

Table des matières

| | |
|---|----|
| Introduction | 11 |
| Chapitre 01 : Internet des objets | 14 |
| 1. Principes de l'internet des objets (IoT)..... | 14 |
| 2. Les composants d'un système IoT..... | 15 |
| 2.1. Les objets (capteurs)..... | 15 |
| 2.2. Le réseau (connectivité)..... | 15 |
| 2.2.A.LAN/PAN (local/personal Area Network) | 15 |
| 2.2.A.1. Bluetooth..... | 16 |
| 2.2.A.2. RFID..... | 16 |
| 2.2.A.3. NFC (Near Field Communication)..... | 17 |
| 2.2.A.4. Zigbee | 17 |
| 2.2.A.5. Z-Wave..... | 18 |
| 2.2.A.6.Wifi(Wireless Fidelity)..... | 18 |
| 2.2.A.7. Bluetooth Low Energy (BLE)..... | 18 |
| 2.2.B. LPWAN (Low Power Wide Area Network)..... | 19 |
| 2.2.C. Réseaux cellulaires (longue portée)..... | 20 |
| 2.3. Les données..... | 21 |
| 2.4. Les applications d'exploitation | 21 |
| 3. Le protocole Ipv6 et internet des objet (IoT)..... | 21 |
| 4. Mode de fonctionnement de l'IoT..... | 22 |
| 5. Avantages et intérêts de l'IoT..... | 23 |
| 6. Défis et challenges de l'IoT..... | 23 |
| 6.1. La sécurité | 23 |
| 6.2. La gestion d'alimentation..... | 23 |
| 6.3. Le coût..... | 24 |
| 7. Exemples d'application IoT | 24 |
| 7.1. La domotique en milieux urbains..... | 24 |
| 7.2. L'Energie..... | 24 |
| 7.3. La santé..... | 24 |
| 7.4. L'agriculture..... | 24 |
| 8. Conclusion..... | 24 |
| Chapitre 02 : Intelligence Artificielle | 26 |
| 1. Définitions de l'intelligence artificielle..... | 27 |
| 1.1. Des systèmes qui pensent comme des humains..... | 27 |

| | | |
|---|---|-----------|
| 1.2. | Des systèmes rationnels..... | 26 |
| 1.3. | Des systèmes qui agissent comme des humaines..... | 26 |
| 1.4. | Des systèmes qui agissent rationnellement..... | 27 |
| 2. | Histoire de l'intelligence artificielle..... | 27 |
| 2.1. | Gestation de l'IA (1943 - 1955)..... | 27 |
| 2.2. | Naissance de l'IA (1956)..... | 28 |
| 2.3. | Enthousiasme des débuts : grandes espérances (1952 - 1969)..... | 28 |
| 2.4. | Epreuve de la réalité (1966 - 1973)..... | 28 |
| 2.5. | Systèmes fondés sur les connaissances (1969 - 1979) | 28 |
| 2.6. | L'IA devient une industrie (1980 - présent)..... | 29 |
| 2.7. | Retour des réseaux de neurones (1986 - présent)..... | 29 |
| 2.8. | L'IA devient une science (1987 - présent) Agents intelligents (1995 - présent)..... | 29 |
| 2.9. | Disponibilités de vastes ensemble de données (2001 - présent)..... | 29 |
| 3. | Prérequis de l'IA..... | 29 |
| 3.1. | Algorithme..... | 30 |
| 3.2. | Big Data..... | 30 |
| 3.3. | Cloud Computing..... | 31 |
| 4. | Branches de l'IA..... | 32 |
| 4.1. | Programmation symbolique..... | 32 |
| 4.2. | Machine Learning..... | 32 |
| 4.2.1. | Apprentissage supervisé..... | 33 |
| 4.2.2. | Apprentissage non supervisé..... | 33 |
| 4.2.3. | Apprentissage semi supervisé..... | 33 |
| 4.2.4. | Apprentissage par renforcement..... | 33 |
| 5. | Réseaux de neurones et Deep Learning..... | 33 |
| 5.1. | Entraînement d'un réseau de neurones profond..... | 34 |
| 5.2. | Types de réseaux de neurones profond..... | 35 |
| 6. | Intelligence Artificielle et IoT..... | 36 |
| 7. | Conclusion..... | 38 |
| Chapitre 03 : Approche proposée..... | | 40 |
| 1. | Description des capteurs d'acquisition IoT sélectionnés..... | 41 |
| 1.1. | La carte Arduino..... | 41 |
| 1.2. | NVIDIA Jetson Nano..... | 41 |
| 1.3. | ESP32..... | 42 |
| 1.4. | Les capteurs..... | 43 |
| 2. | Présentation schématique de notre système de poulailler intelligent..... | 46 |
| 3. | Résultats expérimentaux..... | 47 |
| 3.1. | Partie 01 : dispositif d'acquisition des données environnementales..... | 47 |
| 3.2. | Partie 02 : dispositif Edge IA d'acquisition et traitement de données visuelles | 50 |
| Conclusion et perspectives | | 54 |

Table des figures

| | |
|--|----|
| Figure 01 : Exemples d'applications exploitant l'IoT et l'IA..... | 12 |
| Figure 02 : Logo de l'alliance Zigbee(image via l'alliances Zigbee)..... | 18 |
| Figure 03 : Comparaison entre les principales technologies de réseaux..... | 20 |
| Figure 04 : Comparaison en termes de distance et de débit des technologies réseaux | 20 |
| Figure 05 : composants d'un système lot..... | 21 |
| Figure 06 : Représentation du fonctionnement de l'internet des objets..... | 22 |
| Figure 07 : Le test de Turing (1950)..... | 27 |
| Figure 08 : Exemple d'organigramme permettant de générer un algorithme..... | 30 |
| Figure 09 : Cloud Computing..... | 31 |
| Figure 10 : Architecture d'un réseau de neurones profond | 34 |
| Figure 11 : Exemple de réseau de neurone convolutionnel CNN..... | 36 |
| Figure 12 : Ressource Edge IA "Jeston Nano"..... | 38 |
| Figure 13 : Structure du poulailler..... | 40 |
| Figure 14 : Le Module ESP32..... | 42 |
| Figure 15 : Repérage des broches des modules ESP32..... | 42 |
| Figure 16 : Fonctionnement du capteur..... | 44 |
| Figure 17 : Le capteur DHT11..... | 44 |
| Figure 18 : Le format de trame de communication DHT11..... | 45 |
| Figure 19 : Le capteur de gaz MQ136..... | 45 |
| Figure 20 : Le capteur de gaz MQ135..... | 46 |
| Figure 21 : Schéma de fonctionnement du poulailler intelligent à l'aide de l'ESP32 et capteurs..... | 47 |
| Figure 22 : Schéma du poulailler intelligent à l'aide de la Jetson Nano et la caméra..... | 47 |
| Figure 23 : ESP32 expressif..... | 48 |
| Figure 24 : De gauche à droite, MQ135, MQ136 et AM2302..... | 48 |
| Figure 25 : Pont diviseur..... | 48 |
| Figure 26 : Valeurs des sur écran LCD I2C..... | 49 |
| Figure 27 : Serveur web hébergé sur l'ESP32..... | 49 |
| Figure 28 : (a) détection de O œuf, (b) détection de 10 œufs..... | 50 |
| Figure 29 : Exemple de détection d'animaux à l'aide de Yolo..... | 51 |
| Figure 30 : Exemple d'images de poules à partir de Google..... | 52 |

Liste des tableaux

Tableau I.1: Tableau comparatif des principaux protocoles réseau internet des objets.....19

Introduction :

Les dernières années ont vu une grande évolution des technologies liées à la collecte, au traitement et la transmission des données. Et cela grâce à l'exploitation de l'internet qui offre de plus en plus de capacités de connexions à haut débit. Ces technologies ont fortement contribué à l'émergence d'un nouveau domaine appelé « l'internet des objets, *Internet of Things : IoT* ». L'internet des objets est représenté par l'interconnexion qui peut avoir lieu entre différents objets grâce aux réseaux de télécommunication. Concrètement, l'internet des objets est considéré comme l'intersection du monde numérique et le monde des objets réels. C'est l'extension de l'Internet à des choses et des lieux du monde physique. Sa particularité est l'utilisation de technologies éprouvées comme l'adressages IP, les protocoles de communication, et les systèmes d'exploitation adaptés aux objets matériels connectés par la prise en compte de la taille et la consommation énergétique de ces objets.

L'IoT permet de gérer la communication entre les différents objets connectés sans intervention humaine et par conséquent rassembler et extraire de nouvelles informations à partir des données brutes collectées depuis les objets connectés. Dans ce contexte, l'institut Gartner prévoit d'ici 2025 qu'il y aura 75 milliards d'objets connectés sur le marché. Ceci représentera une véritable révolution numérique qui changera complètement le mode de la vie humaine ou les décisions seront de plus en plus liées aux données collectées en temps réel grâce aux objets connectés.

