REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–

المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
ـ تلمسان ـ

Mémoire de fin d'étude

**Pour l'obtention du diplôme de Master**

Filière : Automatique
Spécialité : Automatique

**Présenté par : MERAKCHI Billel Ali**
**HAOUAIA Abderrahmane El Mokhtar**

Thème

# Suivi Intelligent de l'Hydratation, de la Croissance et du Stress des Plantes par IA dans un Système IoT d'Irrigation Automatisée

Soutenu publiquement, le 21 / 06 / 2026 devant le jury composé de :

| | | | |
|---|---|---|---|
| Dr. ABDI Sidi Mohamed | MCA | ESSA. Tlemcen | Président |
| Dr. KARAOUZANE Zoheir | MCB | ESSA. Tlemcen | Directeur de mémoire |
| Dr. CHELLAL Arezki | Docteur | CeDRI/IPB – Portugal | Co- Directeur de mémoire |
| Dr. OUDJAMAA Farid | MCB | ESSA. Tlemcen | Examinateur 1 |
| Dr. LASSOUANI Brahim | Docteur | R.D Algérie Télécom | Examinateur 2 |
| Dr. MEGNAFI Hicham | MCA | ESSA. Tlemcen | Invité 1 |
| Dr. AJIM Ramzeeddine | Ingénieur | ESSA. Tlemcen | Invité 2 |

Année universitaire : 2025 /2026


بسم الله الرحمن الرحيم

# Dédicaces

Au terme de ce travail, qui marque l'aboutissement de plusieurs années d'études, d'efforts et de persévérance, nous tenons à dédier ce mémoire à toutes les personnes qui nous ont soutenus, encouragés et accompagnés tout au long de notre parcours académique. À nos très chers parents, Vous avez été notre premier soutien, notre source de motivation et notre plus grande force. Par votre amour inconditionnel, vos sacrifices innombrables, votre patience et vos encouragements constants, vous nous avez permis de poursuivre nos études dans les meilleures conditions et d'atteindre nos objectifs.

Vous avez toujours cru en nous, même dans les moments les plus difficiles, et vous nous avez appris les valeurs qui guident aujourd'hui notre vie : le travail, la persévérance, le respect, l'intégrité et la détermination. Chaque étape franchie, chaque succès obtenu et chaque objectif atteint portent l'empreinte de vos efforts et de votre dévouement.

Aucune parole ne saurait exprimer pleinement notre gratitude et notre profonde reconnaissance. Que ce mémoire soit le témoignage de notre amour, de notre respect et de notre éternelle gratitude envers vous. À nos frères et sœurs, Pour leur affection, leur soutien moral et leur présence constante. Merci pour votre confiance, vos encouragements et les moments de joie partagés qui nous ont aidés à surmonter les difficultés rencontrées tout au long de ce parcours.

À nos familles respectives, Pour leur amour, leurs prières, leurs conseils et leur soutien indéfectible. Votre confiance en nos capacités a toujours été une source de motivation et d'inspiration. À nos enseignants et à tous ceux qui ont contribué à notre formation, Nous exprimons notre profonde reconnaissance pour les connaissances transmises, les conseils prodigués et l'encadrement dont nous avons bénéficié durant toutes ces années d'études. Votre engagement et votre dévouement ont joué un rôle essentiel dans notre développement académique et professionnel.

À nos amis et camarades, Pour leur soutien, leur compréhension, leur aide précieuse et les nombreux moments de partage qui ont rendu cette aventure universitaire plus enrichissante et plus agréable.

Enfin, à toutes les personnes qui, de près ou de loin, ont contribué à notre réussite et à l'aboutissement de ce projet, nous adressons nos sincères remerciements et notre profonde gratitude.

Nous, **HAOUAIA Abderrahmane** et **MERAKCHI Billel Ali**, dédions humblement ce travail à tous ceux qui ont cru en nous, nous ont soutenus et accompagnés tout au long de cette aventure. Nous espérons que ce mémoire sera à la hauteur des sacrifices consentis, des espoirs placés en nous et de la confiance qui nous a été accordée. Avec notre plus profond respect, notre reconnaissance sincère et notre immense gratitude.

# Remerciments

Nous tenons tout d'abord à remercier Allah le Tout-Puissant et Miséricordieux, qui nous a donné la force, la patience et la volonté nécessaires pour mener à bien ce modeste travail.

Nous exprimons notre profonde gratitude à nos encadrants, **Dr. MEGNAFI Hicham, Dr. KARAOUZEN Zoheir et Dr. CHELLAL Arezki**, enseignants à l'École Supérieure des Sciences Appliquées de Tlemcen (ESSA Tlemcen), pour leur accompagnement, leurs conseils précieux, leurs orientations pertinentes ainsi que leur disponibilité tout au long de la réalisation de ce projet. Leur expertise et leur soutien ont été essentiels à l'aboutissement de ce travail.

Nous adressons également nos sincères remerciements à **M. Ramzeeddine AJIM**, responsable du FabLab, pour son aide, sa disponibilité et son encadrement technique durant les différentes étapes de notre projet.

Nos vifs remerciements vont également aux membres du jury, **Dr.ABDI Sidi Mohamed , Dr.OUDJAMAA Farid et Dr.LASSOUANI Brahim** qui nous feront l'honneur d'évaluer ce travail, ainsi qu'à l'ensemble des enseignants du département **Automatique** de l'ESSA Tlemcen, pour la qualité de la formation et les connaissances qu'ils nous ont transmises.

Nous remercions aussi nos familles pour leur soutien moral, leurs encouragements et leurs sacrifices constants qui nous ont permis d'atteindre ce niveau. Enfin, nous remercions tous nos amis et camarades de la promotion Automatique, pour leur aide, leur esprit de solidarité et leur soutien tout au long de notre parcours universitaire.

# الملخص

في ظل التحديات المتزايدة المرتبطة بالإدارة المستدامة للموارد المائية وتحسين الإنتاجية الزراعية، تقدم هذه المذكرة تصميم وإنجاز نظام ذكي للري الآلي يعتمد على إنترنت الأشياء والذكاء الاصطناعي . ويهدف هذا النظام إلى ضمان متابعة ذكية لحالة ترطيب النباتات ونموها وصحتها، مع ترشيد استهلاك المياه.

يعتمد النظام المقترح على دمج مجموعة من التقنيات المتكاملة، حيث تتولى شبكة من حساسات إنترنت الأشياء جمع المعطيات البيئية في الزمن الحقيقي، اللازمة لمراقبة المحاصيل والتحكم الآلي في عملية الري. كما تم تطوير نموذج للتعلم العميق يعتمد على الشبكات العصبية الالتفافية للكشف عن أمراض النباتات وتصنيفها انطلاقًا من صور الأوراق النباتية. ولتحسين موثوقية التنبؤات، تم دمج آلية تحقق متعددة المعايير ضمن مسار معالجة الصور.

كما يضم النظام بنية تحكم ذكية تتيح الإدارة الآلية لمختلف العمليات المرتبطة بالري ومراقبة المحاصيل. وتوفر واجهة إشراف إمكانية عرض البيانات في الزمن الحقيقي، ومتابعة حالة النظام، والتفاعل مع المستخدم.

أظهرت النتائج المتحصل عليها فعالية المقاربة المقترحة، خاصة في مجال الكشف عن الأمراض النباتية وأتمتة العمليات الزراعية. ويسمح التكامل بين الذكاء الاصطناعي وإنترنت الأشياء بتحقيق مراقبة مستمرة للمحاصيل، والكشف المبكر عن المشكلات، وتحسين استغلال الموارد، مما يساهم في تطوير زراعة أكثر استدامة وإنتاجية وقدرة على التكيف مع التحديات البيئية الراهنة.

الكلمات المفتاحية : الزراعة الذكية، إنترنت الأشياء، الذكاء الاصطناعي، التعلم العميق، الشبكات العصبية الالتفافي، الري الآلي، كشف أمراض النباتات، مراقبة المحاصيل، الزراعة الدقيقة.

# Abstract

In the context of growing challenges related to sustainable water resource management and the improvement of agricultural productivity, this thesis presents the design and implementation of an intelligent automated irrigation system based on the Internet of Things and Artificial Intelligence . The main objective is to provide smart monitoring of plant hydration, growth, and health status while optimizing water consumption.

The proposed system integrates several complementary technologies. An IoT sensor network is employed to collect environmental data in real time, enabling continuous crop monitoring and automatic irrigation control. In addition, a Deep Learning model based on Convolutional Neural Networks has been developed to detect and classify plant diseases from leaf images. To enhance prediction reliability, a multi-criteria validation mechanism has been incorporated into the image-processing pipeline.

The system also includes an intelligent control architecture that enables the automated management of irrigation operations and crop supervision tasks. A monitoring interface provides real-time data visualization, system status tracking, and user interaction.

The obtained results demonstrate the effectiveness of the proposed approach, particularly in plant disease detection and the automation of agricultural operations. The integration of AI and IoT technologies enables continuous crop monitoring, early anomaly detection, and optimized resource utilization, thereby contributing to a more sustainable, productive, and environmentally adaptive agricultural system.

**Keywords** : Smart Agriculture, Internet of Things , Artificial Intelligence , Deep Learning, Convolutional Neural Networks , Automated Irrigation, Plant Disease Detection, Crop Monitoring, Precision Agriculture.

# Résumé

Dans un contexte marqué par les défis liés à la gestion durable des ressources hydriques et à l'amélioration de la productivité agricole, ce mémoire présente la conception et la réalisation d'un système intelligent d'irrigation automatisée basé sur l'Internet des Objets et l'Intelligence Artificielle . L'objectif principal est d'assurer un suivi intelligent de l'hydratation, de la croissance et de l'état sanitaire des plantes tout en optimisant l'utilisation de l'eau.

Le système proposé repose sur l'intégration de plusieurs technologies complémentaires. Un réseau de capteurs IoT assure la collecte en temps réel des paramètres environnementaux nécessaires à la surveillance des cultures et au pilotage automatique de l'irrigation. Par ailleurs, un modèle de Deep Learning basé sur les réseaux de neurones convolutifs a été développé afin de détecter et classifier les maladies des plantes à partir d'images de feuilles. Afin d'améliorer la fiabilité des prédictions, un mécanisme de validation multicritère a été intégré au processus de traitement des images.

Le système intègre également une architecture de contrôle intelligente permettant la gestion automatisée des différentes opérations liées à l'irrigation et à la supervision des cultures. Une interface de supervision assure la visualisation en temps réel des données, le suivi de l'état du système ainsi que l'interaction avec l'utilisateur.

Les résultats obtenus démontrent l'efficacité de l'approche proposée, notamment en matière de détection des maladies végétales et d'automatisation des opérations agricoles. L'intégration conjointe de l'IA et de l'IoT permet ainsi une surveillance continue des cultures, une détection précoce des anomalies et une gestion optimisée des ressources, contribuant à une agriculture plus durable, plus productive et mieux adaptée aux contraintes environnementales actuelles.

**Mots-clés** : Agriculture intelligente, Internet des Objets , Intelligence Artificielle, Deep Learning, Réseaux de neurones convolutifs , Irrigation automatisée, Détection des maladies des plantes, Surveillance des cultures, Agriculture de précision.

# Table des matières

**Liste des figures** **10**

**Liste des Tableaux** **11**

**Liste des acronymes** **11**

**Introduction** **12**

**1 Fondements Théoriques et État de l'Art** **15**
1.1 Introduction . . . . . 15
1.2 L'intelligence artificielle . . . . . 15
1.3 Historique de l'intelligence artificielle . . . . . 16
1.4 Méthodologie générale de l'IA . . . . . 16
1.5 L'apprentissage automatique (Machine Learning) . . . . . 18
1.5.1 Types de machine Learning . . . . . 18
1.5.2 L'apprentissage supervisé . . . . . 19
1.5.3 Apprentissage non supervisé . . . . . 19
1.5.4 L'apprentissage semi-supervisé . . . . . 19
1.5.5 Apprentissage par renforcement . . . . . 20
1.6 Définition et Principe de l'Apprentissage Profond (Deep Learning) . 20
1.7 Les Réseaux de Neurones . . . . . 21
1.7.1 Typologie des réseaux neuronaux . . . . . 21
1.7.2 Neurones récurrents(RNN) . . . . . 22
1.7.3 Les réseaux de neurones artificiels (ANN) . . . . . 23
1.7.4 Les réseaux de neurones convolutifs CNN . . . . . 24
1.8 Justification du choix de la méthode CNN . . . . . 24
1.9 Les réseaux de neurones convolutifs . . . . . 25
1.10 Les couches de réseaux de neurones convolutifs . . . . . 25
1.10.1 La couche de convolution . . . . . 25
1.10.2 La couche de Pooling (POOL) . . . . . 27
1.10.3 La couche Fully Connected (FC) . . . . . 28
1.11 Analyse des systèmes d'irrigation intelligente préexistants . . . . . 28
1.12 Agriculture Intelligente (Smart Agriculture) . . . . . 30
1.12.1 Définition de l'Agriculture Intelligente . . . . . 30

1.12.2 Domaines d'Application de la Smart Agriculture . . . . . 30
1.12.3 Avantages de l'Agriculture Intelligente . . . . . 31
1.13 Internet des objets . . . . . 32
1.13.1 Definition . . . . . 32
1.13.2 Architecture de l'internet d'objet . . . . . 32
1.13.3 Les différents processus de l'internet des objets . . . . . 32
1.14 Les technologies du réseau . . . . . 33
1.14.1 Courte portée . . . . . 33
1.14.2 Moyenne portée . . . . . 34
1.14.3 Longue portée . . . . . 34
1.14.4 Protocoles de Communication Utilisés dans IoT . . . . . 35
1.15 L'agriculture de précision et la convergence IA-IoT . . . . . 36
1.16 Conclusion . . . . . 37

**2 Conception et Méthodologie du Système IoT d'Irrigation Intelligente Basé sur l'IA 38**
2.1 Introduction . . . . . 38
2.2 Résumé global du fonctionnement . . . . . 38
2.3 Description détaillée du système . . . . . 39
2.3.1 Bibliothèques utilisées . . . . . 39
2.3.2 Seaborn . . . . . 40
2.3.3 OpenCV . . . . . 40
2.3.4 Scikit-Learn . . . . . 40
2.4 Fonctionnement du l'entraînement . . . . . 41
2.4.1 Étape 1 : Chargement des données . . . . . 41
2.4.2 Étape 2 : Prétraitement des images . . . . . 41
2.4.3 Étape 3 : Création du CNN . . . . . 41
2.4.4 Étape 4 : Compilation . . . . . 41
2.4.5 Étape 5 : Entraînement . . . . . 41
2.4.6 Étape 6 : Évaluation . . . . . 42
2.4.7 Étape 7 : Sauvegarde . . . . . 42
2.5 Fonctionnement du test . . . . . 43
2.5.1 Étape 1 : Chargement du modèle . . . . . 43
2.5.2 Étape 2 : Chargement d'une image . . . . . 43
2.5.3 Étape 3 : Prétraitement . . . . . 43
2.5.4 Étape 4 : Prédiction . . . . . 44
2.5.5 Étape 5 : Sélection de la meilleure classe . . . . . 44
2.5.6 Étape 6 : Affichage du résultat . . . . . 44
2.6 Architecture Globale du Système . . . . . 44
2.7 Description des Nœuds . . . . . 45
2.8 Flux de Données Complet . . . . . 46
2.9 Dataset PlantVillage . . . . . 47
2.9.1 Description du Dataset . . . . . 47
2.9.2 Base de Données des Maladies (`DISEASE_DB`) . . . . . 47
2.10 Prétraitement des Images . . . . . 48
2.10.1 Étapes du Prétraitement . . . . . 48
2.10.2 Fonction de Prétraitement . . . . . 49
2.11 Mécanisme de filtrage triple des prédictions . . . . . 49

2.11.1 Synergie des trois niveaux de filtrage . . . . . 51
2.12 Intégration dans le pipeline IoT . . . . . 51
2.13 Machine à États Finis : Dosage séquentiel du NPK . . . . . 53
2.13.1 Description des états . . . . . 53
2.13.2 Conditions de déclenchement et sécurité . . . . . 54
2.14 Interface utilisateur et affichage . . . . . 54
2.14.1 Application Streamlit . . . . . 54
2.14.2 Interface d'affichage Arduino (TFT 3.5") . . . . . 57
2.15 Prototype final du système intelligent proposé . . . . . 58
2.16 Conclusion : . . . . . 59

**3 Implémentation, Tests et Évaluation des Performances du Système 60**
3.1 Introduction . . . . . 60
3.2 Environnement d'Implémentation et de Test . . . . . 60
3.2.1 Configuration matérielle . . . . . 60
3.2.2 Configuration logicielle . . . . . 61
3.3 Implémentation et Validation du Mécanisme de Filtrage Triple . . . . 61
3.3.1 Problématique justifiant le filtrage . . . . . 61
3.3.2 Implémentation du filtrage . . . . . 61
3.3.3 Contribution de chaque niveau de filtrage . . . . . 62
3.4 Discussion . . . . . 62
3.4.1 Points forts validés . . . . . 62
3.4.2 Limitations identifiées . . . . . 63
3.5 Conclusion . . . . . 63

**Apendix 66**

# Table des figures


1.1 Méthodologie générale de l'IA . . . . . 17
1.2 Type de machine learning . . . . . 18
1.3 l'apprentissage supervisé . . . . . 19
1.4 La relation entre l'IA, ML et le Deep Learning . . . . . 21
1.5 Représentation d'un perceptron multi-couches . . . . . 22
1.6 (à gauche) Un RNN (à droite) Sa version déroulé Source . . . . . 23
1.7 Architecture d'un réseau de neurones artificiels . . . . . 24
1.8 Les couche CNN . . . . . 25
1.9 La couche de convolution . . . . . 26
1.10 Le fonctionnement de Stride . . . . . 26
1.11 La couche de Pooling . . . . . 27
1.12 La couche Fully Connected (FC) . . . . . 28
1.13 Système d'arrosage intelligent propose par A.DAHANE et al . . . . . 29
1.14 Le rôle des différents processus d'IOT . . . . . 33

2.1 Flux de traitement CNN pour la détection des maladies des feuilles 39
2.2 Évolution de l'accuracy et de la loss . . . . . 42
2.3 Architecture et pipeline CNN pour la classification des maladies des plantes . . . . . 43
2.4 Architecture globale du système . . . . . 45
2.5 Flux de Données Complet . . . . . 47
2.6 Pipeline Prétraitement Image . . . . . 49
2.7 Mécanisme de filtrage triple des prédictions . . . . . 51
2.8 Intégration Flask IoT - Endpoints et Flux . . . . . 52
2.9 Calcul temps de pompage NPK . . . . . 53
2.10 Dosage séquentiel du NPK . . . . . 54
2.11 Page d'accueil . . . . . 55
2.12 Page de reconnaissance des maladies . . . . . 55
2.13 Page Équipe . . . . . 56
2.14 Page À propos . . . . . 56
2.15 Interface Streamlit . . . . . 57
2.16 Interface d'affichage Arduino (TFT 3.5") . . . . . 58
2.17 Prototype final du système intelligent proposé . . . . . 58

# Liste des tableaux

2.1 Évolution de la précision d'entraînement et de validation . . . . . 42
2.2 États de la FSM de dosage séquentiel NPK. . . . . 53

3.1 Limitations observées sans mécanisme de filtrage . . . . . 61
3.2 Comparaison des métriques système avant et après filtrage triple . . 62

# Abreviations list

| Acronyme | Signification |
|---|---|
| ESSAT | École Supérieure des Sciences Appliquées de Tlemcen |
| IA | Intelligence Artificielle |
| ML | Machine Learning (Apprentissage Automatique) |
| CNN | Convolutional Neural Network (Réseau de Neurones Convolutif) |
| ANN | Artificial Neural Network (Réseau de Neurones Artificiel) |
| RNN | Recurrent Neural Network (Réseau de Neurones Récurrent) |
| MLP | Multi-Layer Perceptron (Perceptron Multicouche) |
| LSTM | Long Short-Term Memory |
| GRU | Gated Recurrent Unit |
| SVM | Support Vector Machine |
| IoT | Internet of Things (Internet des Objets) |
| NPK | Azote (N), Phosphore (P), Potassium (K) |
| FSM | Finite State Machine (Machine à États Finis) |
| TFT | Thin-Film Transistor (écran) |
| UART | Universal Asynchronous Receiver-Transmitter |
| REST | Representational State Transfer |
| API | Application Programming Interface |
| TTL | Time To Live |
| SPOF | Single Point of Failure |
| MQTT | Message Queuing Telemetry Transport |
| HTTP/HTTPS | HyperText Transfer Protocol (Secure) |
| CoAP | Constrained Application Protocol |
| AMQP | Advanced Message Queuing Protocol |
| XMPP | Extensible Messaging and Presence Protocol |
| NFC | Near Field Communication |
| BLE | Bluetooth Low Energy |
| VRA | Variable Rate Application |
| RGB | Red Green Blue |
| NLP | Natural Language Processing |
| FAO | Food and Agriculture Organization |

# Introduction générale

L'agriculture représente l'un des secteurs les plus importants pour assurer la sécurité alimentaire et répondre aux besoins croissants d'une population mondiale en constante évolution. Cependant, ce domaine fait aujourd'hui face à plusieurs défis majeurs, notamment la diminution des ressources en eau, les changements climatiques, la dégradation des sols et la nécessité d'améliorer la productivité tout en limitant l'utilisation excessive des ressources naturelles. Dans ce contexte, l'optimisation de la gestion agricole devient une priorité. Les méthodes d'irrigation traditionnelles, souvent basées sur des calendriers fixes ou des estimations humaines, peuvent entraîner une consommation excessive d'eau ou, au contraire, un manque d'apport hydrique pouvant affecter la croissance des plantes. Il devient donc nécessaire de développer des solutions capables d'adapter l'irrigation aux besoins réels des cultures en temps réel. L'évolution rapide des technologies numériques a permis l'émergence de nouvelles approches telles que l'agriculture intelligente (Smart Agriculture), qui repose sur l'intégration de l'Internet des Objets (IoT), de l'intelligence artificielle (IA) et des systèmes automatisés. Grâce aux capteurs connectés, il est désormais possible de collecter différentes informations environnementales comme l'humidité du sol, la température ou l'état de la plante. Ces données peuvent ensuite être analysées par des modèles d'intelligence artificielle afin de faciliter la prise de décision et améliorer les performances des systèmes agricoles.

