des Objets (IoT) et l'intelligence artificielle (IA) s'unissent pour répondre aux besoins croissants de performance, de durabilité et d'automatisation.

Ce mémoire s'inscrit pleinement dans cette approche, en présentant un système intelligent intégré, conçu pour l'agriculture. Cette solution globale englobe l'acquisition de données environnementales, la communication en temps réel, la commande automatique d'actionneurs et l'analyse avancée par IA, le tout dans une architecture unifiée, accessible et adaptable aux réalités du terrain agricole.

Ce travail a porté sur le développement et la mise en œuvre d'une solution embarquée basée sur le microcontrôleur ESP32, associé à une variété de capteurs et actionneurs. La programmation réalisée en C++ et python a permis d'assurer une gestion en temps réel des données environnementales, ainsi que des actions automatisées selon des conditions précises. La structure hybride de communication, articulée autour d'échanges locaux (via USB et LabVIEW) et distants (via Wi-Fi et page Web embarquée), a offert une flexibilité remarquable, rendant le système autonome et interconnecté.

Une attention particulière a été portée à l'expérience utilisateur à travers le développement d'une interface Web, permettant de suivre les valeurs mesurées et de contrôler les actionneurs en temps réel. Parallèlement, une interface de supervision graphique via LabVIEW a renforcé la lisibilité et la gestion technique du système en contexte local.

Dans une seconde phase, l'intelligence artificielle a été intégrée au projet pour automatiser la détection d'anomalies visuelles à partir d'images issues de l'environnement agricole. Deux modèles récents, YOLOv11 et YOLOv12, ont été

entraînés et évalués pour identifier différentes anomalies. Le déploiement s'est appuyé sur la plateforme Roboflow, permettant une inférence en ligne à partir d'une interface visuelle ou d'une API accessible via navigateur, sans passer par un déploiement matériel local.

Les résultats obtenus tout au long du projet ont confirmé la cohérence, la fiabilité et la réactivité du système embarqué, ainsi que la pertinence des solutions IA intégrées. L'ensemble du système démontre qu'une architecture embarquée légère peut être enrichie par des outils puissants, connectés et modernes, tout en gardant une mise en œuvre simple et accessible.

En somme, cette étude représente un exemple concret de convergence entre systèmes embarqués, interconnexion intelligente et intelligence artificielle dans un contexte agricole, où chaque décision est soutenue par l'automatisation et l'intelligence intégrée.

Publication :

[1] O. Khaldi and F. Slaoui-Hasnaoui, "Real-Time Visual Detection of Water Leaks in Irrigation Networks Using Deep Learning: A Smart Solution for Precision Agriculture," in Proceedings of the 2025 International Conference on Artificial Intelligence and Technological Solutions (ICAITech), Palembang, Indonesia, 2025, Paper ID: 1571195057.

[2] O. Khaldi and F. Slaoui-Hasnaoui, "Embedded System Controlled and Supervised Using LabVIEW," in Proceedings of the 4th International Conference on Embedded Systems and Artificial Intelligence (ESAI), 2025, Paper ID: 97.

[3] O. Khaldi and F. Slaoui-Hasnaoui, "Soil quality classification using deep learning," in Proceedings of the 4th International Conference on Embedded Systems and Artificial Intelligence (ESAI), , 2025, Paper ID: 53.

[4] O. Khaldi and F. Slaoui-Hasnaoui, "Design of an intelligent system based on IoT and AI for precision agriculture," Journal of Electrical Systems and Information Technology, 2025, ID: JESI-D-25-00419

.