Par ailleurs, cette explosion du nombre d'objets connectés et de données requiert l'emploi d'algorithmes et de ressources de calcul efficaces en vue d'extraire les bonnes informations en un temps réduit. Dans ce contexte, les méthodes d'intelligence artificielle (IA) et plus particulièrement d'apprentissage automatique (Machine Learning) et Profond (Deep Learning) offrent une excellente solution pour générer des modèles efficaces exploitant des grandes quantités de données d'apprentissage collectées avec les objets IoT. En effet, les algorithmes IA permettent d'imiter l'intelligence humaine grâce à un processus de collecte de données inspiré du cerveau humain. L'internet des objets et l'IA sont actuellement utilisés dans un grand nombre

de domaines et d'applications : voitures autonomes, maisons intelligentes, bâtiments intelligents, etc. (Fig. 1).

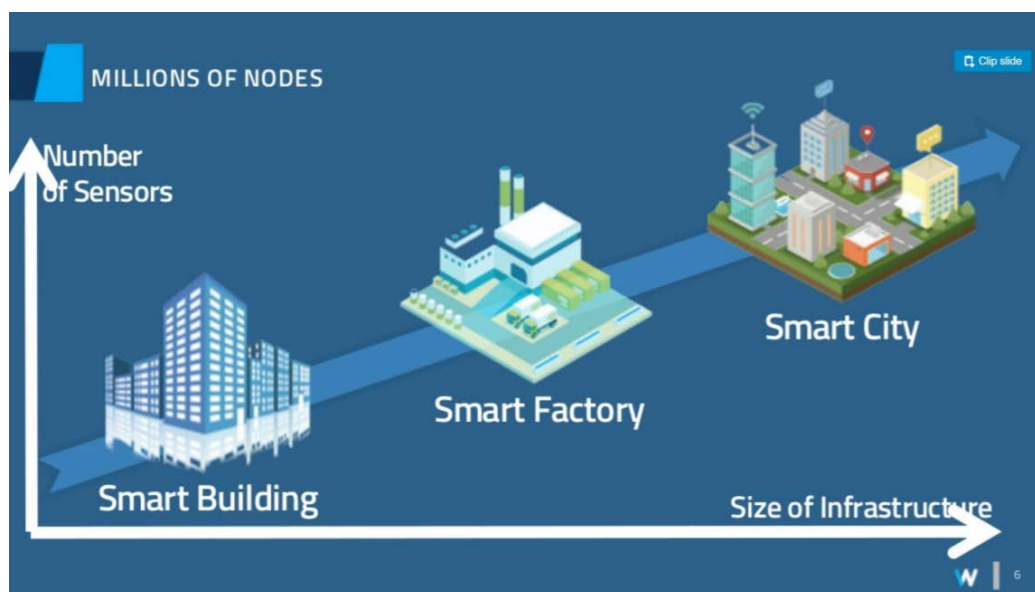


Figure 1 : Exemples d'applications exploitant l'IoT et l'IA

- **Contexte :**

Ce projet de fin d'études a comme objectif de développer et analyser une application exploitant à la fois les techniques IoT pour la collecte de données et les techniques d'intelligence artificielle pour le traitement et l'interprétation efficace de ces données. Après une analyse théorique des différents concepts liés aux deux domaines de l'IoT et de l'IA, nous mettrons en œuvre une application de gestion intelligente d'un poulailler connecté utilisant des capteurs connectés et des algorithmes d'apprentissage profond. Le choix des capteurs IoT et algorithmes IA a été réalisé après une analyse fine des besoins de notre application et une analyse des caractéristiques de chaque capteur IoT, ressource de calcul et algorithme IA.

- **Problématique :**

Nous proposons dans ce travail de fin d'études de développer un système intelligent pour le monitoring d'un poulailler que nous appelons « Système de poulailler intelligent ». Ce système devra inclure les composantes suivantes :

- a. **Système de collection de données capturées dans le poulailler :** ce dernier est composé de différents capteurs connectés (MQ135, MQ136, AM2302, etc.) permettant de

collecter les informations de température, d'humidité, de quantités de CO₂ et d'ammoniac au sein du poulailler.

- b. Systèmes de visualisation des données et contrôle :** à l'aide du microcontrôleur ESP32 qui récupère et centralise les données collectées avant de les visualiser via page Web et aussi un écran LCD. Ces informations permettant de contrôler la ventilation du poulailler afin de garantir des meilleures conditions de vie. Par exemple, si on constate des températures trop élevées, le système de ventilation pourrait être activé automatiquement.
- c. Système d'acquisition et traitement de données visuelles :** ce dernier est composé d'une caméra USB connectée à une ressource de calcul embarquée « Jetson Nano » qui permet d'analyser le flux vidéo collecté via la caméra avec des algorithmes IA. Ces derniers permettent de compter le nombre d'œufs et de poules pour savoir si les poules ont pondu et si toutes poules sont bien rentrées dans le poulailler par exemple. Ces informations permettent de savoir automatiquement le moment convenable pour fermer la porte du poulailler.

Ces composantes ont pour but d'offrir un suivi régulier de l'environnement intérieur du poulailler. Le principal objectif dans ce contexte est d'éviter les pertes dues aux décès des poules et aussi assurer un environnement propice à une meilleure productivité. Le contrôle intelligent est assuré à travers une série de capteurs et d'actuateurs et un algorithme d'intelligence artificielle pour prendre les décisions efficaces au bon moment comme l'aération ou l'ouverture/fermeture de portes.

Ce rapport est composé de quatre chapitres. Le premier décrit les principes théoriques et éléments de base du domaine de l'internet des objets (IoT), tandis que le chapitre deux est dédié à la description des algorithmes d'intelligence artificielle et d'apprentissage automatique (Machine et Deep Learning). Le troisième chapitre décrit notre prototype de poulailler intelligent exploitant à la fois des techniques (et capteurs) IoT et les algorithmes d'intelligence artificielle, une analyse et interprétation détaillée des résultats est présentée à la fin de chapitre. Dans le dernier chapitre, nous présentons la conclusion ainsi que les perspectives envisagées de ce travail.

CHAPITRE I

Internet des objets

Introduction :

Dans ce chapitre, nous présentons les principaux concepts de l'internet des objets (Internet of Things, IoT), les objectifs et les défis de l'IoT ainsi que les différents protocoles de communication qui peuvent être utilisés dans ce domaine. Nous présenterons ensuite des exemples d'applications IoT avec une description de leurs modes de fonctionnement et leurs liens avec le domaine d'intelligence artificielle. Ce chapitre est composé de huit parties :

- Principes de l'Internet des objets (IoT) ;
- Les composantes d'un système IoT ;
- Protocoles de communication dans un système IoT ;
- Le protocole IPv6 et internet des objets (IoT) ;
- Mode de fonctionnement d'un système IoT ;
- Avantages et intérêts de l'IoT ;
- Défis et challenges de l'Internet des Objets ;
- Exemples d'applications IoT.

1. Principes de l'internet des objets (IoT)

L'internet des Objets est appelé en anglais « *Internet of Things, IoT* » est représenté par l'interconnexion entre plusieurs objets connectés avec protocole de communication. Ces très nombreux objets peuvent identifier, mesurer, interagir et revêtir différentes formes (capteurs, appareils mobiles et sans fils, etc.) implantés dans des lieux et des environnements physiques divers.

La chaîne de traitement d'objets connectés permet de collecter, stocker et transmettre les données en vue de permettre l'extraction de nouvelles connaissances et formes de savoirs, notamment grâce aux algorithmes de communication réseaux et de l'intelligence artificielle [1].

L'internet des objets est souvent considéré comme la troisième évolution de l'Internet, appelée **Web 3.0 ou Web sémantique**. Cette grande capacité de collecte de données et d'extraction de connaissances de l'IoT lui a permis de s'implanter dans différents domaines et applications tels que l'e-santé, la vidéo surveillance, la sécurité (villes intelligentes et maisons intelligentes).

L'explosion des données générés grâce aux capteurs IoT a permis de promouvoir le domaine du Big Data qui consiste au développement de méthodes de traitement et de stockage distribué de données massives de manière plus rapide à l'inverse des systèmes de gestion de bases de données traditionnelles (relationnelles) qui ne sont pas adaptés aux données volumineuses. Par ailleurs, le Big Data tire profit des technologies du Cloud Computing qui offrent l'accès à distance aux ressources de calcul (à haute performance, HPC) capable de traiter des gros volumes de données en peu de temps. Le principal objectif de l'IoT est de créer un environnement d'espaces intelligents qui permettra d'améliorer la qualité de vie des gens en la rendant plus facile, plus modern et amusante en s'appuyant sur les données collectées par les capteurs, et qui permettent d'un côté un monitoring continue de l'environnement, et l'autre côté de prendre les décisions adéquates si cela est nécessaire (incendie, inondation, mauvaise qualité d'air, etc.).

2. les composantes d'un système IoT :

Les principaux éléments qui composent un système **IoT** sont :

2.1. Les objets (capteurs) : pour collecter les données depuis différentes sources. Ces données peuvent être de différents types : images de la caméra, température, humidité, position GPS, informations de santé, qualité de l'air, etc.