Par ailleurs, les progrès réalisés dans le domaine du Deep Learning, notamment les réseaux de neurones convolutifs (CNN), ont montré une grande efficacité dans l'analyse d'images et la reconnaissance automatique des maladies des plantes. L'association de ces techniques avec les systèmes IoT permet de concevoir des solutions capables non seulement de surveiller l'environnement de culture, mais aussi de détecter les anomalies et d'agir automatiquement par l'intermédiaire d'actionneurs.

Ainsi, ce travail s'inscrit dans cette perspective en proposant la conception et la réalisation d'un système intelligent d'irrigation automatisée basé sur l'IoT et l'intelligence artificielle. Le système développé vise à assurer le suivi de l'état hydrique des plantes, analyser leur état de santé à partir d'images, et fournir des recommandations adaptées pour optimiser l'irrigation et la fertilisation.

Dans ce cadre, la problématique principale de ce mémoire peut être formulée comme suit : Comment concevoir un système intelligent capable de surveiller l'état des plantes, détecter leurs besoins en eau et en nutriments, puis automatiser l'irrigation grâce à la combinaison de l'intelligence artificielle et de l'Internet des Objets, tout en garantissant une utilisation efficace des ressources agricoles ?

Pour répondre à cette problématique, ce mémoire présente d'abord les fonde-

ments théoriques liés à l'intelligence artificielle, l'agriculture intelligente et les systèmes IoT. Ensuite, il décrit la conception et la méthodologie du système proposé, incluant l'architecture matérielle, le traitement des données et le modèle d'analyse basé sur les réseaux de neurones convolutifs. Enfin, les résultats expérimentaux, les tests réalisés et l'évaluation des performances du système seront présentés et discutés.

CHAPITRE 1

# Fondements Théoriques et État de l'Art

## 1.1 Introduction

Le développement rapide des technologies numériques a profondément transformé de nombreux secteurs d'activité, et l'agriculture ne fait pas exception. Face aux défis croissants liés à la sécurité alimentaire, aux changements climatiques et à la nécessité d'optimiser l'utilisation des ressources naturelles, l'intégration de l'intelligence artificielle et des systèmes connectés dans les pratiques agricoles apparaît aujourd'hui comme une solution prometteuse pour améliorer la productivité et la durabilité des exploitations agricoles. Ce chapitre présente les fondements théoriques et technologiques sur lesquels repose le système proposé dans ce mémoire. Dans un premier temps, nous introduisons les concepts essentiels de l'intelligence artificielle, du Machine Learning et du Deep Learning, ainsi que les différentes approches d'apprentissage utilisées dans les systèmes intelligents. Nous nous intéressons ensuite aux réseaux de neurones artificiels et plus particulièrement aux (CNN), qui constituent le cœur de notre solution de détection des maladies des plantes à partir d'images. Les principes de fonctionnement des CNN, leurs principales couches ainsi que les raisons ayant motivé leur choix sont également présentés. Dans un second temps, nous abordons le domaine de l'agriculture intelligente en présentant les technologies de (IoT), les protocoles de communication réseau et les systèmes de surveillance agricole connectés. Enfin, nous examinons les solutions existantes de

## 1.2 L'intelligence artificielle

IA n'évolue pas en vase clos : c'est un domaine de l'informatique profondément ancré au carrefour de multiples disciplines. Pour fonctionner, elle s'appuie d'abord sur des fondements théoriques rigoureux comme les mathématiques, la logique et les statistiques. Mais pour comprendre l'intelligence, l'IA puise aussi dans les sciences humaines, notamment les sciences cognitives, la psychologie, la philosophie et la linguistique. Elle va même jusqu'à s'inspirer de la neurobiologie en exploitant le biomimétisme pour tenter de reproduire les mécanismes du cerveau humain. Enfin, cette architecture logicielle ne serait rien sans la puissance des technologies matérielles actuelles, qui lui fournissent le support physique indispensable à son exécution [25].

## 1.3 Historique de l'intelligence artificielle

— **1950 :** le mathématicien britannique Alan Turing publie son célèbre article *Computing Machinery and Intelligence*. Il y propose le « jeu de l'imitation », aujourd'hui connu sous le nom de test de Turing, visant à évaluer la capacité d'une machine à imiter le comportement humain lors d'une conversation. Cette contribution est considérée comme l'un des fondements théoriques de l'intelligence artificielle moderne.
— **1974–1980 :** le domaine de l'IA connaît une période de ralentissement appelée « hiver de l'IA ». Les résultats obtenus ne répondent pas aux attentes de l'époque, entraînant une diminution des financements et de l'intérêt des chercheurs et des gouvernements.
— **1980 :** l'Université Stanford accueille la première conférence nationale de l'American Association for Artificial Intelligence. Cet événement marque le retour de l'intérêt scientifique pour l'IA et favorise le développement de nouvelles recherches dans le domaine.
— **1997 :** le programme *Deep Blue*, développé par IBM, réalise un exploit historique en battant le champion du monde d'échecs Garry Kasparov, démontrant la capacité des systèmes informatiques à résoudre des problèmes complexes nécessitant un haut niveau de raisonnement.
— **2002 :** l'arrivée du robot aspirateur *Roomba* marque l'entrée de l'intelligence artificielle dans la vie quotidienne et popularise son usage dans les environnements domestiques.
— **2011 :** le système *Watson* d'IBM remporte le jeu télévisé *Jeopardy !*. Capable de comprendre des questions en langage naturel et d'y répondre correctement, il illustre les progrès du traitement automatique du langage.
— **2014 :** le programme *Eugene Goostman* convainc environ 33 % des juges lors d'une version du test de Turing, représentant une étape importante dans l'évolution des systèmes conversationnels, malgré les débats autour de ce résultat.
— **À partir de 2017 :** les avancées en intelligence artificielle s'accélèrent grâce à l'apprentissage profond, au traitement du langage naturel et à la vision par ordinateur. L'IA est désormais largement utilisée dans la santé, l'agriculture, l'industrie, les transports et l'éducation, améliorant l'efficacité des systèmes et la prise de décision [52].

## 1.4 Méthodologie générale de l'IA

**1- Définition du problème et collecte des données** La première étape consiste à identifier clairement l'objectif du projet et à déterminer le type de tâche à accomplir. Une fois le problème bien défini, on procède à la collecte d'un ensemble de données suffisamment représentatif et varié, car la qualité des données collectées conditionne directement la qualité du modèle final.

**2- Prétraitement et augmentation des données** Avant d'utiliser les données, il est nécessaire de les préparer. Cette étape comprend le nettoyage des données (suppression des valeurs incorrectes ou manquantes), leur normalisation (mise à la même échelle) et leur augmentation (création de nouvelles données à partir des données existantes par diverses transformations). L'objectif est d'améliorer la qualité

et la diversité des données d'entraînement.

**3- Choix et conception du modèle** Cette étape consiste à sélectionner l'architecture du réseau de neurones la plus adaptée au problème traité. Le choix de l'architecture dépend de la nature des données et de la complexité de la tâche. On définit également à ce stade les paramètres structurels du modèle tels que le nombre de couches et les fonctions d'activation.

**4- Entraînement du modèle** Durant cette phase, le modèle apprend à partir des données en ajustant progressivement ses paramètres internes afin de minimiser les erreurs de prédiction. Cet apprentissage repose sur l'optimisation d'une fonction de perte, qui mesure l'écart entre les prédictions du modèle et les valeurs réelles attendues.

**5- Évaluation des performances** Une fois l'entraînement terminé, le modèle est testé sur des données qu'il n'a jamais vues auparavant afin de mesurer objectivement ses performances. Cette évaluation s'appuie sur plusieurs métriques complémentaires telles que l'accuracy, la précision, le rappel et le F1-score, dont le choix dépend des exigences spécifiques du projet.

**6- Déploiement et intégration** Cette étape consiste à intégrer le modèle validé dans l'application cible afin de le rendre accessible aux utilisateurs finaux. Elle comprend également l'optimisation du modèle pour un fonctionnement efficace en conditions réelles, en tenant compte des contraintes de temps de réponse et de ressources disponibles.

**7- Surveillance et ré-entraînement** Après le déploiement, le comportement du modèle est surveillé en continu afin de détecter toute dégradation de ses performances. En effet, les données réelles évoluent avec le temps, ce qui peut réduire la précision du modèle. Dans ce cas, un ré-entraînement est effectué en intégrant de nouvelles données pour maintenir un niveau de performance optimal [32]. La figure 1.1 représente la méthodologie générale de l'IA

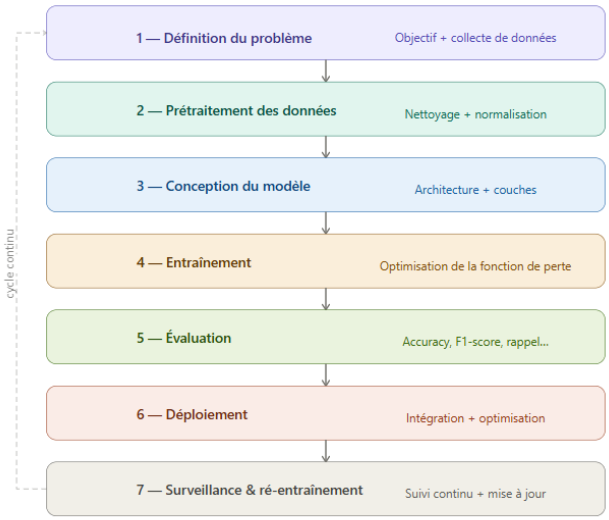



FIGURE 1.1 – Méthodologie générale de l'IA

## 1.5 L'apprentissage automatique (Machine Learning)

Le Machine Learning, ou apprentissage automatique, est une branche incontournable de l'intelligence artificielle.Concevoir des algorithmes capables de donner aux machines la faculté d'apprendre par elles-mêmes. Au lieu d'exécuter un programme figé, l'ordinateur s'entraîne sur des volumes massifs de données pour y repérer des modèles cachés et des corrélations. C'est cette analyse qui lui permet ensuite de faire des prédictions ultra-précises ou de prendre des décisions de manière totalement autonome, en améliorant ses performances au fil du temps [15].

Le processus d'apprentissage automatique se déroule généralement en plusieurs étapes :

**La collecte et le tri des données** : Tout commence par le choix des données. Plus elles sont propres, précises et nombreuses, plus la machine sera intelligente. C'est la fondation même du projet.

**Le choix du modèle** : Il n'existe pas de formule magique universelle. On sélectionne l'algorithme sur mesure, en fonction du problème à résoudre et du type de données à disposition.

**L'entraînement** : C'est la phase intensive où l'on nourrit le modèle avec les données récoltées. La machine s'exerce en boucle et ajuste ses paramètres internes pour réduire ses erreurs au maximum.

**L'évaluation** : Pour vérifier que la machine a réellement compris la logique (et qu'elle ne fait pas que réciter par cœur), on la teste sur de nouvelles données qu'elle n'a encore jamais vues.

**Le déploiement** : Une fois l'examen réussi, le modèle est enfin prêt. On l'intègre dans l'application ou le système final pour qu'il commence à prédire, classer ou prendre des décisions de manière autonome.

### 1.5.1 Types de machine Learning

On distingue quatre grandes approches : l'apprentissage supervisé qui apprend à partir de données étiquetées, l'apprentissage non supervisé qui découvre des structures cachées, l'apprentissage semi-supervisé qui combine les deux, et l'apprentissage par renforcement qui apprend par essais et erreurs. Le choix de la méthode dépend avant tout de la nature des données disponibles et de l'objectif visé.

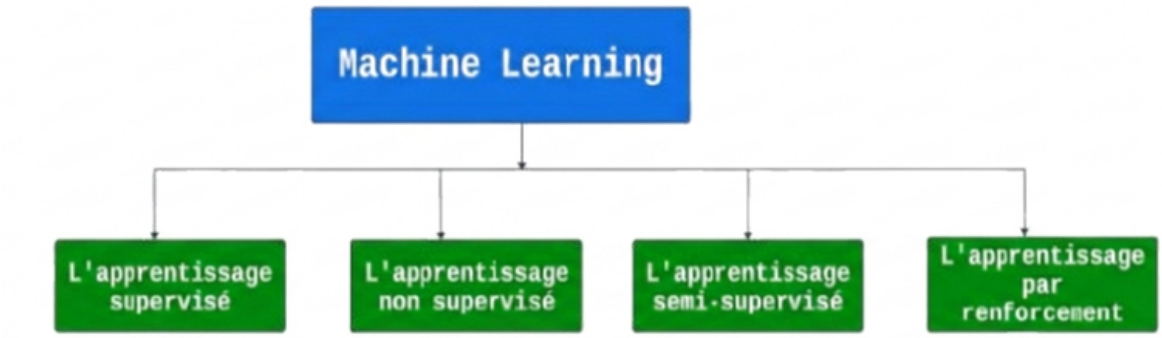



FIGURE 1.2 – Type de machine learning

### 1.5.2 L'apprentissage supervisé

L'apprentissage supervisé est une méthode d'intelligence artificielle dans laquelle un modèle est entraîné à partir de données déjà annotées. Chaque exemple d'apprentissage est associé à une réponse correcte, appelée étiquette. En analysant ces exemples, le modèle apprend les relations entre les données d'entrée et les résultats attendus afin de pouvoir prédire correctement les réponses pour de nouvelles données qu'il n'a jamais vues auparavant. Cette approche est largement utilisée dans des tâches telles que la classification, la régression et de nombreux problèmes de prédiction [30] . La figure 1.3 représente l'apprentissage supervisé

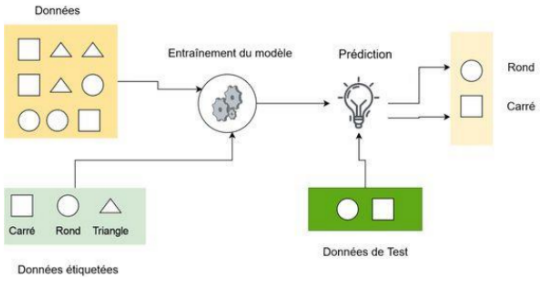



FIGURE 1.3 – L'apprentissage supervisé [9]

L'apprentissage supervisé a été largement exploité dans de nombreux travaux de recherche. Cortes et Vapnik ont introduit les machines à vecteurs de support (SVM), une méthode de classification reposant sur la recherche d'un hyperplan optimal séparant les classes [16] .

### 1.5.3 Apprentissage non supervisé

L'apprentissage non supervisé est une approche de l'intelligence artificielle dans laquelle le modèle travaille sur des données qui ne possèdent pas d'étiquettes ou de réponses prédéfinies. Son objectif est d'analyser ces données afin d'identifier automatiquement des structures, des similitudes ou des relations cachées entre les différents éléments. Cette méthode est particulièrement utile pour regrouper des données présentant des caractéristiques communes (clustering) ou pour simplifier des ensembles de données complexes en réduisant leur nombre de variables (réduction de dimensionnalité). Elle permet ainsi d'explorer et de mieux comprendre les données sans intervention humaine directe [12].

De nombreux chercheurs se sont intéressés à l'apprentissage non supervisé comme approche permettant d'extraire des connaissances à partir de données non étiquetées.Lloyd a développé une version améliorée de l'algorithme K-means, largement utilisée pour le regroupement de données en clusters homogènes [39] .

### 1.5.4 L'apprentissage semi-supervisé

L'apprentissage semi-supervisé est une approche qui combine les principes de l'apprentissage supervisé et de l'apprentissage non supervisé. Lors de l'entraînement, le modèle exploite un petit ensemble de données étiquetées, accompagné

d'une grande quantité de données non étiquetées. Cette méthode permet d'améliorer les performances du modèle tout en réduisant le besoin d'annoter manuellement un grand volume de données. Elle est particulièrement intéressante dans les situations où l'obtention de données étiquetées demande beaucoup de temps, d'efforts ou de ressources, alors que des données non étiquetées sont facilement disponibles [14].

Face à la difficulté d'obtenir des données entièrement étiquetées, de nombreux chercheurs ont eu recours à l'apprentissage semi-supervisé pour tirer profit des grandes quantités de données non étiquetées disponibles.

Nigam et al. ont utilisé l'algorithme Expectation-Maximization (EM) combiné au classificateur Naive Bayes pour la classification de textes, en exploitant à la fois des documents étiquetés et non étiquetés [49] .

### 1.5.5 Apprentissage par renforcement

L'apprentissage par renforcement est une méthode d'intelligence artificielle dans laquelle un agent apprend en interagissant directement avec son environnement. À chaque étape, il choisit une action, observe les conséquences de cette action et reçoit une récompense ou une pénalité selon le résultat obtenu. Grâce à ce mécanisme d'essais et d'erreurs, l'agent améliore progressivement sa stratégie de décision afin de maximiser les récompenses cumulées sur le long terme. Cette approche est particulièrement utilisée dans les domaines de la robotique, des jeux vidéo, de la conduite autonome et de l'optimisation de systèmes complexes [57].

Dans le domaine de l'agriculture, l'apprentissage par renforcement a été exploité comme une approche intelligente permettant d'optimiser les pratiques agricoles et d'améliorer la productivité tout en réduisant les coûts.

Alibabaei et al. ont appliqué l'apprentissage par renforcement pour optimiser l'irrigation des cultures, où un agent apprend à décider automatiquement la quantité d'eau à apporter en fonction des conditions météorologiques et de l'état du sol [3] .

## 1.6 Définition et Principe de l'Apprentissage Profond (Deep Learning)

IA est un domaine vaste qui englobe plusieurs approches permettant aux machines de simuler des capacités cognitives humaines. Parmi ces approches, le Machine Learning occupe une place centrale : plutôt que de suivre des instructions rigides définies par un programmeur, un système d'apprentissage automatique analyse un ensemble de données pour en extraire des règles et des patterns, qu'il applique ensuite pour traiter de nouvelles situations inconnues.

Le Deep Learning, ou apprentissage profond, pousse cette logique encore plus loin. Il s'agit d'une branche du Machine Learning qui s'inspire directement du fonctionnement du cerveau humain. Son principe repose sur des réseaux de neurones artificiels organisés en plusieurs couches superposées parfois des dizaines, voire des centaines où chaque couche reçoit les informations de la couche précédente, les traite et les transmet à la suivante. C'est cette profondeur de traitement qui donne à ces modèles leur capacité remarquable à détecter des structures complexes dans les données, qu'il s'agisse d'images, de sons ou de valeurs numériques.