Références

- [1] «Borges, R.C.; Beuter, C.H.; Dourado, V.C.; Bento, M.E.C. Internet of Things Application in an Automated Irrigation Prototype Powered by Photovoltaic Energy. *Energies* 2024, 17, 2219. <https://doi.org/10.3390/en17092219>».
- [2] « Pablo Catota, Cesar Minaya, Jenny Alexandra Guzmán Acurio, Efrén Damián Alban Andrade and Ricardo Rosero (2024), Sustainability of precision agriculture as a proposal for the development of autonomous crops using IoT . *IJEER* 12(1), 146-153. DOI: 10.3».
- [3] «Bouali Et-taibi, M., Abid, M.R., Boufounas, E.-M., Morchid, A., Bourhnane, S., Abu Hamed, T., & Benhaddou, D.,» *Enhancing water management in smart agriculture: A cloud and IoT-based smart irrigation system. Results in Engineering*, 2024, 22, 102283. <https://doi.org/10.1016/j.rineng.2024.102283>.
- [4] «Idris, F., Latiff, A.A., Buntat, M.A., Lecthmanan, Y., & Berahim, Z.IoT-based fertigation system for agriculture. *Bulletin of Electrical Engineering and Informatics*, 2024, 13(3), 1574–1581. <https://doi.org/10.11591/eei.v13i3.6829>».
- [5] *Kaur, A., Bhatt, D.P., & Raja, L., Developing a hybrid irrigation system for smart agriculture using IoT sensors and machine learning in Sri Ganganagar, Rajasthan. Journal of Sensors*, 2024, Article ID 6676907. <https://doi.org/10.1155/2024/6676907>.
- [6] «G. Dinesh, A. K. Gupta, M. Nagaseshireddy, P. D. Prasanna, M. S. Varshini and K. Gowtham,,» "LoRa - Powered Smart Agriculture System for Monitoring and Controlling," 2024 *IEEE Wireless Antenna and Microwave Symposium (WAMS)*, Visakhapatnam, India, 2024, pp. 1-6, doi: 10.1109/WAMS59642.2024.10528095..
- [7] «K. M. Hazarika et al., "Smart Wireless Irrigation System-A Prototype," 2023 IEEE Silchar Subsection Conference (SILCON), Silchar, India, 2023, pp. 1-6, doi: 10.1109/SILCON59133.2023.10404825.».
- [8] *Yosua, L., & Rino.,Designing smart home prototype using IoT-based ESP32 microcontroller and Telegram. The Tech-E Journal*, 2024, 7(2). <https://doi.org/10.31253/te.v7i2.2651>.
- [9] *P. Kuchár et al., "Development of a Smart Non-Lethal Trap Prototype Utilising ESP32-CAM and PIR Sensor," 2024 ELEKTRO (ELEKTRO), Zakopane, Poland, 2024, pp. 1-6, doi: 10.1109/ELEKTRO60337.2024.10556848..*
- [10] «N. Ya'acob, S. I. Ismail, M. Z. M. Amin and N. F. A. Aziz, "Design and Implementation of a Wireless Health Monitoring System for Real-Time Patient Data Using ESP32," 2024

IEEE 15th Control and System Graduate Research Colloquium (ICSGRC), SHAH ALAM, Malay».

- [11] «N. J. P. Ramírez and A. M. R. Duke, "Smart Agricultural Monitoring and Automated Irrigation Through an IoT-Based Wireless Sensor Network Using ESP32: A MATLAB Analysis," 2024 9th International Conference on Control and Robotics Engineering (ICCRE), Osaka,».
- [12] «Patil, R.J., Mulage, I., & Patil, N., Smart agriculture using IoT and machine learning. Journal of Scientific Research & Technology (JSRT), 2023, 1(3), 47–59. www.jsrtjournal.com».
- [13] «G. Mohyuddin, M. A. Khan, A. Haseeb, S. Mahpara, M. Waseem and A. M. Saleh, "Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture System: A Comprehensive Review," in IEEE Access, vol. 12, pp. 60155-60184, 2024,,» doi: 10.1109/ACCESS.2024.3390581.
- [14] «Jaliyagoda N, Lokuge S, Gunathilake PMPC, Amaratunga KSP, Weerakkody WAP, et al.,» (2023) *Correction: Internet of Things (IoT) for Smart Agriculture: Assembling and assessment of a low-cost IoT system for polytunnels. PLOS ONE 18(12): e0296110, https://doi.org/10.1371/journal.pone.0296110.*
- [15] «S. Usharani, Gayathri, R. S, D. S. Kishore and S. Depuru, "IoT based Animal Trespass Identification and Prevention System for Smart Agriculture," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023».
- [16] «T. S. Gunawan, N. N. Kamarudin, M. Kartiwi and M. R. Effendi, "Automatic Watering System for Smart Agriculture using ESP32 Platform," 2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Melaka, Malaysia,».
- [17] «M. T. I. M. Carcellar, C. J. S. Tychuaco and A. N. Yumang, "Self-Applicable Eye Strain Detection Through the Measurement of Blink Rate Using Raspberry Pi,"» 2024 *IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAET)*, Kota Kinabalu, Malaysia, 2024, pp. 13-17, doi: 10.1109/IICAET62352.2024.10730733..
- [18] «Subramani, S., Suganthi, N. WHO-YOLO NET : prédiction et classification des sols basées sur YOLOV3 avec optimisation des baleines. SIViP 18 , 3495–3507 (2024).».
- [19] «Kumar, S., Sharma, B., Sharma, VK, Poonia, RC : Prédiction automatisée des sols à l'aide d'un algorithme d'optimisation chaotique par sac de caractéristiques. Évol. Intell. 14 , 293–304 (2021).».