2.2. Le réseau (connectivité) : la connectivité dans un réseau IoT permet de relier l'information provenant de différents capteurs et couvrir une zone d'usage des objets tels une maison intelligente, un campus universitaire, une ville intelligente, la planète, etc. Un système IoT doit s'appuyer sur un protocole de communication pour garantir un fonctionnement efficace. En effet, il existe différents protocoles de communication qui diffèrent en fonction leur consommation d'énergie, de leur sécurité ainsi que de la distance à couvrir [2] [3] [4]. Les objets connectés doivent fonctionner en continu et sans interruption, ils doivent donc être reliés à une source d'énergie en permanence. Cependant ils doivent consommer un minimum d'énergie car ils sont souvent alimentés avec des piles ou des panneau photovoltaïques. C'est pour cela que ces protocoles de communication qui permettent la connectivité des capteurs doivent être les moins énergivores possibles. Dans la littérature, on peut citer 03 grandes catégories de protocoles de communication utilisés dans un contexte IoT : LAN/PAN, LPWAN, Réseaux cellulaires (longue portée).

A. LAN/PAN (Local/Personal Area Network) : cette catégorie de réseaux de communications englobe les protocoles de courte portée (Short range communication). Cela concerne les protocoles offrant une connectivité sure, autour et dans le corps humain

(santé, sport) tel que les protocoles RFID (portée jusqu'à 100m) ou NFC (portée de quelques centimètres, env. 10 cm). Cette catégorie englobe également les protocoles disposant d'une portée toujours courte mais un peu plus grande comme Bluetooth (10 à 20 m, 100m), ZigBee (portée de 10 à 50 m en intérieur.), etc. Ces deux catégories de protocoles sont utilisées par exemple dans le cadre des maison connectés (Zigbee, Zwave, etc.) ou pour les objets connectés à usage quotidien comme les bracelets, montres, smartphone ou puces connectées (Bluetooth, RFID, etc.). Dans les paragraphes suivants nous présentons les principaux protocoles utilisés dans ce contexte :

A.1 Bluetooth : La norme Bluetooth connue sous le nom IEEE 802.15.1 représente un outil de communication sans fil permettant l'échange de données numériques entre appareils électroniques, il fait partie des réseaux PAN (Personal Area Network). La connexion Bluetooth utilise des ondes radio en ultra haute fréquence ou UHF, sur la bande de fréquence autour de 2,4 GHz, avec une portée très réduite (quelque mètres, 10 à 20 m, 100m), ce qui la réserve à des usages à très courte distance. La principale utilité de cette technologie consiste à connecter différents appareils à un ou plusieurs hôtes, créant ainsi un réseau personnel (PAN), [22]. Depuis son apparition, la liaison Bluetooth est utilisée pour simplifier les connexions entre appareils électroniques, en supprimant les câbles. Elle permet de s'affranchir des fils. Parmi ces usages les plus fréquents aujourd'hui et qui viennent naturellement à l'esprit, notamment dans le cadre des loisirs, on pourra citer les liaisons entre ordinateur et souris, clavier, imprimantes et scanners etc. ou encore la liaison entre une manette de jeux vidéo et la console de jeux.

A.2 RFID : Le protocole RFID (Radio Frequency Identification) représente une technologie d'identification par radiofréquences permettant la communication entre l'objet et le détecteur situés à une distance allant de 1cm jusqu'à 100 mètres. Elle permet de scanner des produits en masse sans les voir, sans les toucher et donc permet un gain de temps considérable. Le système RFID fonctionne d'une manière très simple : l'étiquette RFID (ou bien le tag) est équipée d'une puce connectée à une antenne, qui permet à la puce de transmettre des informations qui peuvent être lues à l'aide d'un lecteur émetteur-récepteur. Aujourd'hui, cette technologie est répandue dans le monde entier et un des usages les plus évidents se situe dans les univers de la logistique et de la vente. Citons quelques exemples pratiques qui ont mis en œuvre avec succès la RFID :

- L'identification et suivi de personnes
- Gestion des stocks de produits
- Traçabilité des produits
- Identification des animaux de compagnie « Smart Farming »

A.3 NFC (Near Field communication) : c'est un protocole de télécommunication radio à très courte distance et dérivée de la technologie RFID. Cette dernière (NFC) permet de relier rapidement deux appareils électriques, la fonction principale de cette technologie est d'interagir avec un environnement donné. Il est très utile pour les Smartphones et les appareils similaire. Il existe trois modes de communication NFC [2][3]. Le protocole NFC permet à des appareils de communiquer sans contact sur une très courte distance (quelques centimètres). Il est utilisé pour les besoins de la téléphonie et des cartes de paiement. Il existe trois modes de communication NFC [2] [3] :

- **Le mode lecteur :** ou bien le mode actif, ou un dispositif NFC agit comme un lecteur sans contact quand il est mis à proximité d'une tag NFC.
- **Le mode « émulation de carte » :** il s'agit d'une communication entre un lecteur standard et un dispositif NFC vu par le lecteur comme un simple transpondeur. Les applications sont nombreuses : paiement sans contact, tickets de transport, tickets de spectacle, contrôle d'accès aux bâtiments
- **Le mode P2P :** Le mode peer-2-peer permet un échange rapide entre 2 appareils équipés de la technologie NFC.

A.4 Zigbee : connu sous le nom de la norme IEEE 802.15.4 qui représente un protocole de communication sans-fil à courte portée et à faible consommation énergétique et moins couteux. C'est un protocole de radio de communication d'une portée moyenne de 10 mètres. ZigBee est un protocole qui utilise une faible bande passante, idéal pour le transfert de données en faible volume. Parmi les applications de Zigbee on peut trouver :

- L'évitement de collision.
- Les maisons connectés (domotique)
- Les compteur intelligents (d'électricité et de gaz).



Figure 2 : Logo de l'alliance ZigBee (image via l'alliance ZigBee) ¹

A.5 Z-Wave : protocole de communication sans fil dédié à la domotique. Sa gamme de distance opérationnelle est comprise entre 30 et 100 mètres en plein air. Ce protocole peut être utilisé dans un réseau maillé, ce qui lui permet d'élargir sa portée.

A.6 Wifi (Wirless Fidelity) : le Wi-Fi est un réseau local lancé en 1999 qui utilise des ondes radio pour relier, sans fils, plusieurs appareils informatiques et faciliter la transmission de données. Il est aussi connu sous le nom IEEE 802.11. Le Wifi est principalement utilisé pour connecter des appareils à des liaisons haut débit (10.53 Gbit/s) avec une bande de fréquence comprise entre 2,4 GHz et 5 GHz. La sécurité du protocole Wifi repose principalement sur le mécanisme du protocole WPA (Wired Equivalent Privacy), introduit en 2003 et basé sur l'algorithme de cryptage des communications. Le Wifi repose également sur le filtrage d'adresses MAC, ce mécanisme permet de contrôler l'identité de l'appareil et de restreindre l'accès au réseau, mais dans le cas de l'IoT, le filtrage MAC est limité [2]. Le Wi-Fi permet des connexions à haut débit sur des distances comprises entre 20 et 100 mètres. Il permet également de transférer rapidement beaucoup de données. Cependant c'est un protocole très énergivore qui nécessite une connexion à une alimentation électrique. Il est utilisé pour la navigation internet et le transfert de données entre périphériques informatiques et multimédia.

A.7 Bluetooth Low Energy (BLE) : La technologie BLE est un protocole de réseau personnel sans fil à très basse consommation. Le protocole BLE permet de transférer une quantité limitée de données sur une distance moyenne de 60 mètres. Contrairement aux Bluetooth classique, le BLE a une consommation d'énergie dix fois plus petite. Il est utilisé pour le transfert de données entre périphériques informatiques et multimédia.

¹ ZigBee. <https://be.farnell.com/fr-BE/wireless-solutions-part-4-zigbee>

Le tableau 1 ci-dessous donne une comparaison entre les quatre principaux protocoles réseau internet des objets [2] :



























| |  WiFi |  ZigBee (802.15.4) |  Bluetooth |  NFC |
|------------------|--|--|---|---|
| Network topology |  Star |  Mesh |  Point-to-point |  Point-to-point |
| Range |  30-100 m |  10-20 m |  10 m |  < 0.1 m |
| Discovery |  Broadcast |  Broadcast |  Broadcast |  Response to field |
| Power |  High |  Low |  Classic: Mid  LE/Smart: Low |  Tag: Zero  Reader: Very low |
| Privacy |  Low |  Mid |  Mid |  High |

Tableau 1 : Tableau comparatif des principaux protocoles réseau internet des objets [2]

B. LPWAN (Low Power Wide Area Network) : Cette catégorie de réseaux concerne les réseaux sans fils basse consommation de moyenne portée. Ce sont des réseaux bas débit, offrant une longue portée (de 1,2 à une dizaines de kilomètres). Ils sont optimisés pour les équipements aux ressources limitées et pour lesquels une autonomie de plusieurs années est requise. Ces réseaux conviennent particulièrement aux applications dont les capteurs transfèrent peu de données et qui n'exigent pas un débit élevé. Nous trouvons beaucoup d'applications de ces types de protocoles dans le domaine des villes intelligentes (Smart Cities). Les deux protocoles les plus utilisés de cette catégorie sont SigFox et LoraWan

B.1 SIGFOX : c'est un réseau propriétaire payant dont la technologie utilise les bandes de fréquence comprise entre 868 MHz et 915 MHz. Il fonctionne sur une portée moyenne de 15 km. Il permet de transmettre des messages d'une taille de 12 octets (*bytes*).