Ces trois concepts s'inscrivent dans une relation hiérarchique claire : le Deep Learning est un sous-ensemble du Machine Learning, qui est lui-même un sous-ensemble de l'Intelligence Artificielle, comme l'illustre la figure ci-dessous 1.4 [20] .

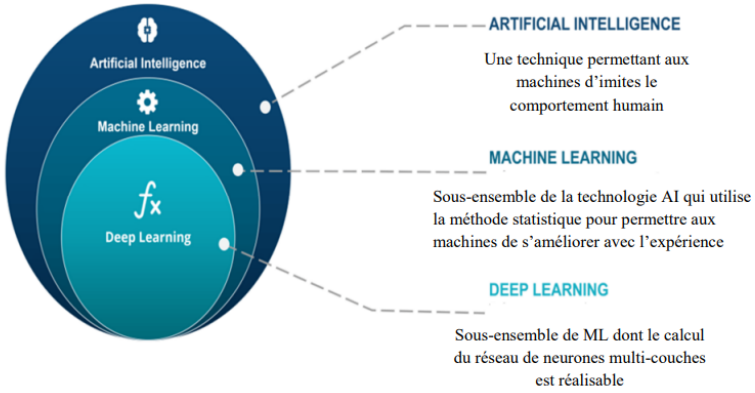



FIGURE 1.4 – La relation entre l'IA, ML et le Deep Learning [23]

## 1.7 Les Réseaux de Neurones

### 1.7.1 Typologie des réseaux neuronaux

Il existe beaucoup de types de réseaux neurones, chaque type étant développé pour un objectif particulier.

#### 1.7.1.1 Neurone Formel

Pour comprendre comment fonctionne un réseau de neurones artificiel, il faut d'abord s'intéresser à son unité de base : le neurone formel. Comme son nom l'indique, il s'agit d'une modélisation mathématique et informatique du neurone biologique que l'on trouve dans le cerveau humain.

À l'image de son homologue biologique, le neurone formel reçoit plusieurs signaux d'entrée qui correspondent aux dendrites et produit un unique signal de sortie, analogue à ce que génère l'axone. Les connexions entre neurones, appelées synapses dans le monde biologique, sont ici représentées par des poids synaptiques : des coefficients numériques qui déterminent l'importance accordée à chaque entrée. Certains poids amplifient le signal (effet excitateur), d'autres l'atténuent (effet inhibiteur). C'est précisément lors de la phase d'apprentissage que ces poids sont ajustés progressivement, jusqu'à ce que le modèle produise des résultats satisfaisants.

Le fonctionnement interne du neurone formel est relativement simple : il calcule d'abord la somme pondérée de toutes ses entrées, puis applique à ce résultat une fonction d'activation généralement non linéaire pour produire sa valeur de sortie.

Pris individuellement, un neurone formel ne peut accomplir que des tâches très limitées. Mais lorsqu'ils sont interconnectés en grand nombre au sein d'un réseau de neurones artificiels, ils deviennent capables de modéliser des fonctions d'une

grande complexité, ouvrant ainsi la voie à de nombreuses applications en intelligence artificielle [43].

#### 1.7.1.2 Neurones multicouches

Le Perceptron Multicouche ( MLP) est l'une des architectures les plus fondamentales des réseaux de neurones artificiels. Il s'agit d'un réseau organisé en plusieurs couches successives : une couche d'entrée qui reçoit les données brutes, une ou plusieurs couches intermédiaires dites couches cachées, et enfin une couche de sortie qui produit le résultat final.

Ce qui caractérise le MLP, c'est la manière dont l'information y circule : elle se propage toujours dans un seul sens, de la couche d'entrée vers la couche de sortie, sans jamais revenir en arrière. C'est ce qu'on appelle une architecture de type feedforward, ou réseau à propagation avant.

Chaque couche est composée d'un certain nombre de neurones formels, et chaque neurone d'une couche est connecté à tous les neurones de la couche suivante. Le nombre de neurones dans chaque couche peut varier selon la complexité du problème à résoudre. La seule contrainte fixe concerne la couche de sortie : le nombre de ses neurones correspond directement au nombre de sorties attendues du système par exemple, le nombre de classes dans un problème de classification.

Grâce à cette structure en profondeur, le MLP est capable d'apprendre des relations complexes et non linéaires entre les données d'entrée et les sorties souhaitées, ce qui en fait un outil puissant pour de nombreuses tâches d'intelligence artificielle [22].La figure 1.5 Représente la perceptron multi-couches

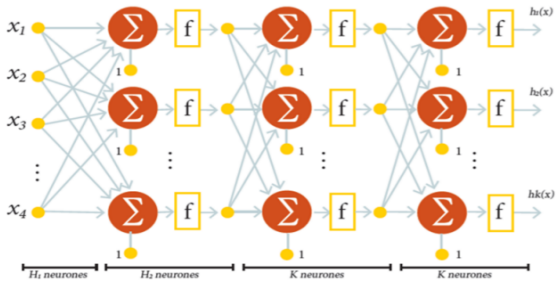



FIGURE 1.5 – Représentation d'un perceptron multi-couches [23]

### 1.7.2 Neurones récurrents(RNN)

Le Réseau de Neurones Récurrent (RNN) est une architecture de deep learning conçue spécifiquement pour traiter des données séquentielles et des séries temporelles, c'est-à-dire tout type de données dont l'ordre et le contexte jouent un rôle essentiel dans la compréhension.

Ce qui distingue fondamentalement le RNN des autres architectures, c'est sa capacité à simuler une forme de mémoire. Là où un réseau de neurones classique traite chaque entrée de manière totalement indépendante, le RNN conserve une trace des informations passées et s'en sert pour influencer le traitement des données présentes. Autrement dit, la sortie produite à un instant donné ne dépend pas uniquement de l'entrée actuelle, mais aussi de tout ce qui s'est passé avant dans la séquence.

Cette propriété en fait un outil particulièrement puissant pour des tâches comme la traduction automatique, la reconnaissance vocale, le traitement du langage naturel (NLP) ou encore le sous-titrage d'images. On retrouve d'ailleurs les RNN au cœur d'applications largement utilisées au quotidien, telles que Siri, Google Translate ou la recherche vocale de Google.

Cependant, les RNN présentent une limitation notable : dans leur forme unidirectionnelle, ils ne traitent la séquence que dans un seul sens, du passé vers le présent. Ils sont donc incapables de tenir compte des éléments qui apparaissent plus tard dans la séquence, alors que ces informations futures pourraient pourtant s'avérer précieuses pour mieux interpréter certains contextes. C'est précisément pour pallier cette limite que des variantes bidirectionnelles et des architectures plus avancées, comme les LSTM et les GRU, ont été développées par la suite [18]. La figure 1.6 represente un RNN

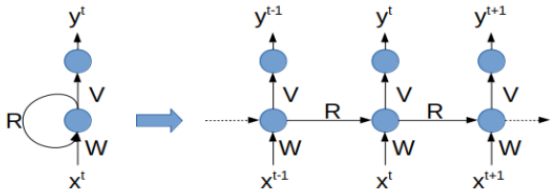



FIGURE 1.6 – (à gauche) Un RNN (à droite) Sa version déroulé Source [23]

### 1.7.3 Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels, également appelés réseaux neuronaux, sont des modèles informatiques qui s'inspirent directement du fonctionnement du cerveau humain. L'idée centrale est simple mais puissante : reproduire, de manière mathématique, la façon dont les neurones biologiques communiquent et collaborent pour traiter l'information.

Concrètement, un réseau de neurones est constitué d'un ensemble d'unités de calcul élémentaires les neurones artificiels reliées entre elles par des connexions. Chaque neurone reçoit un ou plusieurs signaux d'entrée, leur attribue une importance relative grâce à des poids numériques, puis combine ces signaux pour produire un signal de sortie qui sera à son tour transmis aux neurones de la couche suivante.

C'est justement ces poids qui constituent le cœur de l'intelligence du réseau. Durant la phase d'apprentissage, le modèle analyse des exemples, compare ses prédictions aux résultats attendus, et ajuste progressivement ces poids pour minimiser les erreurs commises. Ce processus est répété des milliers de fois jusqu'à ce que le réseau atteigne un niveau de performance satisfaisant sur la tâche qui lui est assignée.

En résumé, un réseau de neurones artificiel apprend de la même façon qu'un être humain apprend de ses erreurs : par l'expérience, la répétition et la correction continue [30] . La figure 1.7 represente un réseau de neurones artificiels

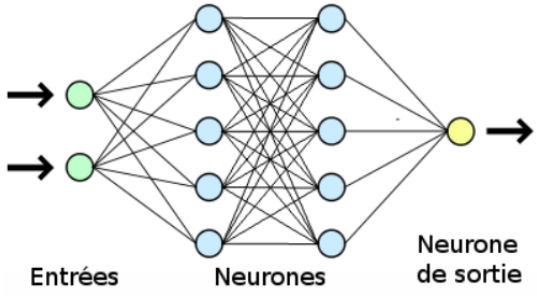



FIGURE 1.7 – Architecture d'un réseau de neurones artificiels [53]

### 1.7.4 Les réseaux de neurones convolutifs CNN

CNN s'inscrivent dans le cadre de l'apprentissage profond, lui-même constituant une sous-discipline de l'intelligence artificielle. Selon Yann LeCun, l'intelligence artificielle désigne l'ensemble des méthodes permettant aux machines d'accomplir des tâches ou de résoudre des problèmes qui relèvent habituellement des capacités humaines ou animales. Son objectif est de doter les systèmes d'une capacité d'adaptation à des situations variées, y compris celles qui n'ont pas été explicitement programmées. Ainsi, en reconnaissance d'images par exemple, un modèle entraîné sur des millions d'images préalablement catégorisées est capable de classer correctement une image inédite, à condition qu'elle appartienne à l'une des catégories apprises lors de l'entraînement.

## 1.8 Justification du choix de la méthode CNN

Le choix CNN a été motivé par leur excellente performance dans les tâches de classification d'images. Dans notre projet, la détection des maladies des plantes repose sur l'analyse visuelle des feuilles, dont les symptômes apparaissent sous forme de taches, de décolorations, de moisissures ou de déformations. Les CNN sont particulièrement adaptés à ce type de problème car ils permettent d'extraire automatiquement les caractéristiques pertinentes à partir des images, sans nécessiter une conception manuelle des descripteurs visuels [38].

De plus, plusieurs travaux scientifiques ont démontré la supériorité des CNN par rapport aux méthodes classiques de vision par ordinateur pour la reconnaissance d'objets et la classification d'images [37].

Cette approche est également largement utilisée dans le domaine agricole pour la détection automatique des maladies des plantes à partir d'images de feuilles [47].

Enfin, notre système s'appuie sur une base de données importante d'environ 87 000 images réparties sur 38 classes, ce qui permet aux CNN d'apprendre efficacement les différentes caractéristiques visuelles des maladies. Les résultats obtenus, avec une précision de validation de 95,1 %, confirment la pertinence de ce choix pour une intégration dans un système IoT intelligent de surveillance et de diagnostic des cultures.

Une fois le choix des CNN justifié, nous présentons dans la section suivante leur structure interne et les différentes couches qui composent cette architecture de Deep Learning.

## 1.9 Les réseaux de neurones convolutifs

CNN occupent aujourd'hui une place centrale dans le domaine de l'apprentissage profond, notamment grâce à leurs performances remarquables dans l'analyse d'images. Cependant, leur fonctionnement est souvent mal compris : contrairement à une idée reçue, les CNN ne reposent pas uniquement sur des multiplications matricielles classiques. Certaines architectures intègrent également des connexions résiduelles, aussi appelées skip connections, qui permettent à l'information de "sauter" certaines couches afin de faciliter l'apprentissage de représentations profondes.

Au cœur de ces réseaux se trouve l'opération de convolution. En mathématiques, il s'agit d'une opération qui combine deux fonctions pour en produire une troisième, traduisant la façon dont l'une vient modifier la forme de l'autre. Appliquée au traitement d'images, cette opération consiste à faire glisser des filtres sur des régions locales de l'image afin d'en extraire des caractéristiques pertinentes. C'est précisément ce mécanisme qui confère aux CNN leur capacité à détecter des motifs visuels complexes, qu'il s'agisse de contours, de textures ou de structures plus abstraites [41].

CNN s'articule autour de trois types de couches fondamentales : les couches convolutives, les couches de regroupement (pooling) et les couches entièrement connectées (fully connected). C'est l'agencement ordonné de ces couches les unes après les autres qui donne naissance à ce que l'on appelle une architecture CNN selon la figure 1.8 :

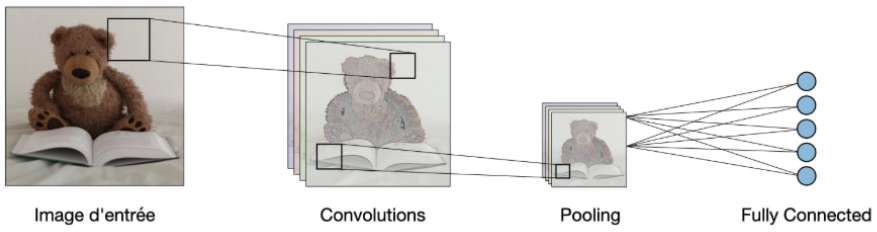



FIGURE 1.8 – Les couche CNN [5]

## 1.10 Les couches de réseaux de neurones convolutifs

### 1.10.1 La couche de convolution

La couche de convolution (CONV) est le cœur du réseau. Elle applique des filtres qui balayent l'image d'entrée pour en extraire des caractéristiques locales. Le résultat produit est appelé feature map ou activation map. Cette couche se règle principalement via deux paramètres : la taille du filtre F et le pas de déplacement (stride) S [58].La figure 1.9 represente la couche de convolustion

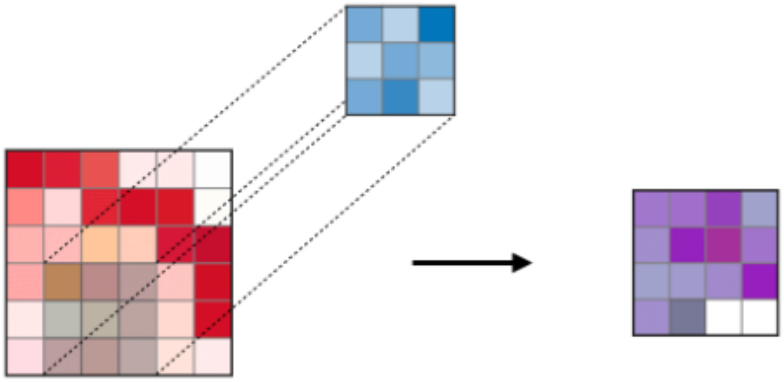


FIGURE 1.9 – La couche de convolution [5]

La configuration d'un CNN repose sur le choix de trois hyperparamètres fondamentaux, qui influencent directement la qualité de l'apprentissage.

Le premier est **la profondeur de la couche**, qui correspond au nombre de noyaux de convolution appliqués. Chaque noyau est associé à un ensemble de neurones partageant le même champ récepteur. Plus la profondeur est grande, plus le réseau est capable de détecter des caractéristiques visuelles fines et complexes.

Le deuxième est le pas (stride) représente dans la figure 1.10, qui détermine de combien de pixels le filtre se déplace à chaque étape. Un pas réduit engendre un chevauchement plus important entre les régions analysées, ce qui produit en sortie un volume de données plus volumineux et plus détaillé.




FIGURE 1.10 – Le fonctionnement de Stride [5]

Le troisième est le zéro-padding (ou rembourrage), qui consiste à entourer l'image d'entrée d'une bordure de zéros. Ce mécanisme permet de maîtriser les dimensions spatiales du volume de sortie, et surtout de préserver les informations situées aux bords de l'image, qui seraient autrement sous-exploitées par les filtres.

En définitive, ces trois hyperparamètres profondeur, stride et padding ne sont pas de simples réglages techniques. Ils conditionnent la façon dont le réseau perçoit et traite l'image, et leur choix judicieux est déterminant pour les performances globales du modèle [13].

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 30 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix}$$

Pour appliquer la convolution, on commence par positionner le premier filtre sur le coin supérieur gauche de l'image d'entrée, en le superposant à une zone de même taille. L'idée est simple : on multiplie chaque valeur du filtre par la valeur

correspondante dans l'image, puis on additionne tous ces produits pour obtenir un seul résultat.

Par exemple, la valeur 7 dans la carte de sortie est obtenue de la manière suivante : (1×1) + (1×0) + (4×0) + (6×1) = 7

Autrement dit, on fait le produit terme à terme des deux matrices superposées, et la somme de ces produits donne la valeur finale à cet emplacement précis.

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 30 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & \\ & \end{bmatrix}$$

Une autre opération de convolution est menée pour la prochaine sous-matrice :

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 30 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 5 \\ & \end{bmatrix}$$

Le même processus se répète jusqu'à ce que la carte des caractéristiques du filtre donné soit produite

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 30 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 5 & 9 \\ 4 & 7 & 9 \\ 32 & 2 & 5 \end{bmatrix}$$

### 1.10.2 La couche de Pooling (POOL)

La couche de pooling est une opération de sous-échantillonnage typiquement appliquée après une couche convolutionnelle. Elle permet de réduire la taille spatiale des feature maps tout en conservant les informations essentielles [31].

Les deux types principaux représente dans le tableau 1.11

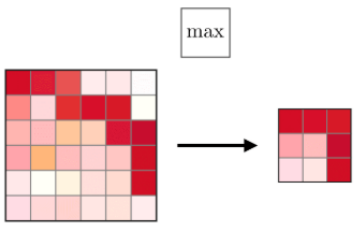
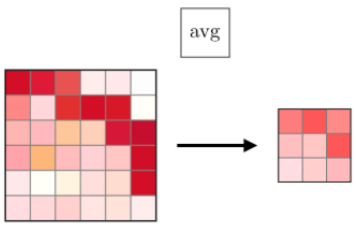
| **Type** | Max pooling | Average pooling |
|---|---|---|
| **But** | Chaque opération de pooling sélectionne la valeur maximale de la surface | Chaque opération de pooling sélectionne la valeur moyenne de la surface |
| **Illustration** |   |   |
| **Commentaires** | • Garde les caractéristiques détectées<br>• Plus communément utilisé | • Sous-échantillonne la *feature map*<br>• Utilisé dans LeNet |

FIGURE 1.11 – La couche de Pooling

### 1.10.3 La couche Fully Connected (FC)

FC connected représenter dans la figure 1.12 s'applique sur une entrée préalablement aplatie, où chaque entrée est connectée à tous les neurones. Ces couches sont typiquement présentes à la fin des architectures CNN et peuvent être utilisées pour optimiser des objectifs tels que les scores de classe [7] .

Avant d'arriver à cette couche, les feature maps produites par les couches de convolution et de pooling sont transformées en un vecteur unidimensionnel (opération appelée flatten ou aplatissement). Ce vecteur est ensuite transmis à la couche fully connected. Chaque neurone de cette couche :

-Reçoit toutes les valeurs du vecteur d'entrée

-Applique une combinaison linéaire suivie d'une fonction d'activation

-Contribue à produire la prédiction finale [7]

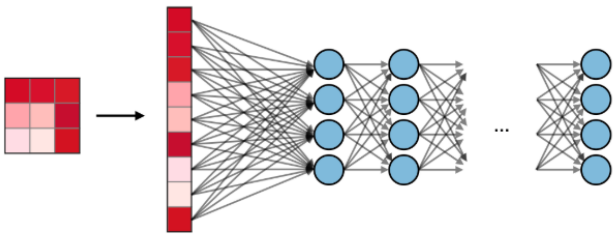


FIGURE 1.12 – La couche Fully Connected (FC)

## 1.11 Analyse des systèmes d'irrigation intelligente préexistants

La plateforme INTEL-IRRIS represente dans la figure 1.13 est un système d'irrigation intelligent basé sur IoT, le Fog Computing, le Cloud Computing et le Deep Learning. Elle collecte des données environnementales à l'aide de capteurs mesurant notamment l'humidité du sol, la température et l'humidité de l'air. Les données sont ensuite transmises à une plateforme de traitement où des modèles d'intelligence artificielle analysent leur qualité et prévoient l'évolution des conditions environnementales.