- [20] «Agarwal, R., Shekhawat, NS, Luhach, AK : Classification automatisée d'images de sol par optimisation chaotique de la solubilité des gaz de Henry : système agricole intelligent. Microprocessus. Microsyst. (2021).».
- [21] «Latha, K. R., & Kumaresan, R. (2025). Prediction of Soil Nutrients and Crop Recommendation using Machine Learning Techniques. Nepal Journal of Environmental and Agricultural Research, 7(1), 45–52.».
- [22] «Saki, M., Keshavarz, R., Abolhasan, M., Lipman, J., & Shariati, N. (2024). Precision Soil Quality Analysis Using Transformer-Based Data Fusion Strategies: A Systematic Review. arXiv preprint arXiv:2410.18353.».
- [23] «Kavitha, R., & Rajesh, M. (2023). Soil Type Classification and Health Monitoring using ResNet Model. Agriculture, 15(6), 661.».
- [24] «Mahore, V., Soni, P., Patidar, P., Nagar, H., Chouriya, A., & Machavaram, R. Development and implementation of a Raspberry Pi-based IoT system for real-time,» *performance monitoring of an instrumented tractor. Smart Agricultural Technology, 2024, 9, 100530. <https://doi.org/10.1016/j.atech.2024.100530>.*
- [25] «L. Mezai, A. Benhamouda, I. Khouns and R. Lakbichi, "Design and Implementation of a Disinfecting Mobile Robot Based on a Raspberry Pi 4 Board,"» *2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1-6, doi: 10.1109/ICEEAC61226.2024.10576543..*
- [26] «S. C H, C. S. A. Varma, A. B R, C. Kumar and A. Sirdeshpande, "Development Of An Uncrewed Vehicle Prototype Using Deep Learning And IOT,"» *2023 Fourth International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2023, pp. 1-7, doi: 10.1109/ICSTCEE60504.2023.10585199..*
- [27] «R. Tarun and B. P. Esther, "Real-time regional road sign detection and identification using Raspberry Pi,"» *2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICNWC57852.2023.10127370..*
- [28] «Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. Sensors (Basel), 18(8), 2674. doi:10.3390/18082674 PMID:30110960».
- [29] «Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big Data in Smart Farming – A review. Agricultural Systems, 153, 69–80.».

- [30] «Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805.».
- [31] «Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.».
- [32] «Marwedel, P. (2021). *Embedded System Design* (3rd ed.). Springer».
- [33] «Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Computing Surveys (CSUR)*, 41(3), 1–58.».
- [34] «Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99, 215–249.».
- [35] «Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE.».
- [36] «Warden, P., & Situnayake, D. (2019). *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media.».
- [37] «Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazekas, A., Holleman, J., ... & Warden, P. (2021). MicroNets: Neural Network Architectures for Deploying TinyML Applications on Commodity Microcontrollers.» *In Proceedings of Machine Learning and Systems (Vol. 3, pp. 517–532)*..
- [38] «Merenda, Massimo & Porcaro, Carlo & Iero, Demetrio. (2020). Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors*. 20. 2533. 10.3390/s20092533.».
- [39] «Architecture du esp32 <https://forum.arduino.cc/t/learning-architecture-and-multi-tasking-programming-of-32-bit-dual-core-microcontroller-using-esp32-dev-module-arduino-ide-esp-idf-and-freertos/1154849>».
- [40] «Schéma de circuit du module DHT11<https://circuitdigest.com/microcontroller-projects/interfacing-dht11-sensor-with-arduino>».
- [41] «Schéma de circuit du module capteur capacitif d'humidité du sol <https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/>».
- [42] «<https://www.ni.com/fr/support/documentation/supplemental/08/labview-for-loops-and-while-loops-explained.html>».
- [43] «YOLOv11 : aperçu des principales améliorations architecturales,» <https://arxiv.org/html/2410.17725v1#bib>.

- [44] «<https://arxiv.org/html/2502.14740v1>, YOLOv12 : aperçu des principales améliorations architecturales».
- [45] «Yunjie Tian, Qixiang Ye et David Doermann. Yolov12 : Détecteurs d'objets en temps réel centrés sur l'attention, 2025.».
- [46] «Schéma de circuit du module INA219 <https://www.electronicclinic.com/ina219-current-sensor-with-arduino-circuit-and-code-explained/>».
- [47] «Shen, Y.; Yang, Z.; Khan, Z.; Liu, H.; Chen, W.; Duan, S. Optimization of Improved YOLOv8 for Precision Tomato Leaf Disease Detection in Sustainable Agriculture. *Sensors* 2025, 25, 1398. <https://doi.org/10.3390/s25051398>».