B.2 LoraWAN : Protocole de communication à faible débit utilisant e la technologie de modulation LoRa. Il permet de transférer des données en petite quantité à des distances de 2 à 3 km en agglomération et jusqu'à 15 km en milieu rural. La taille de messages ne peut pas dépasser 242 octets.

les deux protocoles LoraWan et SigFox permettent de connecter des capteurs ou des objets caractérisés par une autonomie de batterie et un faible coût.

C. Réseaux cellulaires (longue portée) : Les réseaux cellulaires mobiles sont basés sur la technologie Global System for Mobile (GSM) et permettent de transférer une quantité importante de données sur une longue distance.

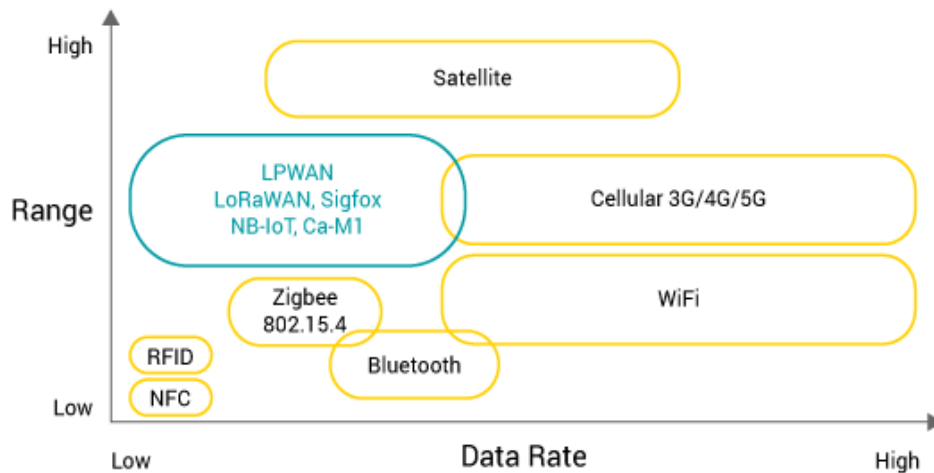


Figure 3 : comparaison entre les principales technologies de réseaux²

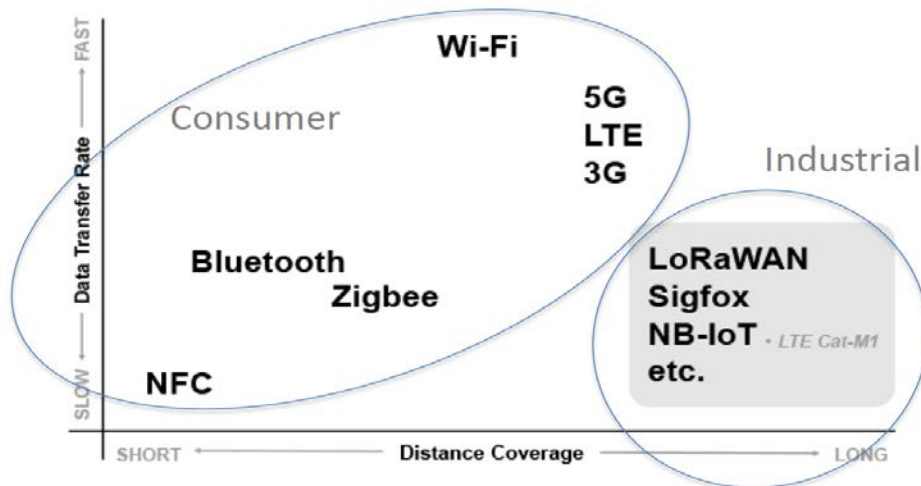


Figure 4 : comparaison en termes de distance et de débit des technologies réseaux³

2.3. Les données : c'est la raison d'être d'une chaîne de traitement. L'analyse de données de l'IoT permet la réalisation d'un traitement pour répondre à un besoin précis. Ces données sont collectées au format brut avant d'être stockées, structurées et archivées correctement

² <https://www.avsystem.com/blog/narrowband-iot/>

³ <https://www.murata.com/products/connectivitymodule/lpwa/outline/about>

afin de faciliter leurs traitements et par conséquent améliorer les performances des systèmes Io. Des informations peuvent être extraites à partir de données brutes collectées. Il faudra aussi stocker et structurer ces informations pour faciliter leur intégration dans les applications IoT finales.

2.4. Les applications d'exploitation : Les applications ont pour but d'exploiter les données et les informations pour prendre des décisions cohérentes et intelligentes. Ces applications sont souvent représentées par des interface **Homme-machine (IHM)** permettant de présenter les données sous forme de tableaux, graphiques, histogrammes et rapports. Ces derniers offrent la possibilité d'avoir une visualisation personnalisée en fonction des besoins des applications et des utilisateurs (tableau de bord, Dashboard). Notons que les traitements peuvent être réalisés via des ressources distantes (ressource Cloud) ou embarquées proches des capteurs (ressources Edge).

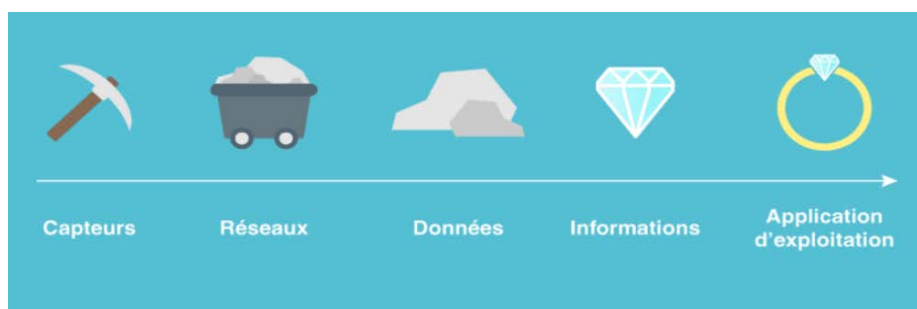


Figure 5 : composants d'un système IoT⁴

3. Le protocole IPv6 et internet des objets (IoT) :

Dans un contexte IoT tout dispositif connecté doit pouvoir être identifié grâce à une adresse IP. Le standard IPV4 qui stocke les adresses IP sur 4 octets ne permet plus de fournir de nouvelles adresses IP. C'est pour cela que le standard IPv6 qui stocke les adresses IP sur 16 octets est de plus en plus utilisé. Ce nouveau standard permet une plus grande disponibilité des adresses IP pour les objets connectés.

IPv6 « **Internet Protocol version 6** » a été principalement développé pour répondre aux exigences des adresses qui ne sont plus satisfaites par IPv4. Sachant que le protocole Internet version 4 est codé sur 32 bits et écrit sous forme décimale, son successeur IPv6 autorise les adresses de 128 bits, qui sont écrites en hexadécimal ce qui permet d'améliorer la lisibilité. Cette petite comparaison met en avant le problème central d'IPv4 qui est au nombre limité de bits « 32 bits ». L'utilisation de 128 bits dans IPv6 permet de générer plus d'adresses IP [5].

⁴ <https://www.digora.com/fr/blog/definition-iot-et-strategie-iot>

- **Espace d'adressage de IPv4** : 32 bits = 232 adresses ≈ 4,3 milliards d'adresses
- **Espace d'adressage de IPv6** : 128 bits = 2128 adresses ≈ 340 sextillions d'adresses

L'objectif principal de IPv6 est de fournir une adresse IP à un nombre maximal d'ordinateurs et d'objets connectés avec une meilleure sécurité.

4. Mode de fonctionnement de l'IoT

Un système IoT consiste à rassembler de nombreux composants techniques : des objets connectés aux applications ou services. Ce type de systèmes repose sur l'identification des objets pour collecter des données et les envoyer vers une plateforme ou une application cloud. Par conséquent, chaque objet a une adresse IP et le réseau permet à tous les objets, identifiés à l'aide d'adresses IP, d'être connectés. Afin de faire fonctionner le réseau Internet des objets, chaque capteur doit disposer d'un protocole de communication lui permettant de transférer les données à un serveur distant. Le capteur peut être connecté directement à internet par exemple en WIFI, mais cela lui demande d'être alimenté sur secteur en continue, ce qui n'est pas possible dans la majorité des cas. C'est pour cela que généralement, un réseau IoT est typiquement composé de capteurs qui communiquent avec des passerelles (Gateway) par un protocole de communication bas débit (ZigBee, Bluetooth, LoraWan, etc.). La passerelle qui connecte à Internet permet d'acheminer les données vers des serveurs Cloud [23].