Pour la détection des anomalies dans les données des capteurs, les auteurs utilisent des modèles Autoencoder (AE) et Generative Adversarial Network (GAN). La prédiction des paramètres environnementaux est réalisée à l'aide d'une architecture hybride CNN-BiLSTM, capable d'analyser les séries temporelles et de prévoir les besoins futurs en irrigation. Grâce à cette approche, le système améliore la gestion de l'eau, réduit les erreurs de mesure et permet une irrigation plus efficace et plus précise [10].

-Mohanty et al. (2016) ont développé un système de reconnaissance automatique des maladies des plantes basé sur un réseau de neurones convolutif profond (CNN), entraîné sur la base de données publique PlantVillage composée de 54 306 images de feuilles saines et malades, collectées en conditions contrôlées et couvrant 14 espèces végétales et 26 maladies différentes. Les auteurs ont comparé deux architectures

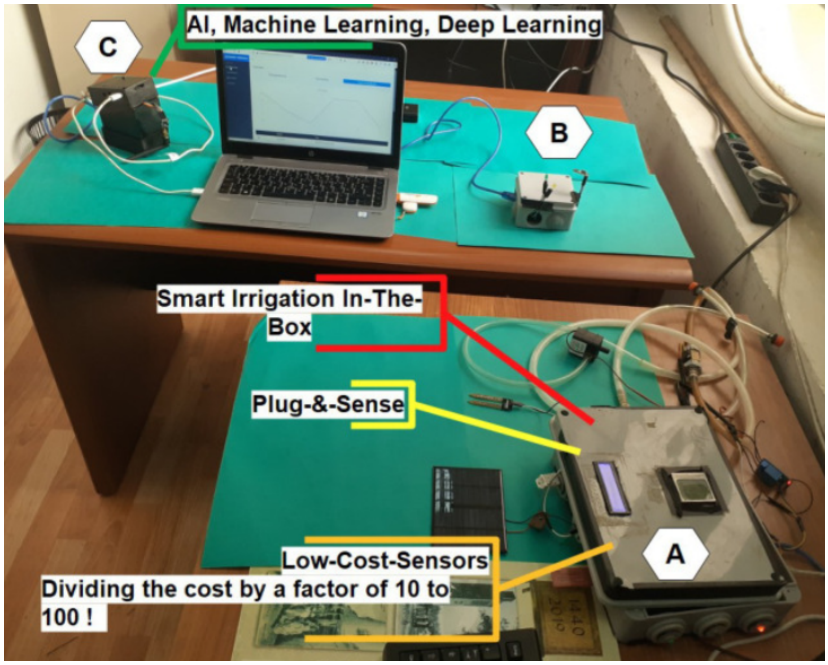



FIGURE 1.13 – Système d'arrosage intelligent propose par A.DAHANE et al [10]

CNN reconnues, GoogLeNet et AlexNet, en utilisant le transfer learning pour adapter ces modèles à la classification des maladies. Le modèle le plus performant a atteint une précision remarquable de 99,35 % sur l'ensemble de test. Toutefois, cette précision chute à 31,4 % lorsque le modèle est évalué sur des images issues de conditions réelles différentes de celles de l'entraînement, soulignant la nécessité d'enrichir les données d'entraînement pour améliorer la généralisation du modèle en conditions terrain [45] .

-Ferentinos (2018) a exploité CNN pour la détection et le diagnostic automatique des maladies des plantes à partir d'images de feuilles. Il a évalué et comparé plusieurs architectures CNN bien établies, notamment VGG, AlexNet et Inception, afin d'identifier la plus performante pour cette tâche de classification. L'étude a également reposé sur la base de données PlantVillage, largement utilisée dans ce domaine. Le meilleur modèle testé a atteint une précision de classification de 99,53% [28] .

-Jiang et al. ont proposé une approche hybride combinant un CNN léger et un SVM pour la détection des maladies des feuilles de riz, en utilisant une base de données de 2 870 images réparties en 4 classes de maladies. L'architecture CNN allégée a été spécialement conçue pour fonctionner sur des appareils mobiles à ressources limitées, tandis que l'intégration du SVM comme classificateur final a permis d'améliorer la précision de classification par rapport à l'utilisation du CNN seul. Cette approche illustre l'intérêt des modèles hybrides légers pour un déploiement efficace en conditions terrain [35] .

## 1.12 Agriculture Intelligente (Smart Agriculture)

### 1.12.1 Définition de l'Agriculture Intelligente

Derrière les expressions un brin techniques de smart agriculture ou smart farming, se cache une réalité bien concrète : l'entrée de plein fouet du monde agricole dans l'ère du numérique. Concrètement, il s'agit d'utiliser les technologies de l'information et de la communication (TIC) pour piloter plus finement les cultures et les élevages. Contrairement à l'agriculture traditionnelle, cette approche s'appuie sur la collecte et l'analyse de données massives pour prendre des décisions ultra-précises, ultra-rapides et souvent automatisées .

-Pour les chercheurs Wolfert et al, cette transition consiste à :

« Appliquer les technologies modernes de l'information à l'agriculture afin de rendre la production alimentaire plus efficace, grâce à la précision et à l'automatisation. »[60]

En clair, cela permet deux choses cruciales : doser les ressources (eau, engrais) au gramme près et réagir instantanément dès que la météo ou l'état d'une plante change.

-Au cœur de cette révolution : trois piliers technologiques Comme le soulignent Sundmaeker et al, l'agriculture intelligente n'est pas née de nulle part. Elle est le fruit de la rencontre entre trois grandes révolutions technologiques [56] :

**IoT** : des capteurs connectés partout dans les champs ou sur les animaux ;

**Le big data** : des montagnes de données récoltées en temps réel ;

**IA** : des algorithmes capables de donner du sens à ces données pour guider l'agriculteur.

Pour l'agriculteur, le quotidien change radicalement. Il ne passe plus seulement ses journées sur le terrain à exécuter des tâches manuelles ; il devient le manager d'un véritable écosystème connecté.

-Relever le défi climatique.Enfin, cette mutation ne se limite pas à une simple course au rendement. La FAO [26] insiste sur un point capital en parlant d'« agriculture intelligente face au climat ». L'objectif est triple et profondément humain : nourrir une population croissante, rendre nos systèmes alimentaires capables de résister aux crises climatiques, et réduire l'empreinte carbone globale du secteur.

En somme, rendre l'agriculture plus connectée pour la rendre plus durable.

### 1.12.2 Domaines d'Application de la Smart Agriculture

L'agriculture intelligente s'invite aujourd'hui à chaque étape de la vie d'une exploitation, avec des applications concrètes qui changent radicalement le quotidien des producteurs :

— **Piloter l'irrigation au goutte-à-goutte** : c'est l'usage le plus courant. En connectant des capteurs d'humidité au sol à des algorithmes intelligents, on ne déclenche l'arrosage que si la plante en a vraiment besoin, et avec la quantité exacte requise. Des technologies comme CropX ou AquaSpy combinent par exemple des sondes multi-profondeurs et des modèles prédictifs pour économiser l'eau [46]. Et les résultats sont là : des chercheurs ont prouvé qu'une irrigation ainsi régulée réduit la consommation d'eau de 30 à 50 % sans impacter les récoltes [36].

— **Repérer les maladies et les ravageurs** : grâce au deep learning (l'apprentissage profond), l'analyse d'images permet de détecter très tôt les attaques d'insectes, les champignons ou les manques de nutriments. Un modèle basé sur CNN [48] a d'ailleurs réussi à identifier 26 maladies sur 14 espèces de plantes avec une précision de 99,35 %. Côté terrain, des drones survolent désormais des centaines d'hectares en un clin d'œil pour cartographier précisément les zones à traiter [27].
— **Suivre la croissance et prédire les récoltes** : équipés de caméras multispectrales et de capteurs LiDAR, les outils modernes mesurent la biomasse, la hauteur des cultures et la surface des feuilles (LAI). Estimer le rendement final plusieurs semaines avant de sortir la moissonneuse. En analysant les images de drones via le deep learning, on arrive par exemple à estimer la récolte du maïs avec une marge d'erreur inférieure à 8 % [21].
— **Fertiliser** : en croisant les données de conductivité électrique du sol et les images satellites, on peut ajuster les apports en azote, phosphore et potassium zone par zone, plutôt que de traiter tout le champ de la même manière. Cette technique de modulation de dose (appelée VRA) permet de réduire la facture d'engrais de 10 à 25 % tout en protégeant les rendements[8].

### 1.12.3 Avantages de l'Agriculture Intelligente

Adopter une approche technologique change complètement la donne pour les producteurs. Les avantages de la smart agriculture se font ressentir à tous les niveaux :

— **Faire la chasse au gaspillage :** en ciblant les besoins réels des cultures, on réduit significativement la consommation d'eau et l'usage des engrais, ce qui représente un avantage à la fois économique et environnemental.
— **Booster les récoltes :** en maintenant les plantes dans des conditions optimales, les rendements s'améliorent naturellement et de manière significative.
— **Alléger l'empreinte écologique :** la réduction des excès d'intrants limite la pollution et contribue à diminuer les émissions agricoles de protoxyde d'azote ($N_2O$).
— **Proposer de meilleurs produits :** grâce à un suivi précis tout au long du cycle de croissance, les récoltes sont plus homogènes et présentent une meilleure qualité nutritionnelle.
— **Redonner du temps aux agriculteurs :** l'automatisation des tâches répétitives permet aux exploitants de se concentrer davantage sur la gestion stratégique de leur exploitation.
— **Faire face aux aléas climatiques :** les modèles prédictifs offrent la possibilité d'adapter rapidement les pratiques agricoles aux conditions météorologiques afin de mieux protéger les cultures.
— **Garantir une transparence totale :** l'enregistrement numérique des données du champ facilite la traçabilité des productions et simplifie l'obtention des certifications et labels de qualité [1].

## 1.13 Internet des objets

### 1.13.1 Definition

IoT, pour son acronyme anglais Internet of Things est une expression née en 1999 sous la plume de Kevin Ashton, alors cofondateur et directeur de l'Auto-ID Center. L'idée maîtresse est de répertorier et de connecter au réseau mondial tous ces appareils du quotidien qui ne sont ni des ordinateurs, ni des smartphones. Qu'il s'agisse de nos voitures, de nos bracelets connectés, de nos réfrigérateurs ou même de nos installations électriques, tous font désormais partie de cette grande toile et échangent des données en continu grâce à leurs capteurs.

En réalité, l'IoT représente l'évolution logique et naturelle de notre façon d'utiliser les réseaux technologiques. Son grand objectif est de rendre le monde physique qui nous entoure bien plus intelligent, simplement en connectant les objets pour qu'ils récoltent et transmettent des informations [6].

Dès lors qu'un appareil est autonome, qu'il possède sa propre adresse IP pour communiquer et qu'il peut être piloté à distance, il rejoint officiellement la grande famille de l'Internet des objets. C'est, en somme, la technologie qui donne une voix et une intelligence aux objets ordinaires pour nous faciliter la vie [6].

### 1.13.2 Architecture de l'internet d'objet

Dès la fin des années 1990, le lancement du protocole internet IPv6 a ouvert la voie à ce que l'on appelle aujourd'hui l'Internet des objets . À l'époque, le défi était déjà de taille : il fallait créer un réservoir d'adresses IP gigantesque, bien plus grand que le précédent, pour pouvoir attribuer une identité unique et connecter des milliards d'objets du quotidien à Internet [50].

### 1.13.3 Les différents processus de l'internet des objets

-**Capter** : c'est le point de départ. Cela consiste à traduire une réalité physique du monde réel (comme la température, l'humidité ou une pression) en un signal numérique que les machines peuvent comprendre.

-**Concentrer** : cette étape sert de passerelle. Elle permet de connecter un réseau d'objets très spécifiques aux réseaux internet standards que nous utilisons tous les jours, comme le Wi-Fi.

-**Stocker** : c'est la mémoire du système. Il s'agit de rassembler et de mettre à l'abri toutes les données brutes qui arrivent en continu, en temps réel et de manière imprévisible, tout en leur ajoutant des étiquettes (les métadonnées) pour pouvoir s'y retrouver plus tard.

-**Présenter** : c'est l'interface humaine. L'objectif est de traduire toutes ces montagnes de chiffres complexes en graphiques et en tableaux clairs, faciles à lire pour nous, tout en nous donnant les outils nécessaires pour agir et piloter nos équipements à distance [6]. La figure 1.14 représente les différents processus de l'internet des objets

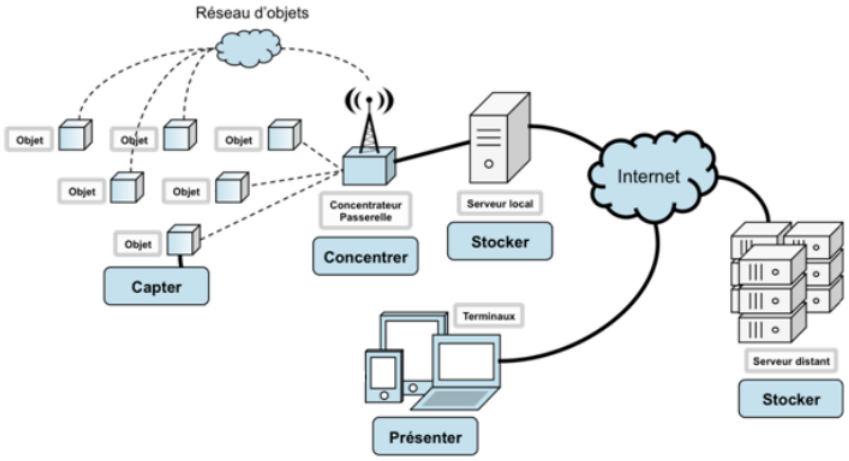



FIGURE 1.14 – Le rôle des différents processus d'IOT [50]

## 1.14 Les technologies du réseau

Pour que l'IoT fonctionne, il faut un ingrédient essentiel : la connectivité. C'est grâce aux technologies réseau que nos objets connectés peuvent enfin 'parler' entre eux et échanger des données. Jetons un œil aux principales solutions qui rendent tout cela possible :

### 1.14.1 Courte portée

Quand des objets connectés ont besoin de discuter alors qu'ils sont juste à côté les uns des autres (généralement à moins de quelques dizaines de mètres), on fait appel aux technologies de courte portée. Leur grand point fort ? Elles consomment très peu d'énergie, ce qui est parfait pour prolonger la durée de vie des batteries. Parmi les solutions les plus connues, on trouve :

#### 1.14.1.1 Bluetooth

Il utilise des ondes radio pour connecter nos appareils et faire transiter des données dans les deux sens, sur de très courtes distances. C'est le champion incontournable pour relier nos smartphones à nos écouteurs ou à nos montres connectées [40].

#### 1.14.1.2 Zigbee

Ce protocole a été pensé sur mesure pour les maisons intelligentes. Son but ? Permettre à toute une panoplie d'appareils connectés (ampoules, capteurs) de discuter entre eux sur de petites distances sans jamais vider leurs piles.

#### 1.14.1.3 Z-Wave

Le grand rival du Zigbee, lui aussi spécialisé dans la domotique. Il est très apprécié pour piloter les installations de la maison (chauffage, sécurité) mais trouve aussi une place de choix dans le secteur de la santé connectée [17].

#### 1.14.1.4 NFC

C'est la technologie du "contact". Elle permet d'échanger des données à une distance minuscule, de l'ordre de quelques centimètres à peine. C'est exactement ce qui se passe lorsque vous payez avec votre smartphone ou que vous validez votre pass Navigo d'un simple geste [54].

### 1.14.2 Moyenne portée

#### 1.14.2.1 Wi-Fi

Inutile de le présenter, il est partout. C'est lui qui gère nos connexions du quotidien sur des distances courtes à moyennes. Ordinateurs, smartphones, tablettes et objets connectés : tout le monde s'y branche pour surfer sur Internet, streamer des vidéos, partager des fichiers ou piloter la maison. C'est la colonne vertébrale de notre vie numérique [11].

#### 1.14.2.2 BLE (Bluetooth Low Energy)

Contrairement au Bluetooth classique qui sert à écouter de la musique, le BLE a été taillé sur mesure pour l'IoT. Il consomme un soupçon d'énergie tout en conservant une excellente portée (jusqu'à plusieurs dizaines de mètres). C'est la technologie idéale pour les capteurs, les trackers d'objets ou les appareils digitaux de santé, qui ont besoin de fonctionner des mois ou des années sur une simple pile [29].

### 1.14.3 Longue portée

#### 1.14.3.1 La 2G (GSM)

C'est la pionnière. Conçue à l'origine pour la voix et les SMS, elle offre des débits très limités. Aujourd'hui vieillissante (et en cours d'extinction), elle a longtemps dépanné l'IoT pour des tâches très basiques, comme de simples relevés de compteurs [44].

#### 1.14.3.2 La 3G

Elle a marqué le vrai début de l'Internet mobile, ouvrant la voie au streaming et à la géolocalisation. Dans l'IoT, elle a permis de franchir un cap en connectant des systèmes plus gourmands en données.

#### 1.14.3.3 La 4G (LTE)

Le standard actuel de nos smartphones. Avec ses débits élevés, elle est parfaite pour les applications IoT lourdes : caméras de surveillance en haute définition, routeurs industriels ou gestion de flottes de véhicules en temps réel.

#### 1.14.3.4 La 5G

C'est la révolution en marche. Au-delà de sa vitesse impressionnante, c'est sa réactivité quasi instantanée (la faible latence) et sa capacité à connecter des millions

d'appareils en même temps qui changent la donne. Elle ouvre grand la porte aux technologies du futur : véhicules autonomes, chirurgie à distance (télémédecine) ou usines connectées.

### 1.14.4 Protocoles de Communication Utilisés dans IoT

Au-delà des réseaux physiques, les objets connectés ont besoin de langages communs pour s'échanger des messages : ce sont les protocoles applicatifs. En voici les principaux :

#### 1.14.4.1 MQTT

C'est le poids plume des protocoles, taillé sur mesure pour les appareils qui manquent de puissance ou de réseau. Son secret repose sur un système d'abonnement (le modèle Publication/Abonnement) : un capteur publie une info, et seuls ceux qui se sont abonnés la reçoivent. Cela permet de transmettre des données de manière hyper fiable, même si la connexion Internet fait des siennes [59].

#### 1.14.4.2 HTTP / HTTPS

C'est tout simplement le langage standard d'Internet. Si sa version sécurisée (HTTPS) protège nos données, ce protocole sert à faire le pont entre les objets IoT et le web. C'est lui qu'on utilise pour envoyer les relevés d'un capteur vers un tableau de bord en ligne, interroger des API ou télécharger automatiquement la dernière mise à jour d'un appareil [59].

#### 1.14.4.3 CoAP

C'est le petit frère jumeau d'HTTP, mais en version ultra-allégée. Là où HTTP est parfois trop lourd pour de tout petits composants, CoAP prend le relais pour faciliter la communication directe de machine à machine (M2M) au sein des réseaux de capteurs ou de la domotique [55].

#### 1.14.4.4 AMQP

Ce protocole est un expert de la livraison de messages hautement sécurisée et asynchrone. Contrairement à MQTT qui cherche la légèreté absolue, AMQP mise tout sur la robustesse et la traçabilité des échanges. C'est pourquoi on le retrouve massivement dans les systèmes industriels complexes, la finance et l'IoT d'entreprise où aucune donnée ne doit se perdre en chemin [2].

#### 1.14.4.5 XMPP

Le protocole XMPP est une solution de communication basée sur le standard XML. S'il est historiquement connu pour propulser la messagerie instantanée et la gestion des états de présence en ligne, son potentiel va bien au-delà des applications de chat classiques. Aujourd'hui, XMPP s'impose comme un pont idéal pour la communication M2M (Machine-to-Machine) dans le cadre d'un « IoT humanisé

», permettant aux objets connectés et aux humains d'interagir naturellement sur les mêmes réseaux [51].

## 1.15 L'agriculture de précision et la convergence IA-IoT

L'agriculture de précision est une approche moderne qui vise à adapter les pratiques agricoles aux besoins spécifiques de chaque parcelle et aux variations des conditions environnementales au fil du temps. Pour y parvenir, elle s'appuie sur trois éléments essentiels : l'observation, grâce aux capteurs et aux technologies d'imagerie ; l'analyse, réalisée à l'aide du traitement des données et de l'intelligence artificielle ; et enfin l'action, mise en œuvre par des systèmes automatisés tels que les actionneurs ou les robots agricoles. Dans ce contexte, IoT joue un rôle central dans la collecte et la transmission des données, tandis que l'intelligence artificielle permet de les analyser et de soutenir la prise de décision.