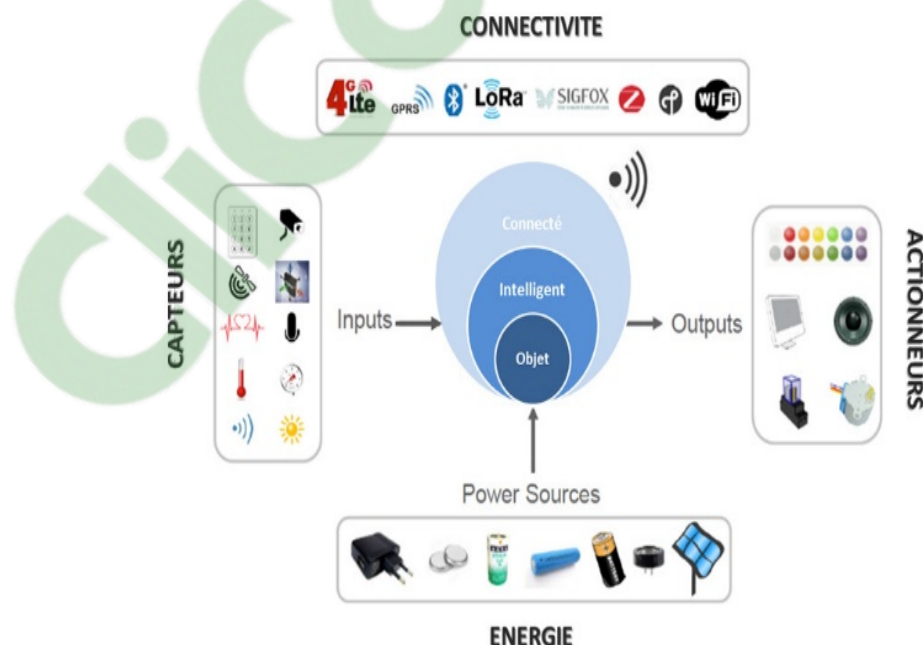


Figure 6 : Représentation du fonctionnement de l'internet des objets⁵

⁵ <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/>

5. Avantage et intérêts de l'IIoT :

Le principal avantage de l'IIoT est représenté par la valeur qu'on peut avoir grâce à la collecte rapide, fiable, continue de suffisamment de données. Ces aspects permettent aux entreprises d'augmenter leurs chiffres d'affaires et garantir une meilleure rentabilité puisque les technologies développées pourront exploiter suffisamment de données pour décider intelligemment et convenablement.

Un avantage majeur de l'IIoT est la quantité des données provenant de différents capteurs. Cela permet principalement un monitoring continu et une prise de décision rapide dans des situations critiques. La fiabilité des données peut être assurée grâce un post-traitement qui vérifie chaque donnée au moment de sa collection grâce à une analyse comparative par rapport aux données d'autres capteurs et les données précédemment collectées. Cette fiabilité de données est nécessaire et est d'une grande utilité dans les applications de vidéo surveillance de développement de systèmes de maisons et villes intelligentes.

6. Défis et challenges de l'IIoT :

L'Internet des objets peut être défini par une infrastructure basée sur l'interconnexion intelligente des objets connectés. Cependant, l'IIoT doit faire face à plusieurs défis et obstacles qui sont à franchir avant d'atteindre les objectifs d'efficacité, de robustesse et de qualité continue de service. Parmi ces défis nous citons principalement :

6.1. La sécurité : c'est le principal défi pour l'IIoT puisque la majorité des objets connectés à Internet sont des cibles importantes pour les pirates et autres mauvais acteurs. Jusqu'à présent, il n'existe pas de réel standard de sécurité pour protéger les informations des appareils connectés (capteurs, actionneurs, smartphones, etc.) [6]. Il y'a des solutions basées sur la technologie Blockchain, mais elles sont encore difficiles à mettre en œuvre et ne sont pas utilisées massivement pour tous les objets connectés.

6.2. La gestion d'alimentation : aujourd'hui, nous avons plus de 50 milliards d'objets connectés qui doivent être alimentés avec de l'énergie. Ce grand besoin en énergie présente un véritable défi majeur pour l'IIoT (Énergie limitée et durée de vie limitée). La

valorisation énergétique environnementale comme les cellules photovoltaïques, les générateurs thermoélectriques ou encore les générateurs piézoélectriques peuvent être utilisés comme solutions ou compléter les systèmes d'alimentation par batterie [8].

6.3. Le coût : afin de pouvoir couvrir un public plus large, en particulier les entreprises de taille moyenne, le coût des composants utilisés (capteurs, microcontrôleur, etc.) doit être considérablement réduit.

7. Exemples d'applications IoT :

Dans la littérature, il existe plusieurs applications intelligentes dans le domaine de l'IoT qui touchent principalement le domaine de la santé, l'agriculture, l'industrie, etc. Voici quelques exemples [7] :

7.1. La domotique en milieux urbains : cette technique permet de contrôler les différents équipements d'une maison depuis une même interface et ceci par la mise en place de dispositifs domestiques connectés, contrôlable à distance [7].

7.2. L'Énergie : une distribution et une gestion efficace de l'énergie peut être géré en temps réel par des objets connectés, ce qui permet le control de la consommation d'Énergie et la détection des anomalies [7].

7.3. La santé : Le secteur de la santé a vu de nombreuses applications permettant aux patients et à leurs médecins de recevoir des informations, parfois même en temps réel, qui ne pouvaient pas être connues avant l'avènement de l'IoT. L'une des principales utilisations des objets connectés dans ce domaine est le maintien des patients à domicile grâce au monitoring en temps réel des fonctions vitales et une connectivité permanente avec le médecin traitant et les services de secours.

7.4. L'agriculture : L'utilisation des capteurs dans ce domaine d'agriculture intelligente permet la collecte d'informations utiles relatives au sol, à l'humidité, la teneur en sels minéraux et d'autres conditions en vue de renforcer la capacité des systèmes agricoles, de contribuer à la sécurité alimentaire.

Conclusion

Nous avons présenté dans ce chapitre une vision générale des principes, technologies et domaines d'applications de l'IoT. Nous pouvons constater l'importance de cette technologie dans notre vie quotidienne ainsi que son impact sur le bien-être des animaux que nous allons traiter dans le troisième chapitre avec l'exemple du poulailler intelligent.

CHAPITRE II

Intelligence Artificielle

Introduction :

Le domaine d'intelligence artificielle (IA) devient de plus en plus connu par le grand public et fait partie de notre vie quotidienne puisque les applications IA sont en constantes évolutions. L'IA est utilisé dans plusieurs domaines tels que la vision par ordinateur, l'imagerie médicale, la vidéo surveillance, la robotique, les jeux vidéo, etc. Dans ce chapitre, nous allons présenter les principes et différentes techniques d'intelligence artificielle et leur utilité dans le domaine IoT et plus particulièrement notre exemple du poulailler intelligent. Ce chapitre est décrit en Cinq parties :

1. Définitions de l'intelligence artificielle
2. Histoire de l'intelligence artificielle
3. Branches de l'intelligence artificielle
4. Réseaux de neurones et Deep Learning
5. Intelligence artificielle et IoT

1. Définitions de l'intelligence artificielle :

L'Intelligence artificielle peut être définie par les techniques permettant de construire des machines ou des ordinateurs avec une intelligence semblable ou même supérieure à celle de l'homme. Le but est de donner à ces machines intelligentes la capacité d'effectuer des tâches qui nécessitent généralement une intelligence humaine. Depuis la création du domaine de l'IA en 1950, sa définition a évolué au fur et mesure de son avancement. Nous pouvons citer quatre principales définitions de l'IA :

- 1.1. Des systèmes qui pensent comme les humains :** le but de ces systèmes est d'automatiser la pensée humaine qui englobe la prise de décision, la résolution des différents problèmes ainsi que tous ce qui est lié à l'apprentissage. Le développement de ces systèmes s'appuie sur l'analyse interne du fonctionnement du cerveau humaine à l'aide des sciences cognitives.
- 1.2. Des systèmes rationnels :** Le but de ces systèmes est de rendre possible la perception, le raisonnement ainsi que l'action. Ces systèmes reposent en partie sur la pensée logique du philosophe « Aristote », l'un des premiers à essayer de codifier « le bien penser ». Exemple : « Socrate est un homme, tous les hommes sont mortels Donc Socrate est mortel »
- 1.3. Des systèmes qui agissent comme les humains :** Le but de ces systèmes est de développer les outils permettant aux ordinateurs (ou machines) d'accomplir des tâches requérant une intelligence humaine. L'idée est de pouvoir agir comme l'homme sans devoir s'inquiéter de son mode de raisonnement ou de sa pensée. Ce type de systèmes s'appuie sur le test de Turing proposé en 1950 qui suppose que : « une machine est considérée comme intelligente si elle peut mener une

conversation de telle manière que les interrogateurs humains ne peuvent la distinguer d'un être humain [8] » (Fig. 7). Le test de Turing demeure valide Jusqu'à présent [9].