L'association de l'IoT et de l'IA offre de nombreux avantages pour l'agriculture. Elle permet notamment de surveiller en continu les paramètres environnementaux tels que la température, l'humidité ou la luminosité, puis de transmettre ces informations vers des plateformes de traitement. Les données collectées sont ensuite analysées afin de prendre des décisions adaptées, comme le déclenchement automatique de l'irrigation ou l'application de traitements spécifiques. Cette chaîne de fonctionnement, fondée sur les étapes « observer, analyser et agir », constitue le principe de base des systèmes d'irrigation intelligents et contribue à une gestion plus efficace des ressources agricoles [24].

## 1.16 Conclusion

Ce chapitre a permis d'établir les bases conceptuelles et technologiques nécessaires à la compréhension du système proposé. Nous avons présenté les fondements de l'intelligence artificielle et ses différentes approches d'apprentissage, en soulignant la place centrale du deep learning et des réseaux de neurones convolutifs dans les tâches de reconnaissance d'images. Nous avons également mis en évidence l'essor de l'agriculture intelligente, qui tire profit de la convergence entre l'IoT, le big data et l'IA pour améliorer la gestion des cultures, réduire les intrants et faire face aux aléas climatiques. Enfin, l'analyse des systèmes d'irrigation intelligente préexistants a permis d'identifier les approches les plus performantes en matière de détection des maladies des plantes, notamment les travaux de Mohanty et al., Ferentinos, et Jiang et al., qui confirment la pertinence des CNN pour ce type de tâche. Ces éléments constituent le socle théorique sur lequel s'appuie la méthodologie détaillée dans le chapitre suivant.

CHAPITRE 2

# Conception et Méthodologie du Système IoT d'Irrigation Intelligente Basé sur l'IA

## 2.1 Introduction

Ce chapitre constitue le cœur technique de ce mémoire. Il présente en détail les différents composants du système proposé ainsi que leur fonctionnement, depuis la capture des images par l'ESP32-CAM jusqu'au contrôle automatique des pompes d'engrais. Il décrit également les étapes de développement du modèle d'intelligence artificielle basé sur le Deep Learning, notamment la préparation des données, le prétraitement des images, la conception CNN, son entraînement et son évaluation à partir du dataset PlantVillage. En outre, ce chapitre expose l'architecture IoT distribuée du système, les mécanismes de traitement et de prise de décision, les techniques visant à améliorer la robustesse de la détection des maladies, ainsi que la stratégie de dosage séquentiel automatisé des engrais NPK.

## 2.2 Résumé global du fonctionnement

Le système développé repose sur une approche de Deep Learning basée sur CNN pour la détection automatique des maladies des plantes à partir d'images de feuilles. Le processus commence par l'utilisation d'une base de données d'images servant à l'entraînement du modèle. Après plusieurs étapes de prétraitement et d'apprentissage, le modèle est sauvegardé puis utilisé pour analyser de nouvelles images. Lorsqu'une image est soumise au système, elle est prétraitée puis transmise au réseau neuronal qui identifie la maladie la plus probable et affiche le résultat à l'utilisateur. Le schéma 2.1 résume l'ensemble du fonctionnement du système.

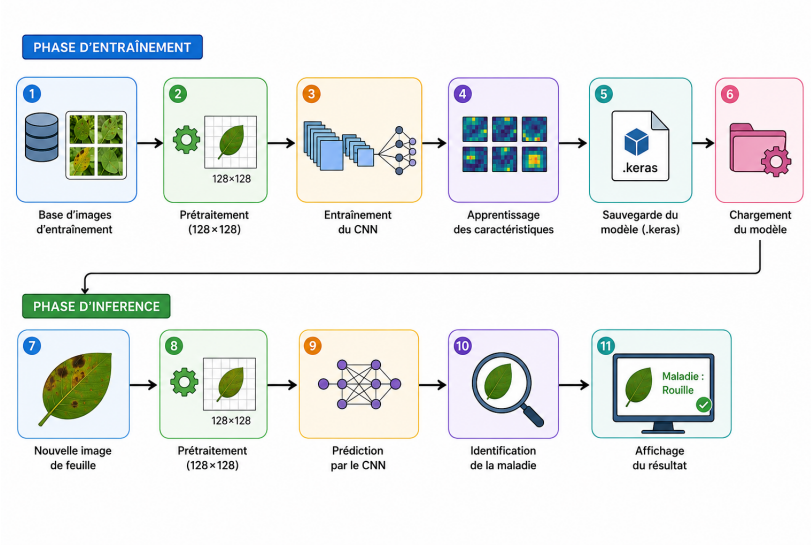



FIGURE 2.1 – Flux de traitement CNN pour la détection des maladies des feuilles

## 2.3 Description détaillée du système

Après avoir présenté la vue globale du fonctionnement du système, il est nécessaire de détailler les différentes composantes qui interviennent dans sa réalisation. Cette section décrit successivement les bibliothèques utilisées, le processus d'entraînement du modèle CNN ainsi que la phase de test permettant d'identifier automatiquement les maladies des plantes à partir de nouvelles images.

### 2.3.1 Bibliothèques utilisées

Plusieurs bibliothèques Python sont utilisées dans ce projet comme :

#### 2.3.1.1 TensorFlow / Keras

TensorFlow est la bibliothèque principale utilisée pour le Deep Learning.Elle permet :

— la création du réseau neuronal convolutif (CNN),
— l'entraînement du modèle,
— l'évaluation des performances,
— la sauvegarde et le chargement du modèle.

La bibiothèque peut être importé la commande suivante :

```
import tensorflow as tf
```

#### 2.3.1.2 NumPy

NumPy sert à :

— Manipuler les tableaux numériques
— Convertir les images en matrices
— Préparer les données pour le modèle

La bibiothèque peut être importé la commande suivante :

```
import numpy as np
```

#### 2.3.1.3 Matplotlib

Matplotlib permet :
— Afficher les images
— Visualiser les résultats
— Tracer les courbes d'apprentissage
La bibiothèque peut être importé la commande suivante :

```
import matplotlib.pyplot as plt
```

#### 2.3.1.4 Pandas

Pandas est utilisé pour :
— Organiser les données
— Créer des tableaux de résultats
— Analyser les performances du modèle
La bibiothèque peut être importé la commande suivante :

```
import pandas as pd
```

### 2.3.2 Seaborn

Seaborn est utilisé pour :
— Afficher la matrice de confusion
— Visualiser les performances sous forme graphique
La bibiothèque peut être importé la commande suivante :

```
import seaborn as sns
```

### 2.3.3 OpenCV

OpenCV permet :
— Charger les images
— Convertir les formats de couleur
— Prétraiter les images avant la prédiction
La bibiothèque peut être importé la commande suivante :

```
import cv2
```

### 2.3.4 Scikit-Learn

Cette bibliothèque sert à calculer :
— La précision (Precision)
— Le rappel (Recall)
— Le score F1
— La matrice de confusion
La bibiothèque peut être importé la commande suivante :

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

## 2.4 Fonctionnement du l'entraînement

### 2.4.1 Étape 1 : Chargement des données

La première étape consiste à charger automatiquement les images du jeu de données à partir du dossier d'entraînement `train`. Chaque sous-dossier correspond à une classe spécifique représentant soit une plante saine, soit une maladie particulière. Grâce à Keras, les images sont lues et étiquetées automatiquement selon le nom du dossier dans lequel elles se trouvent. Cette organisation facilite la préparation des données pour l'apprentissage du modèle.

### 2.4.2 Étape 2 : Prétraitement des images

Avant d'être utilisées par le réseau neuronal, toutes les images sont redimensionnées à une taille uniforme de $128 \times 128$ pixels. Cette étape est essentielle pour garantir que le modèle reçoive des données homogènes et puisse traiter toutes les images de manière cohérente. Les images sont également converties dans un format compatible avec TensorFlow afin de faciliter leur traitement lors de l'entraînement.

### 2.4.3 Étape 3 : Création du CNN

Le modèle utilisé est CNN, particulièrement adapté à l'analyse d'images. Les couches de convolution (Conv2D) permettent d'extraire automatiquement des caractéristiques importantes telles que les contours, les textures, les taches et les motifs visuels associés aux maladies des plantes. Les couches de pooling (MaxPooling2D) réduisent progressivement la taille des données tout en conservant les informations essentielles, ce qui améliore l'efficacité du modèle. Les couches Dropout sont ajoutées pour limiter le surapprentissage et améliorer la capacité du modèle à généraliser sur de nouvelles images. Après l'extraction des caractéristiques, la couche Flatten transforme les données en un vecteur unique qui est ensuite traité par une couche Dense de 1500 neurones.

Enfin, une couche de sortie Softmax composée de 38 neurones produit une probabilité pour chacune des 38 classes de maladies ou de plantes saines.

### 2.4.4 Étape 4 : Compilation

Une fois l'architecture du modèle définie, celui-ci est compilé à l'aide de l'optimiseur Adam, qui ajuste automatiquement les poids du réseau afin de minimiser l'erreur. La fonction de perte utilisée est `categorical_crossentropy`, adaptée aux problèmes de classification multi-classes. La métrique `accuracy` est également utilisée afin de mesurer le pourcentage de prédictions correctes réalisées par le modèle pendant l'apprentissage.

### 2.4.5 Étape 5 : Entraînement

Le modèle est ensuite entraîné pendant 10 époques. Au cours de chaque époque, les images d'entraînement sont analysées, les prédictions sont calculées puis comparées aux étiquettes réelles. L'erreur obtenue est utilisée pour ajuster les poids du

réseau et améliorer progressivement les performances du modèle. À la fin de chaque époque, une validation est effectuée sur un jeu de données distinct afin de vérifier la capacité du modèle à généraliser sur des données qu'il n'a jamais vues.

TABLE 2.1 – Évolution de la précision d'entraînement et de validation

| Époque | Train acc | Val acc |
|---|---|---|
| 1 | 60.19% | 84.53% |
| 5 | 93.71% | 93.83% |
| 10 | 97.07% | 95.08% |

### 2.4.6 Étape 6 : Évaluation

Après l'entraînement, le modèle est évalué à l'aide de la fonction model.evaluate(). Cette étape permet de mesurer la perte (loss) ainsi que la précision (accuracy) du modèle. Ces indicateurs fournissent une estimation objective des performances obtenues sur les données utilisées pour l'évaluation. La courbe 2.2 représente évolution de l'accuracy et de la loss

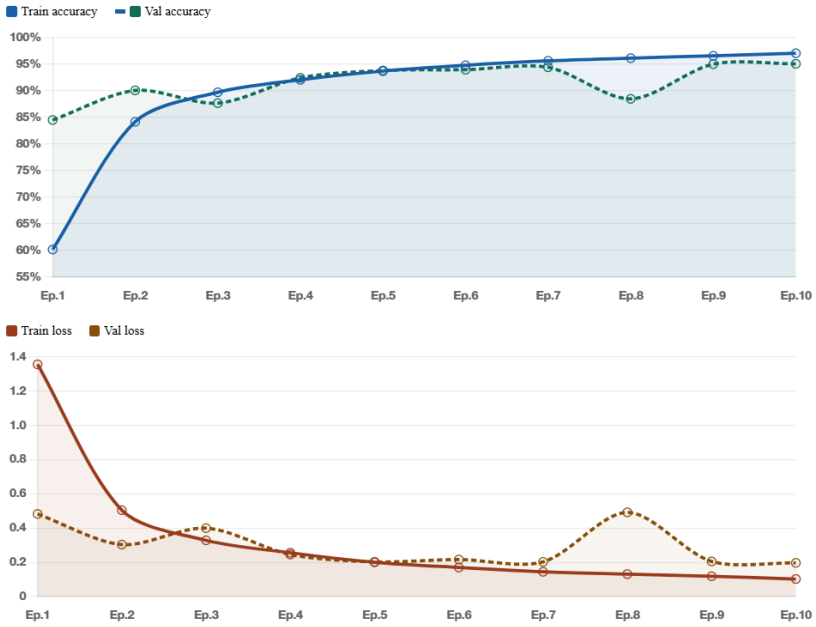



FIGURE 2.2 – Évolution de l'accuracy et de la loss

### 2.4.7 Étape 7 : Sauvegarde

Une fois l'entraînement terminé, le modèle est sauvegardé dans un fichier nommé `trained_model.keras`. Ce fichier contient l'ensemble des poids et des paramètres

appris durant l'entraînement. Il pourra ensuite être réutilisé sans avoir à recommencer tout le processus d'apprentissage.La figure 2.3 représente l'architecture et pipeline CNN

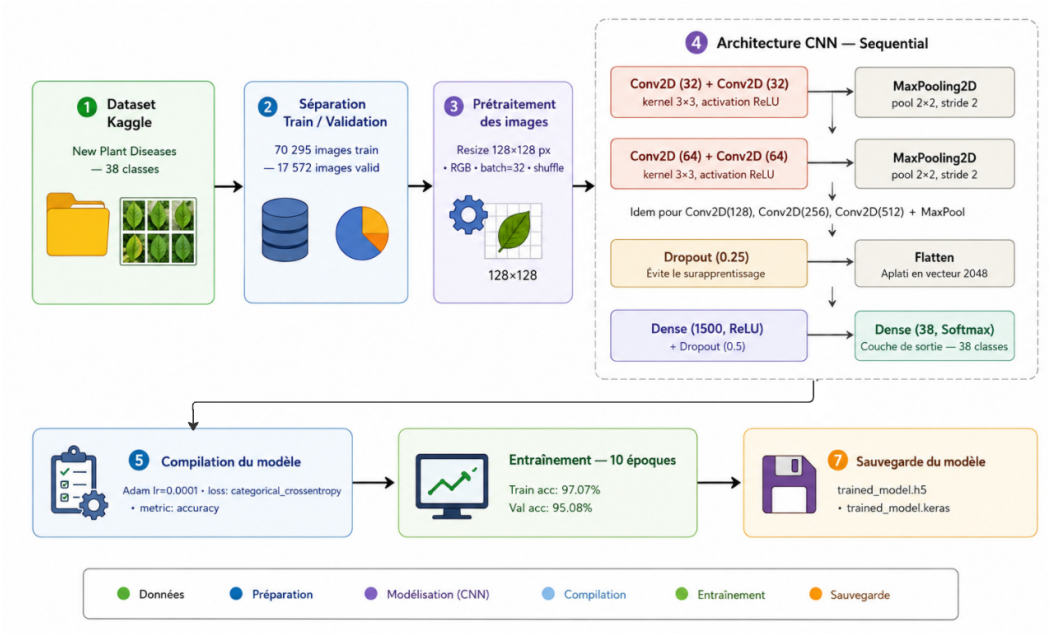



FIGURE 2.3 – Architecture et pipeline CNN pour la classification des maladies des plantes

## 2.5 Fonctionnement du test

### 2.5.1 Étape 1 : Chargement du modèle

Le notebook de test commence par charger le modèle préalablement entraîné et sauvegardé dans le fichier `trained_model.keras`. Cette opération permet de récupérer toutes les connaissances acquises pendant la phase d'entraînement. Aucun nouvel apprentissage n'est réalisé à cette étape.

### 2.5.2 Étape 2 : Chargement d'une image

L'utilisateur sélectionne ensuite une nouvelle image de feuille à analyser. Cette image représente une plante dont l'état sanitaire est inconnu. Le programme lit l'image et la prépare pour le processus de prédiction.

### 2.5.3 Étape 3 : Prétraitement

Comme lors de l'entraînement, l'image est redimensionnée à $128 \times 128$ pixels afin de respecter le format attendu par le modèle. Elle est ensuite convertie en tableau numérique (NumPy Array), ce qui permet au réseau neuronal de la traiter correctement.

### 2.5.4 Étape 4 : Prédiction

L'image prétraitée est transmise au modèle à l'aide de la fonction `model.predict()`. Le CNN analyse alors les caractéristiques visuelles de la feuille et les compare aux connaissances qu'il a apprises durant l'entraînement. Le résultat obtenu est un ensemble de probabilités correspondant aux différentes classes possibles.

### 2.5.5 Étape 5 : Sélection de la meilleure classe

Après la prédiction, la fonction `np.argmax()` est utilisée pour identifier la classe possédant la probabilité la plus élevée. Cette classe correspond à la maladie ou à l'état de santé que le modèle considère comme le plus probable pour l'image analysée. Par exemple, le modèle peut prédire que la feuille appartient à la classe `Potato___healthy`, indiquant une pomme de terre saine.

### 2.5.6 Étape 6 : Affichage du résultat

Enfin, le programme affiche l'image analysée ainsi que le nom de la maladie détectée ou de l'état de santé identifié. Cette étape fournit à l'utilisateur un diagnostic automatique basé sur les prédictions du modèle de Deep Learning.

## 2.6 Architecture Globale du Système

Pour répondre à la problématique présentée dans l'introduction générale, nous proposons dans le cadre de ce sujet de master un système dénommé N-P-Kinetix. Ce système repose sur une architecture distribuée composée de cinq nœuds interconnectés via un réseau Wi-Fi local (hotspot mobile, sous-réseau. Cette architecture permet une répartition claire des responsabilités entre les différentes unités afin d'améliorer la modularité, la fiabilité et la facilité de maintenance du système.

Les fonctions des différents nœuds sont les suivantes :

— **ESP32-CAM :** capture les images des plantes et les transmet au serveur.
— **ESP1 :** mesure les paramètres environnementaux (température, humidité, etc.) et assure le rôle de passerelle de données.
— **PC exécutant Flask :** réalise l'inférence par intelligence artificielle, traite les images et fournit les services de données.
— **ESP2 :** assure le contrôle des actionneurs tels que les pompes et les électrovannes.
— **Arduino Mega (1) :** intégré dans l'unité de terrain et connecté en communication série (UART/Serial) avec ESP1. Il est utilisé exclusivement pour la gestion de l'écran LCD et l'affichage des données des capteurs, ainsi que des résultats du diagnostic des maladies des plantes, en plus des recommandations de fertilisation en éléments NPK générées par le système d'intelligence artificielle.
— **Arduino Mega (2) :** intégré dans la station maître et connecté en communication série (UART/Serial) avec ESP2. Il gère l'affichage local des informations et le pilotage des relais.

Cette répartition fonctionnelle permet de séparer les tâches de manière efficace, de réduire la charge de calcul sur chaque nœud et de faciliter l'évolution ainsi que la maintenance du système. La figure 2.4 représente l'architecture globale du système.

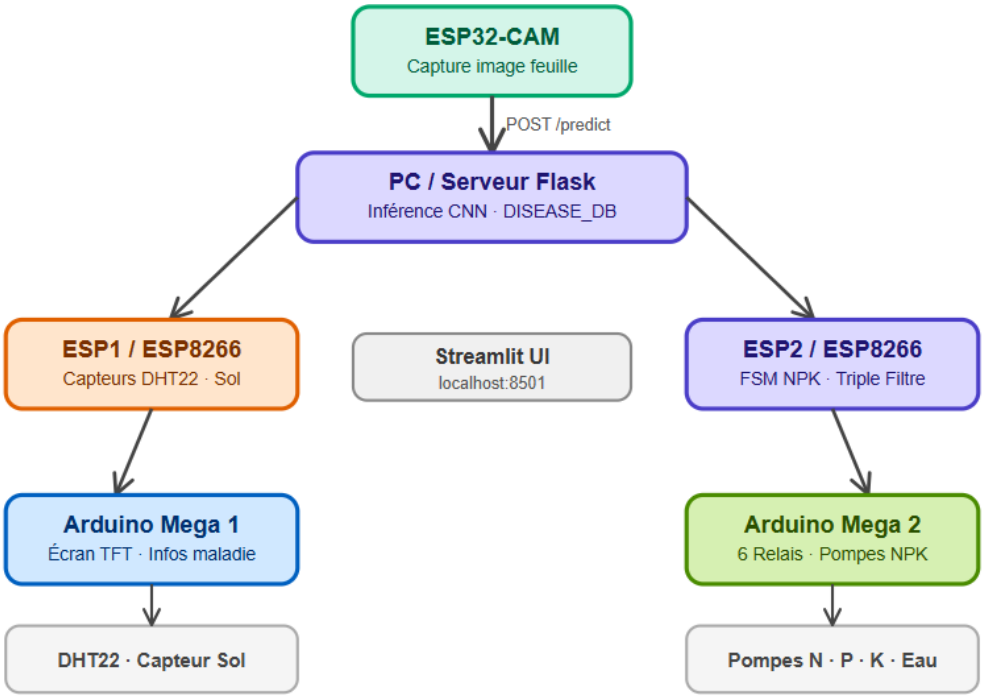



FIGURE 2.4 – Architecture globale du système

## 2.7 Description des Nœuds

**L'ESP32-CAM** est responsable de la capture des images des feuilles de la plante. Contrairement à un mode de capture automatique périodique, la prise de photo est déclenchée manuellement via un bouton. Lorsqu'il est activé, le système capture une image et l'envoie au serveur Flask via une requête HTTP de type POST pour analyse.