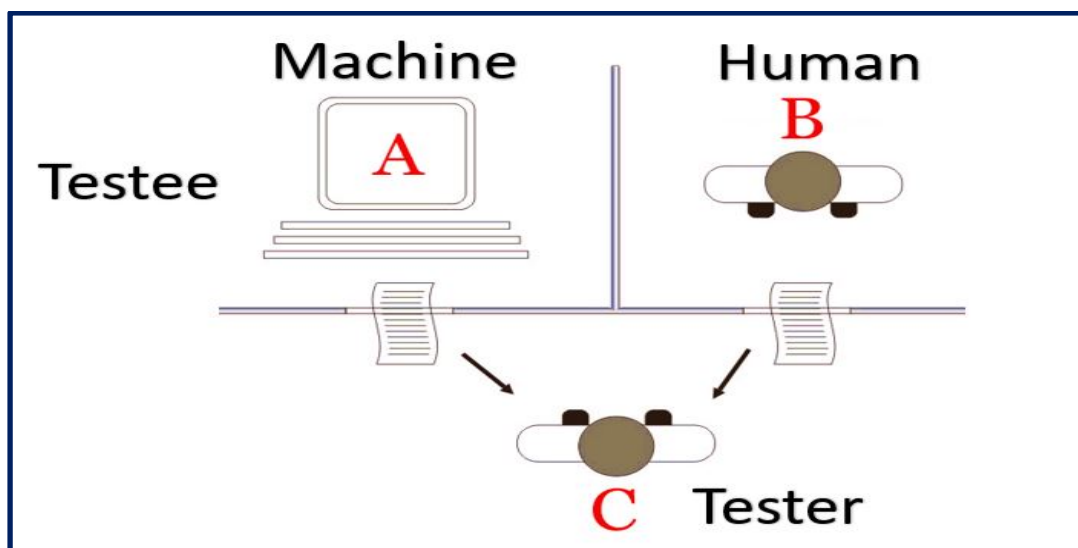


Figure 7 : le test de Turing (1950)⁶

- 1.4. **Des systèmes qui agissent rationnellement** : le but de ces systèmes est de concevoir des agents intelligents ou chacun doit fonctionner de manière autonome après avoir collecté les informations de son environnement qui lui permettent de réaliser les actions nécessaires. Les agents doivent persister pendant une période prolongée et s'adapter aux nouveaux changements d'environnement et objectifs. Les agents peuvent également communiquer entre eux afin d'exploiter l'information et l'intelligence provenant des différents agents. Ce type de systèmes est appelé « Systèmes Multi-agents » qui est fort utilisé dans plusieurs applications.
2. **Histoire de l'intelligence artificielle** : les différentes définitions de l'intelligence artificielle décrites ci-dessus montrent que l'IA dispose d'une longue histoire depuis sa création ou même avant sa création. Nous pouvons citer les principales étapes de l'histoire de l'IA :
 - **Gestation de l'IA (1943 -- 1955)** : durant cette phase, le premier modèle de neurones artificiels [8] a été proposé en 1943 où chaque neurone est caractérisé par un état : marche ou arrêt. Toute fonction est donc calculée par un réseau de neurones connectés ce qui a conduit au développement du premier ordinateur (SNARC) avec un réseau de (40) neurones [8] en 1950. Depuis cette avancée, plusieurs conférences ont été données par Alan Turing dégageant une feuille de route dans son article de 1950 « Computing Machinery and Intelligence ». Il a introduit ensuite le test de Turing décrit ci-dessus.

⁶ <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/>

- **Naissance de l'IA (1956)** : en 1956, le Professeur John McCarthy a décidé d'organiser un séminaire de deux mois réunissant dix chercheurs américains dans le but d'étudier l'intelligence artificielle via les domaines de théorie des automates, réseaux de neurones et formalisation des concepts. A l'issue de ce séminaire, les deux chercheurs « Newell et Simon » ont présenté un programme capable de raisonner logiquement « le Logic Theorist (LT) ». Malheureusement, les résultats n'ont pas pu être publiés suite au rejet de l'article mais cette étude a tout de même permis de faire faire connaître le domaine de l'IA dans monde scientifique.
- **Enthousiasme des débuts : grandes espérances (1952 - 1969)** : cette période a été marquée par les premiers succès de l'IA en termes d'outils de programmation et ordinateurs. En effet, après la théorie LT, Newell et Simon ont proposé le GPS « General Problem Solver », qui a la différence de LT, permet d'imiter les humains dans la résolution des problèmes. GPS représente le premier programme à intégrer l'approche de la pensée humaine. En 1959, Herbert Gelernter (1959) a construit le Geometry Theorem Prover capable de démontrer de nombreux théorèmes.
- **Epreuve de la réalité (1966 - 1973)** : cette période a connu un excès de confiance suite aux résultats promoteurs de l'IA obtenus durant la phase précédente. Par exemple, on prédisait qu'un ordinateur pourrait être champion du monde au jeu d'échecs dans 10 ans, or il a fallu 40 ans pour cela se concrétiser. Suite à cet avancement plus lent que prévu, plusieurs programmes de financement de recherche en IA ont été arrêtés notamment aux Etats Unis et en Russie. Cette période est également connue pour l'intitulé « **Hiver de l'IA** »
- **Systèmes fondés sur les connaissances (1969 – 1979)** : afin de garantir des résultats flexibles et supportant les changements d'échelles, les chercheurs ont pensé à recourir aux connaissances déjà acquises pour résoudre un problème difficile. Cela veut dire que pour résoudre un problème il faut avoir une idée des anciennes solutions. Cette période a connu la découverte du premier « système expert » utilisant de manière intense une connaissance spécifique. De plus, plusieurs langages de représentation de connaissances et de raisonnements ont été proposés : OWL, SKOS, W3C, etc.
- **L'IA devient une industrie (1980 - présent)** : cette période a connu le premier système expert « R1 » réussi commercialement, chez Digital Equipment Corporation (McDermont, 1982). Ce dernier permet de configurer les ordinateurs en fonction des commandes clients. Durant cette phase, le chiffre d'affaires des projets en lien avec l'IA passait de quelques millions de dollars en 1980 à plusieurs milliards de dollars en 1988.

- **Retour des réseaux de neurones (1986 - présent) :** durant les années 1980, quatre groupes de chercheurs ont inventé l'apprentissage par rétro-propagation [10] introduit initialement en 1969. Ce principe a permis de développer des réseaux de neurones avec une capacité d'apprentissage très importante. En effet, grâce à la rétropropagation, l'entraînement des réseaux de neurone s'appuie sur la modification des paramètres d'entraînement et modèles en fonction des erreurs précédemment obtenues.
- **L'IA devient une science (1987 - présent) :** L'intérêt à l'IA devient de plus en plus important jusqu'au point de considérer l'IA comme une science ce qui permis de reproduire des expérimentations grâce aux dépôts et partages de code et de données de test par la communauté des scientifiques. Ceci a permis d'améliorer les solutions existantes sans devoir les redévelopper de nouveau. Par conséquent, l'IA se retrouve utilisée dans plusieurs domaines : reconnaissance de la parole, traduction automatique, réseau de neurones, data mining, etc.
- **Agents intelligents (1995 - présent) :** les agents rationnels ont une place centrale dans le domaine de l'IA puisqu'ils offrent la possibilité d'appliquer le concept de rationalité à un grand nombre d'agents opérants dans des différents environnement. On trouve les agents intelligents utilisés dans différentes applications : robots aspirateurs, moteurs de recherche, etc.
- **Disponibilités de vastes ensembles de données (2001 – présent) :** les 60 années précédant cette phase avaient mis l'accent sur les algorithmes en premier lieu or les derniers travaux proposent de s'appuyer sur les données et d'être moins exigeant sur les algorithmes. Ce choix a permis de développer des algorithmes plus performances grâce à l'exploitation des connaissances extraites à partir de gros volumes de données qu'ont peut avoir à l'aide des technologies IoT et Big Data et Cloud computing. Cette période est également appelée par le « **Printemps de l'IA** » qui vient après l' « **Hiver de l'IA** »

3. Prérequis de l'IA : le succès de l'IA est obtenu grâce à trois facteurs principaux : algorithme, Big Data et Cloud Computing :

- **Algorithme :** ce terme est issu du nom du mathématicien « Al-Khwarizmi » . Un algorithme est représenté par une série limitée et bien définie d'opérations ou d'instructions qui peuvent résoudre un type de problème. Il permet en particulier d'appliquer des formules mathématiques au mode réel à travers l'informatique (un programme exécutable par un ordinateur). L'algorithme offre la possibilité de réutiliser une solution sur plusieurs

données et par conséquent garantir une flexibilité de solutions. Cette propriété de réutilisabilité représente la principale caractéristique de algorithmes et modèles IA.

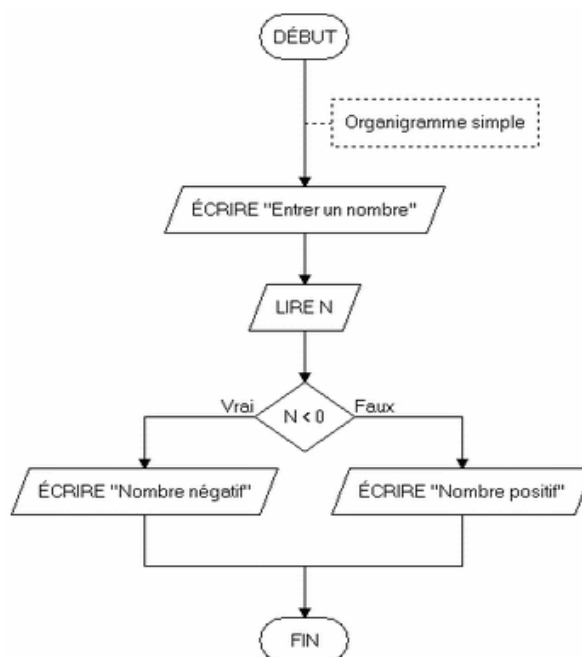


Figure 8 : exemple d'organigramme permettant de générer un algorithme⁷

- **Big Data** : avec l'évènement de l'IoT, des réseaux sociaux et l'augmentation de la bande passante des réseaux, le monde a connu une explosion de la quantité de données lors des 10 dernières années. Cette explosion s'est retrouvée face à un problème majeur de stockage et gestion de données puisque les systèmes traditionnels de gestion de bases de données (relationnelles) ne sont pas capables de gérer une aussi grande quantité de données en un temps raisonnable. Dans ce contexte, les technologies de Big Data ont été proposées pour gérer le stockage distribué et rapide de gros volumes de données, ainsi que le traitement distribué de ces données à l'aide d'infrastructure de calcul à haute performances. Cette capacité de gérer des volumes massifs est d'une grande utilité pour les algorithmes d'intelligence artificielle puisque ces derniers peuvent exploiter les connaissances extraites à partir d'un volume suffisamment grand pour identifier les bons paramètres. Le Big Data est connu par quatre (4) caractéristiques principales (4V) :
 - **Volume** : Volumes massifs de données ;
 - **Vélocité** : Vitesse d'acquisition de données notamment grâce aux capteurs IoT et réseaux sociaux ;

⁷ <https://larp.marcolavoie.ca/fr/description/description.htm>

- **Variété** : diversité des données qu'on peut collecter suite à la diversité des capteurs et d'informations.
- **Véracité** : fiabilité de données pouvant être vérifiées à l'aide de ressources embarquées « Edge »
- **Cloud Computing** : est représenté par les technologies permettant d'accéder à distance à des services informatiques (serveurs, stockage, mise en réseau, logiciels) via Internet à par un fournisseur (exp. AWS, Google, Cloud, Microsoft Azure, etc.). Le Cloud Computing est souvent représenté comme origine du Big Data puisqu'il permet d'avoir accès à des ressources de calcul adaptées aux technologies Big Data qui requièrent des traitements parallèles et distribués ne pouvant pas être réalisé sur des simples ordinateurs de bureau. Par transition, on peut dire que le Cloud Computing est fort utile pour l'IA grâce à son aide pour l'aide et la gestion de données volumineuses, mais aussi sa mise à disposition du matériel capable d'entraîner des modèles d'apprentissage exigeant une puissance de calcul élevé (Fig. 9).

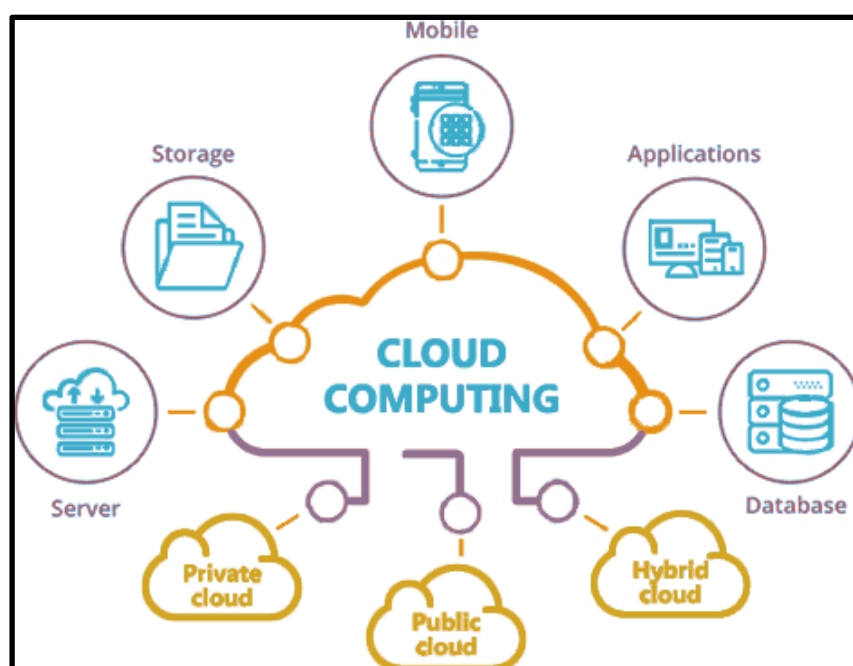


Figure 9: Cloud Computing⁸

4. Approches principales de l'IA : nous pouvons citer deux principales approches de l'intelligence artificielle : programmation symbolique et apprentissage machine ;

4.1. Programmation symbolique : consiste à coder les problèmes à l'aide d'un ensemble d'instructions qui doivent être vérifiées afin de fournir une décision. Ce type de solutions

⁸ <https://networkencyclopedia.com/cloud-computing/>

est simple à développer mais difficilement généralisable puisqu'on peut l'appliquer à de simple problèmes mais pas à des problèmes complexes. Par exemple, à partir d'une image, la présence d'une queue en plus de quatre pattes (2 règles) permet de conclure qu'il s'agit d'un animal. Par contre, si on veut savoir la race de l'animal cela impliquerait d'analyser tous les pixels de l'image, ce qui devient irréalisable en termes de nombre de possibilités.

4.2. Apprentissage machine « Machine Learning » : pour faire face au problème de la programmation symbolique, nous pouvons exploiter les techniques d'apprentissage machine qui s'appuient sur l'apprentissage à partir de données et non pas des règles. Autrement dit, les algorithmes de « Machine Learning » utilisent des techniques statistiques, issues d'exemples fournis, pour repérer des modèles dans les données avant de prendre les meilleures décisions. Les algorithmes d'apprentissage s'appuient généralement sur trois phases : entraînement, validation et test.

- **Entraînement** : à partir d'un ensemble de données suffisamment grand, le modèle sera testé et modifié après chaque itération « époque » jusqu'à obtention du meilleur résultat.
- **Validation** : après chaque itération, le modèle est testé avec un autre jeu de données. Cela permettra d'éviter que le modèle soit entraîné de manière aveugle sur les données d'apprentissage ce qui générerait un problème de surapprentissage.
- **Test** : après génération du modèle, ce dernier peut être testé avec un nouveau jeu de test. Des bons résultats confirment que la méthode proposée est efficace et vice versa.

Les phases de collecte et de préparation des données représentent des étapes très importantes mais qui demandent énormément de ressources humaines. Notons aussi qu'il est indispensable de disposer de suffisamment de données pertinentes et variées. De plus, il est très souvent nécessaire de passer par l'étape d'annotation qui consiste à étiqueter les données avec le nom de la catégorie à laquelle elles appartiennent ou encore la valeur numérique qu'elles représentent (apprentissage supervisé). Il est également nécessaire d'indiquer à l'algorithme les caractéristiques à prendre en compte dans les données. Il existe différents types d'apprentissages, parmi les plus répandus, on retrouve l'apprentissage supervisé, non supervisé, semi-supervisé, ainsi que l'apprentissage par renforcement :

4.2.1. Apprentissage supervisé : consiste à développer des algorithmes permettant d'apprendre à partir d'un ensemble de données où chaque entrée (X) est étiquetée par un label (Y). Les labels permettent d'évaluer le modèle et réduire l'erreur durant l'entraînement [11]. Parmi les algorithmes les plus populaires pour ce type d'apprentissage, on peut citer : les arbres de décision, la régression linéaire, la régression logistique, KNN (K Nearest Neighbors), les réseaux de neurones, etc.

4.2.2. Apprentissage non supervisé : dans ce type d'apprentissage, on ne connaît pas les étiquettes (Y) mais seulement les données non labélisées (X). Le modèle d'apprentissage non supervisé doit analyser et explorer les données afin de d'extraire les caractéristiques les plus représentatives pour chaque groupe (cluster) de données [12].

4.2.3. Apprentissage semi supervisé : ce type d'apprentissage s'appuie à la fois sur des données étiquetées (X, Y) et des données non étiquetées (X) pour entraîner les modèles. D'une part, ce type d'apprentissage permet d'exploiter une approche supervisée et d'autre part de profiter d'une grande quantité de données non supervisées qui sont généralement moins chères et plus faciles à collecter [13].