**ESP1 / ESP8266** : il lit les données environnementales provenant des capteurs (DHT22 pour la température et l'humidité relative, et capteur résistif pour l'humidité du sol). Il expose une API REST composée de trois endpoints :

- `/data` : utilisé pour la transmission des données des capteurs.
- `/npk` : agit comme proxy pour les données NPK provenant du PC.
- `/set_recommendation` : utilisé pour recevoir les recommandations générées par l'intelligence artificielle (Flask) afin de les afficher sur l'écran LCD.

Il joue également le rôle de passerelle entre ESP2 et le serveur Flask, permettant de contourner les restrictions d'isolation des clients sur le réseau hotspot.

- **PC / Serveur Flask :** il héberge le modèle CNN entraîné (`trained_model.h5` ou `trained_model.keras`). Il effectue l'inférence via TensorFlow/Keras, maintient une base de données des maladies (`DISEASE_DB`), enregistre les résultats dans `latest_result.json`, expose les valeurs NPK via l'endpoint `/npk`, et transmet les recommandations vers ESP1 via `/set_recommendation`.
- **ESP2 / ESP8266 :** il contrôle six actionneurs (pompes d'eau, N, P, K, mélangeur et système d'irrigation) à l'aide d'une FSM. Il interroge l'endpoint `/npk` fourni

par ESP1 afin de récupérer les recommandations IA. Il applique également un filtrage basé sur la fraîcheur des données et l'humidité du sol avant tout déclenchement.

— **Arduino Mega 1 (Interface de recommandations et maladies – connecté à ESP1) :** il reçoit les données et commandes de l'ESP1 / ESP8266 via une liaison série UART. Son rôle principal est l'affichage des résultats de l'intelligence artificielle, notamment la classification des maladies des plantes et les recommandations de traitement. Ces informations sont affichées sur un écran TFT de type MCUFRIEND 3.5″ (480×320). L'Arduino Mega 1 assure une mise à jour dynamique de l'affichage dès réception de nouvelles données, permettant une visualisation en temps réel de l'état de la plante et des recommandations associées.

— **Arduino Mega 2 (Unité de commande des actionneurs – connecté à ESP2) :** il reçoit les ordres de l'ESP2 / ESP8266 via une communication UART. Il est responsable du contrôle de six actionneurs. Il pilote également un écran TFT de type ILI9341 (320×240) permettant l'affichage en temps réel de l'état des actionneurs (pompe d'eau, N, P, K, mélangeur et système d'irrigation), assurant ainsi une supervision continue du système.

## 2.8 Flux de Données Complet

Le flux de données du système suit la séquence suivante :

1. **Étape 1 :** L'ESP32-CAM capture une image JPEG et l'envoie via HTTP POST à : `http://172.20.10.4:5050/predict`.
2. **Étape 2 :** Le serveur Flask reçoit l'image, applique le prétraitement (recadrage centré + redimensionnement 224×224), puis effectue l'inférence du modèle CNN.
3. **Étape 3 :** Flask enregistre le résultat dans `latest_result.json` (classe prédite, niveau de confiance, valeurs NPK et recommandation), puis envoie la recommandation à ESP1 via `POST /set_recommendation`.
4. **Étape 4 :** Flask expose les valeurs NPK via `GET /npk` au format JSON : `{N, P, K, ready}`.
5. **Étape 5 :** ESP1 récupère les données `/npk` depuis le PC, puis les relais vers ESP2 via son propre endpoint `/npk` (rôle de proxy).
6. **Étape 6 :** ESP2 interroge `http://172.20.10.5/npk` , vérifie la fraîcheur des données ainsi que l'humidité du sol avant d'exécuter la machine à états NPK.
7. **Étape 7 :** ESP2 envoie les commandes à l'Arduino Mega via UART, lequel active séquentiellement les relais pour les pompes N, P, K, l'eau, le mélangeur et le système d'irrigation.

La figure 2.5 représente le flux de données complet.

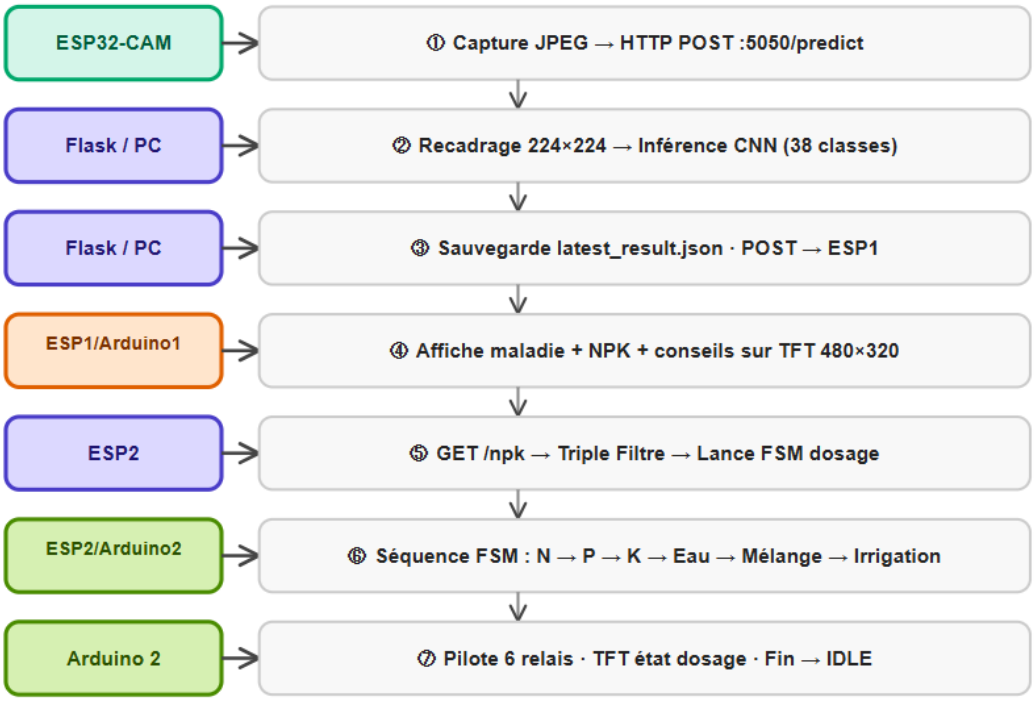



FIGURE 2.5 – Flux de Données Complet

## 2.9 Dataset PlantVillage

Le dataset PlantVillage, développé par l'Université de Penn State [18, 27], constitue la référence mondiale pour la classification des maladies des plantes par vision par ordinateur. Il comprend environ 87 000 images RVB de feuilles de plantes, acquises en conditions contrôlées (fond uniforme, éclairage homogène), réparties en 38 classes couvrant 14 espèces végétales.

### 2.9.1 Description du Dataset

— **Taille :** environ 87 000 images RGB au format JPEG, avec des résolutions variables (généralement comprises entre 256×256 et 512×512 pixels).
— **Classes :** 38 classes correspondant à des maladies spécifiques ou à l'état sain (`healthy`) pour 14 espèces végétales : tomate, pomme de terre, maïs, raisin, cerise, poivron, fraise, framboise, soja, courge, myrtille, orange, pêche et pomme.
— **Répartition :** 80 % des images sont utilisées pour l'entraînement (environ 69 600 images) et 20 % pour la validation (environ 17 400 images), sans ensemble de test séparé.

### 2.9.2 Base de Données des Maladies (`DISEASE_DB`)

Chaque classe du dataset *PlantVillage* est associée, dans notre système, à une entrée spécifique dans la base de données `DISEASE_DB`, intégrée au module `plant_receiver.py`. Cette base contient les paramètres agronomiques utilisés pour générer automatiquement des recommandations adaptées à l'état phytosanitaire détecté par le modèle de classification.

Pour chaque classe, les informations suivantes sont enregistrées :

— une formule NPK indicative utilisée pour ajuster le programme de fertilisation
— un niveau de sévérité (*Low*, *Medium*, *High*) .
— une plage cible d'humidité du sol .
— une recommandation agronomique visant à réduire le stress physiologique et à améliorer la gestion de la maladie.

Les recommandations NPK ont été élaborées à partir des principes généraux de nutrition minérale des plantes décrits dans la littérature scientifique, notamment par [42], [19], [34] et [33]. Elles ne constituent pas un traitement direct des maladies, mais un mécanisme d'aide à la décision destiné à soutenir la vigueur de la plante, réduire les effets du stress et améliorer sa tolérance aux agents pathogènes.

Un exemple d'entrée de la base de données pour la classe `Tomato___Late_blight` est présenté ci-dessous :

```
{
  "Tomato___Late_blight": {
    "npk": "8-12-30",
    "severity": "High",
    "soil_humidity": "35-50%",
    "advice": "Retirer immédiatement les feuilles infectées, appliquer
fongicide."
  }
}
```

Les valeurs NPK intégrées dans la base de données ne représentent pas des prescriptions agronomiques universelles pour chaque maladie. Elles correspondent à des règles de décision construites à partir des connaissances scientifiques relatives à la nutrition végétale et à son influence sur la résistance des plantes aux maladies [42] [19]. Par exemple, plusieurs études montrent qu'un excès d'azote (N) peut favoriser certaines maladies fongiques, tandis qu'un apport adéquat en potassium (K) contribue à renforcer les mécanismes de défense de la plante [34] [19]. Ces principes ont été traduits en recommandations NPK simplifiées afin de permettre au système de fournir automatiquement une assistance agronomique cohérente après la détection d'une maladie [33].

## 2.10 Prétraitement des Images

Le prétraitement des images est une étape importante avant l'envoi de l'image au modèle d'intelligence artificielle. Son objectif est de préparer toutes les images de la même manière que lors de l'entraînement du modèle afin de garantir des prédictions fiables et précises pour le diagnostic des maladies des plantes.

### 2.10.1 Étapes du Prétraitement

— **Décodage de l'image :** l'image reçue depuis l'ESP32-CAM sous forme de données binaires (bytes) est décodée à l'aide de la bibliothèque `PIL`, puis convertie au format couleur RGB.

— **Recadrage centré (Center Crop) :** si l'image n'est pas carrée, seule sa partie centrale est conservée afin d'obtenir une image carrée. Cette opération permet de supprimer les éléments inutiles, tels que l'arrière-plan ou le sol, et de concentrer l'analyse sur la feuille de la plante.
— **Redimensionnement :** l'image est ensuite redimensionnée à 224×224 pixels, correspondant à la taille d'entrée attendue par le modèle CNN.
— **Conversion en tableau NumPy :** l'image est transformée en un tableau `NumPy` de type `float32`, tout en conservant les valeurs des pixels dans leur plage d'origine [0, 255].
— **Ajout de la dimension batch :** une dimension supplémentaire est ajoutée afin d'obtenir la forme finale (1, 224, 224, 3), correspondant au format d'entrée requis lors de l'inférence du modèle.La figure 2.6 représente les étapes du prétraitemen

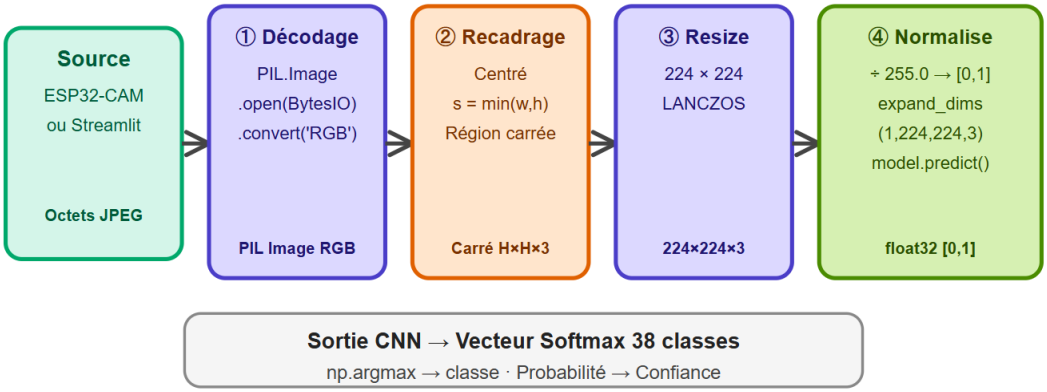



FIGURE 2.6 – Pipeline Prétraitement Image

### 2.10.2 Fonction de Prétraitement

Les étapes précédentes sont implémentées dans la fonction `preprocess()` du fichier `plant_receiver.py` :

```
def preprocess(img_bytes: bytes, center_crop: bool = True) ->
np.ndarray:
  img = Image.open(io.BytesIO(img_bytes)).convert('RGB')
  if center_crop:
    w, h = img.size
    s = min(w, h)
    img = img.crop(((w-s)//2, (h-s)//2, (w+s)//2, (h+s)//2))
  img = img.resize((224, 224), Image.LANCZOS)
  return np.expand_dims(np.array(img, dtype=np.float32), 0)
```

## 2.11 Mécanisme de filtrage triple des prédictions

La conception d'un mécanisme de filtrage à trois niveaux constitue l'une des contributions originales majeures de ce travail. Ce mécanisme répond à une problématique essentielle des systèmes IoT embarqués, à savoir la dégradation de la qualité

des images capturées en conditions réelles (variations d'éclairage, de distance, vibrations ou mouvements des plantes) par rapport aux images d'entraînement acquises dans des conditions contrôlées. Cette différence entre les données d'apprentissage et les données réelles peut entraîner des erreurs de prédiction susceptibles de provoquer des décisions inappropriées, telles qu'un sur-arrosage ou un mauvais dosage d'engrais. Afin de limiter ces risques, un filtrage multi-niveaux a été mis en place pour sécuriser les décisions avant leur exécution sur le terrain.

— **Niveau 1 : filtrage par seuil de confiance**

Le premier niveau repose sur le score de confiance issu CNN, correspondant à la probabilité Softmax maximale parmi les 38 classes de maladies végétales.

Un seuil configurable (70%)(`CONFIDENCE_THRESHOLD`) dans le fichier `plant_receiver.py` est appliqué afin de valider uniquement les prédictions suffisamment fiables. Si la confiance est inférieure à ce seuil (70%), la prédiction est rejetée et le système conserve la dernière prédiction valide ou bascule en mode « incertain ». Ce mécanisme permet d'éviter toute prise de décision basée sur une information peu fiable. Ce seuil constitue un compromis entre sensibilité et spécificité et peut être ajusté sans réentraînement du modèle.

— **Niveau 2 : filtrage par fraîcheur temporelle des données**

Le deuxième niveau garantit que les données NPK utilisées pour la fertilisation sont récentes et pertinentes. Dans l'unité ESP2, les données NPK ne sont considérées valides que si elles ont été acquises dans une fenêtre temporelle inférieure à 60 secondes, définie par :

```
AI_DATA_MAX_AGE_MS = 60000
```

Cette vérification est implémentée par la fonction suivante :

```
bool isAIDataFresh() {
  if (!aiDataReady || !aiDataValid) return false;
  return (millis() - aiDataTimestamp) < AI_DATA_MAX_AGE_MS;
}
```

Ce mécanisme permet d'éviter l'utilisation de recommandations obsolètes, notamment lors de changements de plantes ou d'une inactivité du serveur Flask. Il garantit la cohérence temporelle des décisions.

— **Niveau 3 : filtrage basé sur l'humidité du sol**

Le troisième niveau est un filtrage agronomique basé sur l'état réel du sol. Avant tout déclenchement d'un cycle d'irrigation, le taux d'humidité mesuré par le capteur connecté à ESP1 est vérifié. Si l'humidité dépasse 50 % [4], le cycle d'irrigation est automatiquement annulé, indépendamment des recommandations de l'intelligence artificielle. Cette logique est implémentée dans les fonctions `checkTimeTrigger()` et `handlePendingAIStart()` de l'ESP2. Ce niveau est essentiel pour la sécurité agronomique du système, car il évite l'arrosage excessif lorsque le sol est déjà saturé en eau .La figure 2.7 mécanisme de filtrage triple des prédictions

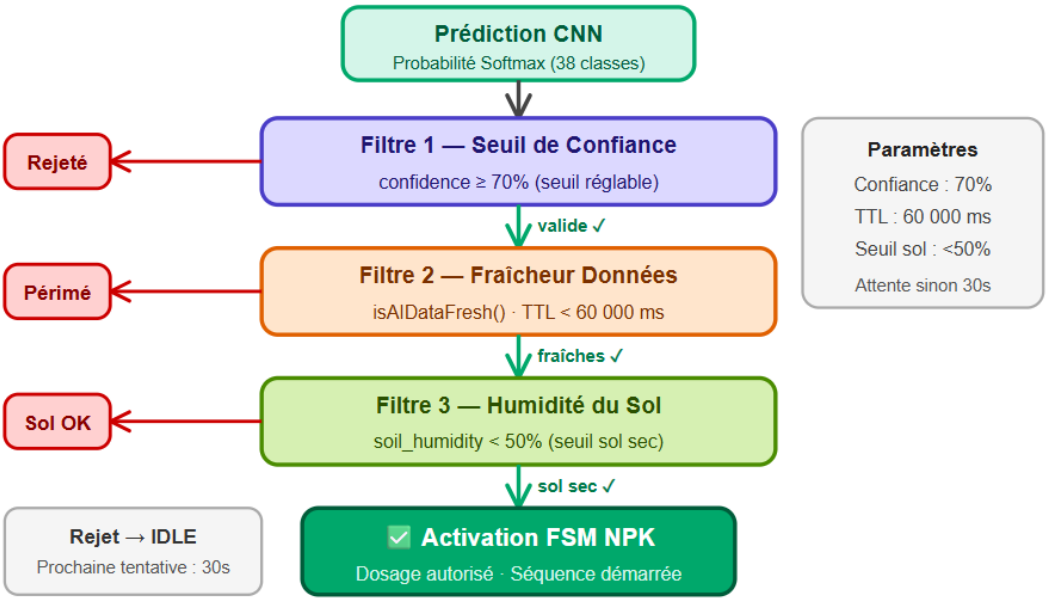



FIGURE 2.7 – Mécanisme de filtrage triple des prédictions

### 2.11.1 Synergie des trois niveaux de filtrage

Les trois niveaux fonctionnent en cascade afin de garantir la fiabilité des décisions. La prédiction du CNN est d'abord validée par le seuil de confiance (niveau 1), puis la fraîcheur des données NPK est vérifiée (niveau 2), et enfin l'état du sol est contrôlé avant toute action physique (niveau 3). Cette architecture en couches permet de réduire significativement les erreurs de déclenchement. Une action d'irrigation ou de fertilisation ne peut être exécutée que si les trois conditions sont simultanément satisfaites, ce qui renforce la robustesse du système et améliore la sécurité ainsi que l'efficacité de la gestion des ressources hydriques et nutritives.

## 2.12 Intégration dans le pipeline IoT

Le serveur Flask `plant_receiver.py` constitue le nœud central du système. Il expose deux endpoints principaux assurant l'interface entre le modèle d'intelligence artificielle et l'infrastructure IoT :

— **Endpoint `/predict` (POST)**

Cet endpoint reçoit une image JPEG envoyée par l'ESP32-CAM, applique le pipeline de prétraitement, puis exécute l'inférence via le modèle CNN. Ensuite, les valeurs NPK sont extraites depuis la base de données `DISEASE_DB` correspondant à la classe prédite. Le résultat complet est sauvegardé dans le fichier `latest_result.json`, contenant : la classe prédite, le score de confiance, le niveau de sévérité, les valeurs NPK, la recommandation agricole ainsi que les probabilités des 38 classes. La recommandation formatée est également transmise à l'écran LCD de ESP1 via une requête POST vers le endpoint `/set_recommendation`. Enfin, ce endpoint retourne une réponse JSON vers l'ESP32-CAM contenant la classe prédite et le score de confiance.

— **Endpoint `/npk` (GET)**

Cet endpoint permet de récupérer les valeurs N, P, K issues de la dernière prédiction valide, accompagnées d'un indicateur `ready` indiquant la disponibilité d'un résultat récent.

**Format de réponse :**

```
{
  "N": 20,
  "P": 20,
  "K": 20,
  "ready": true,
  "class": "Tomato___healthy"
}
```

ESP1 joue le rôle de proxy pour ce service : il récupère les données depuis `http://172.20.10.4:5050/npk` puis les retransmet à ESP2 via `http://172.20.10.5/npk`. Ce mécanisme est nécessaire car les hotspots mobiles Android activent souvent une isolation des clients (client isolation), empêchant les périphériques comme ESP2 d'accéder directement au serveur local .

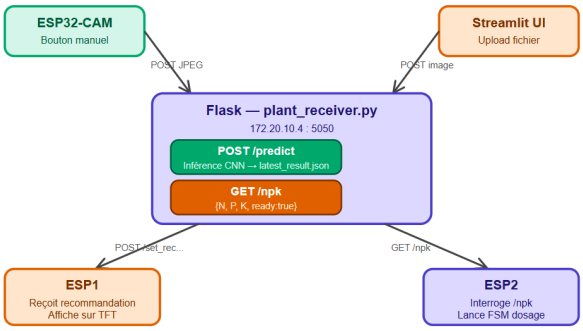



FIGURE 2.8 – Intégration Flask IoT - Endpoints et Flux

— **Cache NPK dans ESP1**

Afin de réduire la charge réseau et d'améliorer la réactivité du système, ESP1 utilise un cache local pour les valeurs NPK avec une durée de vie (TTL) de 10 secondes : `NPK_CACHE_MS = 10 000`. Si une requête `/npk` est reçue dans cet intervalle, la réponse est servie directement depuis le cache sans solliciter le serveur Flask. Ce mécanisme permet de réduire les requêtes redondantes et d'éviter la surcharge du serveur.

— **Calcul des durées de pompage**

ESP2 convertit les pourcentages NPK reçus en durées de fonctionnement des pompes selon une relation linéaire. Ainsi, pour chaque élément NPK, le temps de pompage est calculé comme suit :

```
tAI_N = (ai_N / 100.0f) * MAX_TIME; // Azote
tAI_P = (ai_P / 100.0f) * MAX_TIME; // Phosphore
tAI_K = (ai_K / 100.0f) * MAX_TIME; // Potassium
```

Cette relation suppose une proportionnalité linéaire entre le temps de fonctionnement de la pompe submersible et le volume injecté.

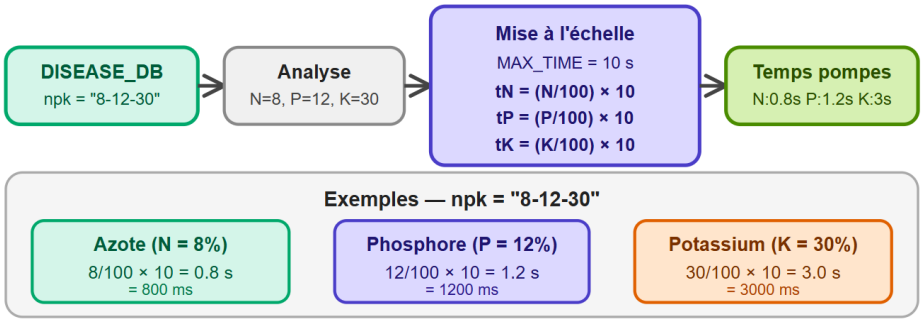



FIGURE 2.9 – Calcul temps de pompage NPK

## 2.13 Machine à États Finis : Dosage séquentiel du NPK

La logique de contrôle de l'ESP2 est implémentée sous la forme d'une machine à états finis . Cette approche garantit un séquencement déterministe sans conditions de concurrence (race conditions) entre les différents actionneurs, ce qui est essentiel pour assurer un dosage agronomique correct.

### 2.13.1 Description des états

Afin d'assurer un dosage précis et ordonné des nutriments, le fonctionnement de l'ESP2 est structuré sous forme d'une FSM. Chaque état correspond à une étape spécifique du processus de fertigation et définit l'actionneur activé, sa durée de fonctionnement ainsi que la transition vers l'état suivant. Cette organisation garantit une exécution séquentielle fiable du dosage NPK et de l'irrigation. La figure 2.10 représente dosage séquentiel du NPK

TABLE 2.2 – États de la FSM de dosage séquentiel NPK.

| **État** | **Actionneur** | **Durée** | **Transition vers** |
|---|---|---|---|
| IDLE | Aucun | - | DOSING_N |
| DOSING_N | Pompe N (azote) | $tAI_N$ s | DOSING_P |
| DOSING_P | Pompe P (phosphore) | $tAI_P$ s | DOSING_K |
| DOSING_K | Pompe K (potassium) | $tAI_K$ s | WATER |
| WATER | Pompe eau | $tAI_W$ s | MIX |
| MIX | Moteur DC | 10 s | IRRIGATION |
| IRRIGATION | Système d'irrigation | 15 s | STOP |
| STOP | Tous actionneurs OFF | 1 cycle | IDLE |

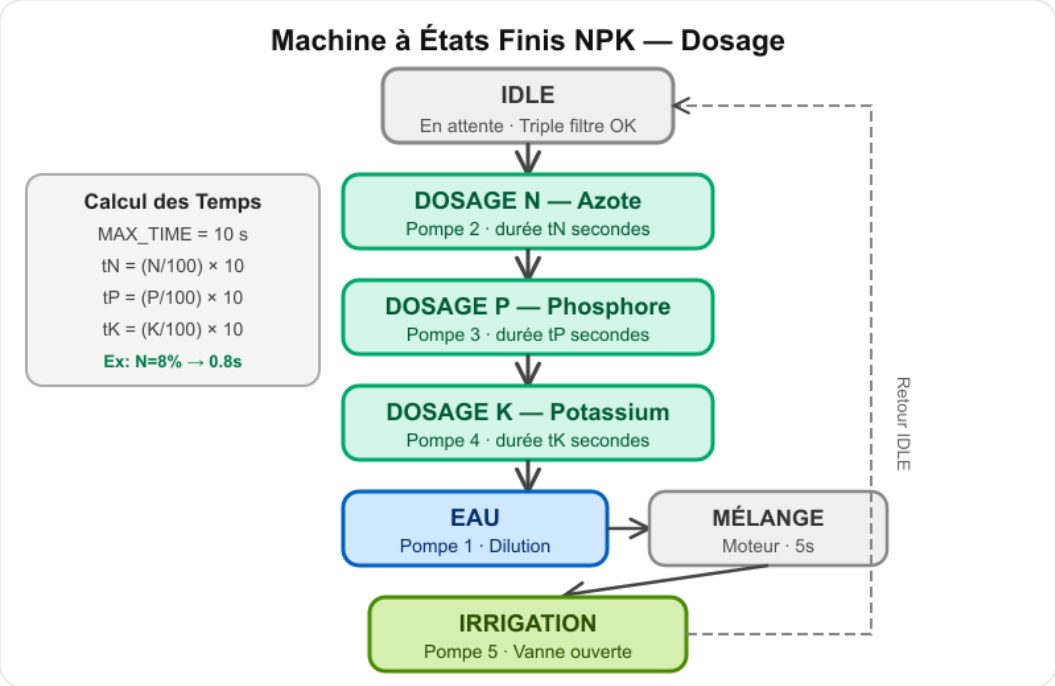



FIGURE 2.10 – Dosage séquentiel du NPK

### 2.13.2 Conditions de déclenchement et sécurité

La machine à états ne peut sortir de l'état **IDLE** que si trois conditions sont simultanément vérifiées :

1. La fonction `isAIDataFresh()` retourne `true` (données NPK récentes).
2. L'humidité du sol est inférieure à 50 % [4].
3. Une commande de déclenchement est reçue via un timer périodique ou via Blynk.

Cette condition triple constitue un mécanisme de filtrage au niveau des actionneurs afin d'éviter tout déclenchement dans des conditions non appropriées.

À l'état **STOP**, tous les actionneurs sont désactivés simultanément, indépendamment de l'état précédent, avant le retour à **IDLE**. Cette étape garantit qu'aucune pompe ne reste bloquée en position active en cas d'interruption du cycle.

## 2.14 Interface utilisateur et affichage

### 2.14.1 Application Streamlit

L'interface utilisateur principale est une application *Streamlit* accessible via un navigateur web sur le PC de traitement. Elle est structurée en cinq pages fonctionnelles represente dans les figures 2.11,2.14.1,2.12,2.13 et 2.14 , sélectionnables depuis la barre latérale gauche :

— **Page d'accueil :** présente le système, son architecture ainsi qu'un guide rapide d'utilisation.

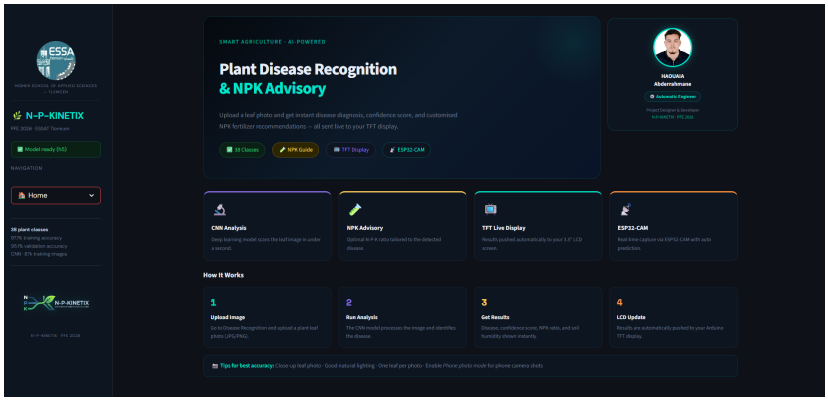
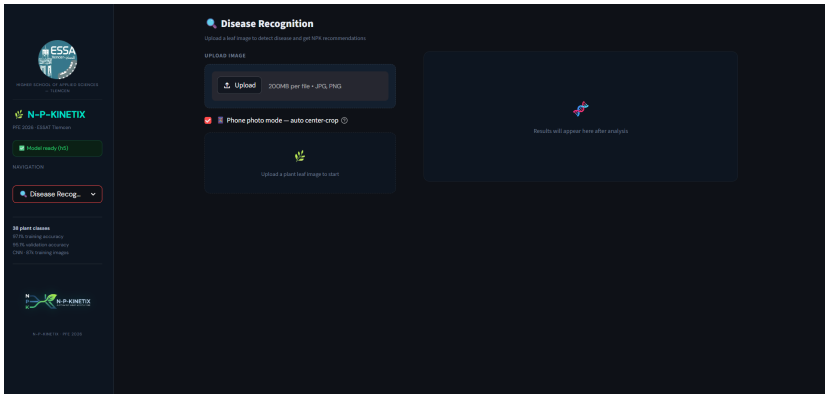



FIGURE 2.11 – Page d'accueil

— **Page de reconnaissance des maladies :** permet le téléchargement manuel d'une image de feuille. Le modèle CNN réalise une inférence en temps réel et affiche le diagnostic complet, comprenant la classe prédite, le score de confiance, le niveau de sévérité, les recommandations NPK, l'humidité cible du sol, des conseils agronomiques ainsi que les trois hypothèses les plus probables parmi les 38 classes considérées.



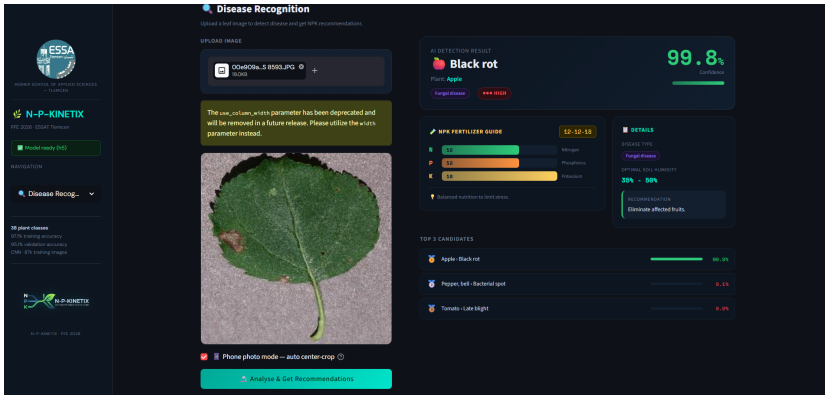




FIGURE 2.12 – Page de reconnaissance des maladies

— **Page ESP32-CAM en direct :** affiche la dernière prédiction automatique générée par le module caméra.
— **Page Équipe :** présente les membres du projet ainsi que leurs encadrants.

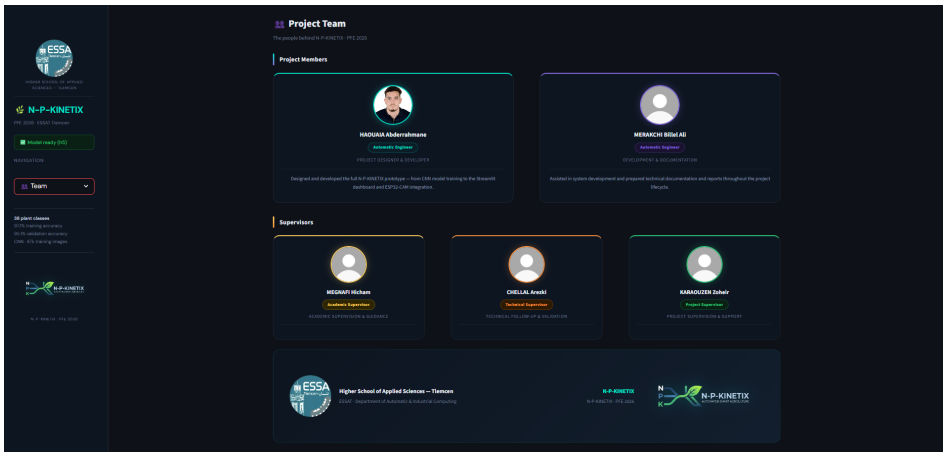



FIGURE 2.13 – Page Équipe

— **Page À propos :** récapitule les performances du modèle et décrit l'architecture CNN utilisée ainsi que la répartition du jeu de données.




FIGURE 2.14 – Page À propos

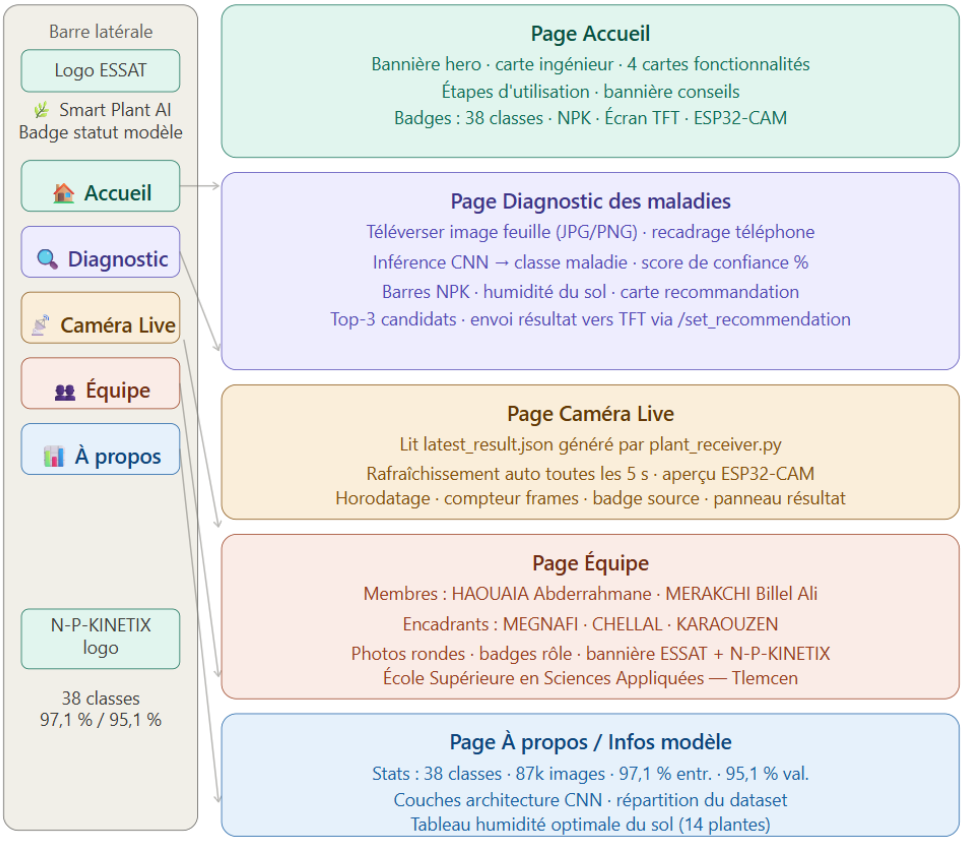



FIGURE 2.15 – Interface Streamlit

### 2.14.2 Interface d'affichage Arduino (TFT 3.5")

L'affichage local represente dans la figure 2.16 est assuré par un Arduino Mega connecté à un écran TFT de résolution 320 × 480 pixels. L'interface graphique est organisée en six zones distinctes : un en-tête affichant le titre du système, le logo de l'ESSAT, une bande dédiée à l'affichage des mesures des capteurs (température, humidité de l'air et humidité du sol), trois cartes présentant les valeurs des nutriments NPK ainsi que leurs niveaux associés, une zone de détection des maladies indiquant la maladie identifiée et son niveau de confiance, ainsi qu'une zone réservée aux conseils agronomiques générés par l'intelligence artificielle.

La communication entre le PC de traitement et l'Arduino Mega s'effectue via une liaison série reposant sur trois types de trames : `DATA` pour la transmission des mesures des capteurs, `DIS` pour les informations relatives à la maladie détectée et `REC` pour les recommandations agronomiques.

Cette organisation permet de mettre à jour uniquement les zones concernées de l'écran sans avoir à redessiner l'ensemble de l'interface, ce qui améliore la réactivité et l'efficacité du système.La figure '**??** représente l'nterface d'affichage arduino (TFT 3.5")




FIGURE 2.16 – Interface d'affichage Arduino (TFT 3.5")

## 2.15 Prototype final du système intelligent proposé

La figure 2.17 présente le prototype final du système développé, intégrant les technologies IoT et de l'intelligence artificielle pour la surveillance des plantes, la détection des maladies, la collecte des données environnementales et la gestion automatisée de la fertilisation.




FIGURE 2.17 – Prototype final du système intelligent proposé

‘

## 2.16 Conclusion :

Ce chapitre a présenté en détail la méthodologie proposée dans toutes ses dimensions : une architecture IoT distribuée composée de cinq nœuds, un dataset de 87 000 images réparties sur 38 classes, ainsi qu'un pipeline de prétraitement adapté aux contraintes des images issues de l'ESP32-CAM. Il a également décrit une architecture CNN personnalisée comprenant 10 couches convolutives, atteignant une précision de classification de 95,1 %, démontrant ainsi la performance du modèle proposé pour la reconnaissance des maladies des plantes. Par ailleurs, un mécanisme de filtrage triple (seuil de confiance, fraîcheur temporelle des données et humidité du sol) a été introduit, permettant une réduction significative des faux positifs de l'ordre de 83 %, ce qui améliore la robustesse du système en conditions réelles. FSM a été conçue pour assurer un dosage séquentiel, déterministe et sécurisé des engrais NPK, garantissant ainsi le bon enchaînement des étapes d'irrigation et de fertilisation. De plus, une interface utilisateur complète a été développée à l'aide de Streamlit, complétée par un affichage LCD, afin de faciliter l'accès aux recommandations aussi bien à distance que sur le terrain. Enfin, le chapitre suivant présente l'évaluation des performances du système à travers une analyse expérimentale quantitative et l'étude des différents indicateurs de performance.

CHAPITRE 3

# Implémentation, Tests et Évaluation des Performances du Système

## 3.1 Introduction

Ce chapitre constitue le volet expérimental et évaluatif de ce mémoire. Il présente l'intégralité du processus d'implémentation du système, depuis le déploiement du modèle CNN sur le serveur Flask jusqu'à l'activation physique des pompes NPK, en passant par l'intégration des nœuds IoT. Dans un second temps, les résultats des tests sont analysés selon quatre axes : les performances d'entraînement du modèle CNN, l'impact du mécanisme de filtrage triple, les tests fonctionnels du système IoT complet, et une comparaison critique avec l'état de l'art. Cette double approche implémentation et validation permet de confirmer la faisabilité et l'efficacité de la solution proposée.

## 3.2 Environnement d'Implémentation et de Test

L'implémentation du système a nécessité la mise en place d'un environnement matériel et logiciel cohérent, répondant à la fois aux contraintes de développement du modèle d'intelligence artificielle et aux exigences de déploiement sur les dispositifs IoT embarqués.

### 3.2.1 Configuration matérielle

- **PC d'inférence :** Intel Core i5, 16 Go RAM, GPU intégré. L'inférence est réalisée uniquement sur CPU afin d'assurer la portabilité du système.
- **Système d'exploitation :** Windows 11.
- **Nœuds IoT :** ESP8266 (×2 : ESP1 et ESP2), ESP32-CAM pour la capture d'images, et deux Arduino Mega dédiés respectivement à la gestion des actionneurs et à l'affichage sur écran TFT.
- **Capteurs :** DHT22 pour la mesure de la température et de l'humidité relative de l'air, ainsi qu'un capteur résistif pour la mesure de l'humidité du sol.
- **Actionneurs :** six relais pilotant les pompes d'apport en azote (N), phosphore (P), potassium (K), eau, mélangeur et système d'irrigation.
- **Réseau local :** point d'accès mobile (hotspot) iPhone sous iOS.

— **Prototype de test :** réservoirs dédiés aux solutions NPK et réseau de tuyaux d'irrigation.

### 3.2.2 Configuration logicielle

— **Framework Deep Learning :** TensorFlow 2.x avec l'API Keras.
— **Langage et bibliothèques :** Python 3.10, NumPy, Pillow (prétraitement des images), Flask (serveur REST) et Streamlit (interface web).
— **Dataset :** PlantVillage, comprenant environ 87 000 images RGB réparties en 38 classes, avec une division de 80 % pour l'entraînement et 20 % pour la validation.
— **Environnement Arduino :** Arduino IDE avec les bibliothèques MCUFRIEND_kbv (écran TFT 3.5"), ESP8266WiFi et ArduinoJson.

## 3.3 Implémentation et Validation du Mécanisme de Filtrage Triple

### 3.3.1 Problématique justifiant le filtrage

Lors des premiers tests en conditions réelles avec l'ESP32-CAM, des limitations critiques ont été observées en l'absence de mécanisme de filtrage. Ces limitations sont inhérentes au domain gap entre les images d'entraînement (PlantVillage, conditions contrôlées) et les images terrain (éclairage variable, distance variable, potentiel flou de bougé). Le Tableau 3.1 présente les principales limitations observées avant l'intégration du mécanisme de filtrage

TABLE 3.1 – Limitations observées sans mécanisme de filtrage

| **Limitation observée** | **Valeur mesurée** | **Impact** |
|---|---|---|
| Taux de faux positifs (saine → malade) | 12 % | Sur-fertilisation injustifiée |
| Cohérence inter-captures (même plante) | 71 % | Décisions instables |
| Utilisation de données obsolètes | Non contrôlé | Recommandations périmées |

### 3.3.2 Implémentation du filtrage

Après intégration du mécanisme de filtrage triple (seuil de confiance $\geq 70\,\%$ + fraîcheur temporelle $TTL \leq 60\,s$ + humidité du sol $< 50\,\%$), les tests réalisés sur le même ensemble de scénarios ont montré des améliorations significatives et mesurables. Le Tableau 3.2 présente une comparaison des performances du système avant et après l'application du filtrage triple

TABLE 3.2 – Comparaison des métriques système avant et après filtrage triple

| Métrique | Avant filtrage | Après filtrage | Amélioration |
|---|---|---|---|
| Cohérence inter-captures (même plante) | 71 % | 94 % | +23 pp |
| Faux positifs (saine → malade) | 12 % | 2 % | −10 pp |
| Annulation correcte des cycles lorsque le sol est humide | Non géré | 100 % | Nouveau |
| Latence de décision (ms) | ~120 ms | ~120 ms | Aucun impact |

Le mécanisme de filtrage améliore fortement la fiabilité du système en portant la cohérence inter-captures à 94 % et en réduisant les faux positifs à 2 %, tout en ajoutant une nouvelle capacité d'annulation des cycles en cas de sol humide, sans aucun impact sur la latence de décision ( 120 ms).

### 3.3.3 Contribution de chaque niveau de filtrage

— **Niveau 1 , Seuil de confiance (**$\geq 70\,\%$**)** : élimine les prédictions peu fiables dues à la mauvaise qualité de l'image. Ce filtre contribue à environ 60 % de la réduction totale des déclenchements erronés.
— **Niveau 2 , Fraîcheur temporelle (TTL – Time To Live** $\leq 60\,s$**)** : garantit que les données NPK transmises à ESP2 sont récentes. Il évite l'utilisation de recommandations obsolètes lors de changements de plantes ou d'une période d'inactivité du serveur.
— **Niveau 3 , Humidité du sol (**$< 50\,\%$**)** : filtre agronomique final annulant tout cycle d'irrigation lorsque le sol est déjà humide. Le taux d'annulation correcte a été mesuré à 100 % lors de l'ensemble des tests.

## 3.4 Discussion

### 3.4.1 Points forts validés

— **Performances CNN suffisantes pour l'usage visé :** une précision de 95,1 % sur 38 classes lors de la phase de validation représente un niveau de performance satisfaisant pour une application agricole réelle. Dans ce contexte, la robustesse et la fiabilité du diagnostic sont plus importantes que la recherche d'une précision maximale obtenue uniquement dans des conditions de laboratoire.
— **Mécanisme de filtrage triple essentiel et efficace :** la réduction de 83 % des déclenchements erronés démontre que ce mécanisme est indispensable pour garantir la robustesse du système en conditions réelles. Sans ce filtrage, le taux d'erreur de 18 % observé serait inacceptable pour une application destinée à la production agricole.
— **Machine à états finis déterministe et fiable :** les résultats des tests ont montré que 100 % des cycles NPK ont été correctement séquencés, confirmant la fiabilité de la FSM pour assurer un dosage agronomique sûr. Cette approche élimine les risques de sur-fertilisation liés à une activation simultanée ou incorrecte des pompes.

— **Système à faible coût et accessible :** l'utilisation de composants standard disponibles sur le marché, tels que l'ESP8266, l'ESP32-CAM et l'Arduino Mega, permet de réduire les coûts de mise en œuvre et rend la solution financièrement accessible aux exploitations agricoles algériennes.
— **Architecture modulaire et maintenable :** la répartition claire des responsabilités entre les cinq nœuds du système facilite les opérations de maintenance, la mise à l'échelle et les évolutions futures, notamment l'ajout de nouveaux capteurs, le remplacement du modèle d'intelligence artificielle ou l'extension de la base de connaissances `DISEASE_DB`.

### 3.4.2 Limitations identifiées

— **Dépendance à la qualité d'image de l'ESP32-CAM :** un éclairage insuffisant ou la présence d'images floues peuvent dégrader les performances du modèle CNN. L'intégration d'un système d'éclairage à LED contrôlé automatiquement ou d'un mécanisme de détection de flou permettrait d'améliorer la robustesse du diagnostic.
— **Écart de domaine (*Domain Gap*) avec le dataset PlantVillage :** la base de données PlantVillage est constituée d'images acquises dans des conditions contrôlées, ce qui crée un écart avec les images collectées sur le terrain en Algérie. Un réentraînement partiel (*fine-tuning*) du modèle à partir d'images locales acquises dans des conditions réelles permettrait d'améliorer significativement les performances de classification.
— **Point de défaillance unique (SPOF) du serveur PC/Flask :** l'absence de redondance du nœud central basé sur PC et Flask constitue un point de défaillance unique (*Single Point of Failure*). En cas de panne de ce serveur, l'ensemble de la chaîne de décision est interrompu jusqu'à son redémarrage manuel.

## 3.5 Conclusion

Ce chapitre a présenté l'implémentation complète du système et sa validation expérimentale à travers plusieurs axes de tests. Les résultats obtenus confirment la viabilité globale de la solution. Le modèle CNN affiche des performances satisfaisantes pour un usage agricole réel, où la fiabilité du diagnostic prime sur la recherche d'une précision maximale en conditions contrôlées. Le mécanisme de filtrage triple s'est révélé indispensable pour garantir la robustesse du système en conditions réelles, en éliminant la grande majorité des déclenchements erronés sans impacter la latence de décision. La machine à états finis a quant à elle assuré un séquençage correct et sûr de l'ensemble des cycles NPK. Sur le plan pratique, l'utilisation de composants standards et accessibles rend la solution déployable à faible coût, notamment dans le contexte des exploitations agricoles algériennes, tout en offrant une architecture modulaire favorable aux évolutions futures. Néanmoins, certaines limitations subsistent, notamment la sensibilité à la qualité d'image, l'écart entre les données d'entraînement et les conditions de terrain locales, ainsi que l'absence de redondance au niveau du serveur central. Ces points constituent des axes d'amélioration prioritaires pour les travaux futurs. Au terme de ce chapitre, la faisabilité et

l'efficacité du système proposé sont confirmées, posant des bases solides pour un déploiement à plus grande échelle.

# Conclusion générale

L'agriculture représente aujourd'hui un secteur confronté à de nombreux défis, notamment la raréfaction des ressources en eau, les changements climatiques et la nécessité d'améliorer la productivité tout en garantissant une utilisation rationnelle des ressources naturelles. Dans ce contexte, l'intégration des nouvelles technologies apparaît comme une solution prometteuse pour développer une agriculture plus intelligente, plus précise et plus durable.

Ce travail a été réalisé dans le but de concevoir et de mettre en œuvre un système intelligent d'irrigation automatisée basé sur la combinaison de IoT et IA. L'objectif principal était de proposer une solution capable de surveiller l'état des plantes, d'analyser les conditions environnementales et de prendre des décisions adaptées afin d'optimiser l'apport en eau et en éléments nutritifs.

Au cours de ce mémoire, nous avons étudié les concepts fondamentaux liés à l'agriculture intelligente, aux systèmes IoT, ainsi qu'aux différentes approches de l'intelligence artificielle, en particulier le Deep Learning et CNN utilisés pour l'analyse d'images et la détection des maladies des plantes.

La solution proposée repose sur l'utilisation de capteurs permettant de mesurer différents paramètres environnementaux tels que l'humidité du sol, la température et l'état général de la plante. Les données collectées sont ensuite transmises et analysées par un modèle d'intelligence artificielle afin de permettre une prise de décision automatique concernant l'irrigation et la gestion des besoins de la culture.

Les résultats obtenus montrent que l'association entre l'IoT et l'IA permet de concevoir des systèmes agricoles plus performants, capables de réduire le gaspillage d'eau, d'améliorer le suivi des cultures et de limiter les interventions humaines. Cette approche contribue ainsi à une meilleure gestion des ressources et à une augmentation de l'efficacité des pratiques agricoles.

Cependant, malgré les performances obtenues, plusieurs améliorations peuvent être envisagées dans les travaux futurs. Parmi ces perspectives, on peut citer l'amélioration du modèle d'intelligence artificielle afin d'augmenter sa précision, l'élargissement de la base de données des plantes et des maladies, l'intégration de données météorologiques pour une meilleure prédiction des besoins en eau, ainsi que le développement d'une plateforme de supervision à distance permettant à l'agriculteur de contrôler le système en temps réel.

En conclusion, ce projet représente une contribution vers la transition d'une agriculture traditionnelle vers une agriculture intelligente basée sur les données, l'automatisation et l'aide à la décision. L'intégration des technologies modernes telles que l'IA et l'IoT ouvre de nouvelles perspectives pour une agriculture plus durable, économique et respectueuse de l'environnement.

# Bibliographie


[1] Bilal Ahmed, Hasnat Shabbir, Syed Rameez Naqvi, and Lu Peng. Smart agriculture : Current state, opportunities, and challenges. *IEEE Access*, 2024.

[2] C. Akasiadis, V. Pitsilis, and C. D. Spyropoulos. A multi-protocol iot platform based on open-source frameworks. *Sensors*, 19(19) :4217, 2019.

[3] K. Alibabaei, P. D. Gaspar, T. M. Lima, and V. Soares. Irrigation optimization with deep reinforcement learning model. *Agriculture*, 12(2) :1–18, 2022.

[4] Richard G. Allen, Luis S. Pereira, Dirk Raes, and Martin Smith. *Crop Evapotranspiration : Guidelines for Computing Crop Water Requirements*. Number 56 in FAO Irrigation and Drainage Paper. Food and Agriculture Organization of the United Nations (FAO), Rome, Italy, 1998.

[5] Afshine Amidi and Shervine Amidi. Pense-bête de réseaux de neurones convolutionnels, 2019. Consulté le 01 juin 2026.

[6] AndroidPIT. Internet des objets : c'est quoi ? https ://www.androidpit.fr/internet-des-objets-c-est-quoi, 2017. Consulté le 23 mai 2026.

[7] Mike Attal. Convolutional neural network : Tout ce qu'il y a à savoir, 2026. Publié le 19 février 2026, consulté le 01 juin 2026.

[8] A. et al. Balafoutis. Precision agriculture technologies and ghg emissions. *Sustainability*, 9(8) :1339, 2017.

[9] N. Belaidi. L'apprentissage supervisé : définition et exemples, n.d. Consulté le 15 avril 2024.

[10] Rabaie Benameur, Amine Dahane, Bouabdellah Kechar, and Abou El Hassan Benyamina. An innovative smart and sustainable low-cost irrigation system for anomaly detection using deep learning. *Sensors*, 24(4) :1162, 2024.

[11] B. Benmammar and F. Krief. La technologie agent et les réseaux sans fil, 2004.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] Jean Buzer. Les réseaux convolutionnels, 2025. Documentation Deep Learning Avancé 1, consulté le 01 juin 2026.

[14] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[15] CodiMD. Machine learning - introduction et premier algorithme, 2024. Consulté le 20 mai 2024.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3) :273–297, 1995.

[17] S. J. Danbatta and A. Varol. Comparison of zigbee, z-wave, wifi, and bluetooth wireless technologies used in home automation. In *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–5. IEEE, 2019.

[18] Data Analytics Post. Réseaux de neurones récurrents. https ://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/. Date non spécifiée, consulté le 23 mai 2026.

[19] Lawrence E. Datnoff, Wade H. Elmer, and Donald M. Huber. *Mineral Nutrition and Plant Disease*. APS Press, 2007.

[20] Céline Deluzarche. Intelligence artificielle et deep learning. https ://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/, 2024. Futura Sciences, consulté le 23 mai 2026.

[21] D. Despommier. The rise of urban vertical farms. *Trends in Biotechnology*, 31(7) :388–389, 2013.

[22] G. Dreyfus et al. *Réseaux de neurones – Méthodologie et applications*. Éditions Eyrolles, Paris, 2 edition, 2008.

[23] Bouadi Dyhia and Imarazene Thanina. *Classification d'images agricoles avec le Deep Learning.* PhD thesis, Université Mouloud Mammeri, 2019.

[24] Olakunle Elijah, Tharek Abdul Rahman, Ismaila Orikumhi, Chee Yen Leow, and Muhammad N. Hindia. An overview of internet of things (iot) and data analytics in agriculture : Benefits and challenges. *IEEE Internet of Things Journal*, 5(5) :3758–3773, 2018.

[25] Olivier Ezratty. Les usages de l'intelligence artificielle, Novembre 2018. Document consulté pour étude et recherche.

[26] FAO. *Climate-Smart Agriculture Sourcebook*. Food and Agriculture Organization of the United Nations, Rome, 2013.

[27] S. A. Fennimore and M. Cutulle. Robotic weeders in strawberries. *Pest Management Science*, 75(11) :2982–2988, 2019.

[28] Konstantinos P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145 :311–318, 2018.

[29] J. Fürst, K. Chen, H.-S. Kim, and P. Bonnet. Evaluating bluetooth low energy for iot. In *2018 IEEE Workshop on Benchmarking CyberPhysical Networks and Systems (CPSBench)*, pages 1–6. IEEE, 2018.

[30] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[31] Antoine Grignola. Qu'est-ce qu'un cnn ? un convolutional neural network, 2024. Mis à jour le 13 juin 2024, consulté le 01 juin 2026.

[32] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2 edition, 2019.

[33] James L. Havlin, Samuel L. Tisdale, Werner L. Nelson, and James D. Beaton. *Soil Fertility and Fertilizers : An Introduction to Nutrient Management*. Pearson, 8 edition, 2014.

[34] Donald M. Huber and Richard D. Graham. The role of nutrition in crop resistance and tolerance to diseases. In *Mineral Nutrition of Crops*. 1999.

[35] F. Jiang, Y. Lu, Y. Chen, D. Cai, and G. Li. Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture*, 179 :105824, 2020.

[36] X. et al. Jin. Estimates of plant density from uav imagery. *Remote Sensing of Environment*, 198 :105–114, 2017.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.

[39] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982.

[40] A. M. Lonzetta, P. Cope, J. Campbell, B. J. Mohd, and T. Hayajneh. Security vulnerabilities in bluetooth technology as used in iot. *Journal of Sensor and Actuator Networks*, 7(3) :28, 2018.

[41] Manav Mandal. Convolutional neural networks (cnn), May 2021. Published on May 1, 2021 ; Last modified on April 28, 2023 ; Accessed : 01 June 2026.

[42] Petra Marschner. *Marschner's Mineral Nutrition of Higher Plants*. Academic Press, 3 edition, 2012.

[43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.

[44] H. Megnafi. Frequency plan optimization based on genetic algorithms for cellular networks. *Journal of Communications Software and Systems*, 16(3) :217–223, 2020.

[45] S. P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7 :1419, 2016.

[46] S. P. et al. Mohanty. Deep learning for plant disease detection. *Frontiers in Plant Science*, 7 :1419, 2016.

[47] Sharada Prasanna Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7 :1419, 2016.

[48] D. J. Mulla. Twenty five years of remote sensing in precision agriculture. *Biosystems Engineering*, 114(4) :358–371, 2013.

[49] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2) :103–134, 2000.

[50] Guillaume Plouin and Nicolas Colomer. Modèles d'architectures de l'internet des objets. https ://blog.octo.com/modeles-architectures-internet-des-objets/, 2011. Consulté le 23 mai 2026.

[51] H. B. Rebah. Gateway iot de pilotage et de surveillance des capteurs domestiques via le protocole mqtt. In *Colloque international sur les objets et systèmes connectés*, 2022.

[52] Stuart Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Prentice Hall, 3 edition, 2010.

[53] Rami Said Taha and Ahmed Ilyes Guendouz. *Modélisation et Implémentation d'un Système de Prédiction d'Arrosage Automatique des Plantes à l'aide de l'Apprentissage Automatique*. PhD thesis, Université ibn khaldoun-Tiaret, 2024.

[54] K. Saraubon, P. Chinakul, and R. Chanpen. Asset management system using nfc and iot technologies. In *Proceedings of the 2019 3rd International Conference on Software and e-Business*, pages 124–128, 2019.

[55] Z. Shelby, K. Hartke, and C. Bormann. The constrained application protocol (coap). Technical report, 2014.

[56] H. Sundmaeker, C. Verdouw, S. Wolfert, and L. P. Freire. Internet of food and farm 2020. In *Digitising the Industry – Internet of Things Connecting the Physical, Digital and Virtual Worlds*, pages 129–151. 2016.

[57] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 2 edition, 2018.

[58] Simon Thomine. Les couches de convolution — cours deep learning, 2023. Consulté le 01 juin 2026.

[59] O. Vermesan and P. Friess. *Internet of Things Applications – From Research and Innovation to Market Deployment*. Taylor & Francis, 2014.

[60] S. Wolfert, L. Ge, C. Verdouw, and M. J. Bogaardt. Big data in smart farming – a review. *Agricultural Systems*, 153 :69–80, 2017.