4.2.4. Apprentissage par renforcement : il s'agit d'un apprentissage par essai/erreur qui consiste à apprendre les actions à prendre, à partir d'expériences, de façon à optimiser une récompense quantitative au cours du temps et à mesure. L'algorithme multiplie les tentatives pour tenter de découvrir les actions permettant de maximiser une fonction objective calculée à partir des récompenses. Dans ce contexte, les données d'apprentissage sont directement collectées à partir de l'environnement. Ce type d'apprentissage est fort utilisé dans la programmation des jeux vidéo.

5. Réseaux de neurones et Deep Learning :

L'apprentissage profond (Deep Learning) représente une branche de machine Learning qui s'inspire du fonctionnement du cerveau humain basé sur des réseaux de neurones connectés. Le Deep Learning se base sur le développement d'architectures neuronales (ANN, Artificial Neural Network) composés d'un ensemble de couches connectées :

- **Couche d'entrée :** permettant de représenter les données d'entrée (X)
- **Couches intermédiaires :** pour combiner les données d'entrée et extraire de nouvelles caractéristiques. La combinaison des caractéristiques se fait à l'aide de poids (W).

- **Couche de sortie** : permettant de présenter le résultat.

Un réseau de neurones est considéré comme réseau de neurones profond s'il dispose de plus d'une seule couche intermédiaire (les couches d'entrée et sortie doivent être obligatoirement présentes). Les réseaux de neurones profonds peuvent être utilisés dans différents types d'applications : classification d'images, reconnaissance d'objets, reconnaissance de la parole, reconnaissance d'actions, segmentation d'images médicales, prédiction de prix de logements, détection de défauts de fabrication, etc.

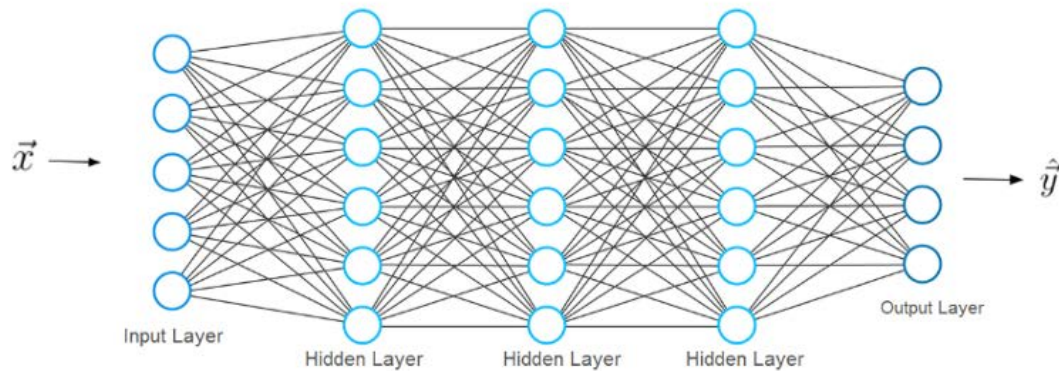


Figure 10 : Architecture d'un réseau de neurones profond ⁹

5.1. Entraînement d'un réseau de neurones profond : La figure 3 illustre la connexion entre les couches puisque chaque neurone est connecté à tous les neurones issus des couches adjacentes. Cette connexion est représentée par une valeur numérique appelée « poids ». Chaque neurone calcule une somme pondérée des valeurs qu'il reçoit de la couche précédente avant de transformer cette somme avec une fonction mathématique appelée « fonction d'activation ». Ceci permettra d'avoir des valeurs normalisées pour les neurones ou les neurones ayant une valeur de sortie proche de 1 sont considérés comme caractérisâtes intéressantes et vice versa. Les principales étapes de fonctionnement d'un réseau de neurones profond sont :

- Initialisation** : des poids (W) pour chaque neurone avec des valeurs aléatoires ;
- Forward Pass** : appliquer le réseau de neurones à partir des données d'entraînement (X) en utilisant les poids aléatoires ;
- Calcul d'erreur** : comparer les résultats obtenus à l'aide du réseau avec les prédictions réelles (Y) et calcul un score d'erreur.

Clicours.COM

⁹ <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac?gi=febcb3e940d>

- d. **Backpropagation** : mettre à jour les poids du réseau de neurones afin de réduire l'erreur à l'aide de l'algorithme de descente du gradient
- e. **Itération** : répéter le processus jusqu'à obtention d'un modèle avec une erreur faible voire nulle.

5.2. Types de réseaux de neurones profonds : la diversité des données qu'on peut rencontrer a conduit aux développements de différents types de réseaux de neurones profonds. Les principaux types de réseaux de neurones que nous pouvons citer sont :

- a. **Réseaux de neurones multicouches (MLP, Multi-Layer Perceptron)** : réseaux qui possèdent un nombre de couches cachées limitées et dont les données d'entrée sont représentés par un vecteur à une dimension.
- b. **Réseaux de neurone convolutionnels (CNN, Convolutional Neural Networks)** : ce type de réseaux de neurones est fort utilisé dans le domaine de vision par ordinateur et plus particulièrement la classification d'images et la localisation d'objets. La couche d'entrée de ces réseaux est représentée par une image (représentée par une matrice en deux dimension) à partir de laquelle des convolutions peuvent être appliquées pour extraire des caractéristiques d'images. Cette extraction peut être réalisé durant plusieurs couches (phase d'extraction de caractéristiques). À l'issue de cette phase, les caractéristiques sont concaténées en un vecteur à une dimension ce qui permettra d'appliquer un réseau MLP et de prédire le résultat de classification ou de localisation ou autre. La figure 4 illustre un exemple d'architecture de classification d'images à l'aide d'un réseau de neurones convolutionnel [14].
- c. **Réseaux de neurones récurrents (RNN)** : à l'instar des réseaux CNN permettant de prendre en compte l'information spatiale des pixels d'images, les réseaux RNN permettent de prendre en compte l'information temporelle des données pour fournir une précision optimale. Ce type de réseaux est fort utilisé pour des applications de prédiction ou de classification des séries temporelle ou des séquences vidéo où les données sont fort liées à la caractéristique du temps [15].
- d. **Réseaux de neurones génératifs (GAN, Generative Adversarial Networks)** : ces réseaux permettent de générer des données (images) avec un fort degré de réalisme en utilisant deux réseaux mis en concurrence « le générateur et le discriminateur ». Le générateur permet de générer un échantillon d'images tandis que le discriminateur a comme objectif de différencier entre images réelles et images générées. Ce type de

réseaux est fort utilisé pour la détection des faux visages (fake faces) fausse informations (fake news) [16].

Notons que dans le cadre de travail de fin d'études, nous utiliserons les réseaux de neurones convolutions CNN afin de classifier les images et localiser les œufs et poules dans notre système de poulailler intelligent qui capture les images à partir de la caméra.

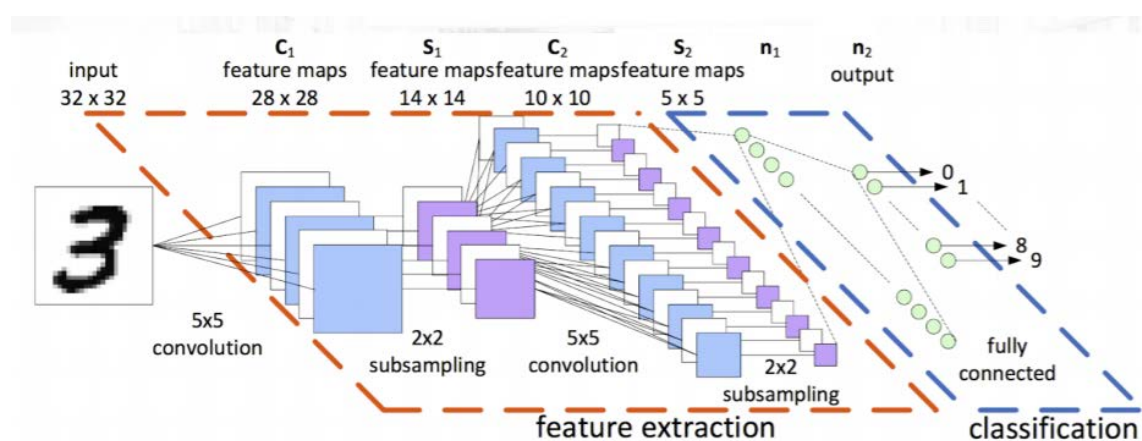


Figure 11 : exemple de réseau de neurone convolutionnel CNN ¹⁰

6. Intelligence artificielle et IoT :

La section précédente a démontré l'intérêt des données (annotées ou non) dans le domaine de l'intelligence artificielle et plus particulièrement dans les approches de Machine et Deep Learning ou les données représentent le cœur de la solution. D'autre part, le chapitre 1 nous a montré le rôle de la collecte des données dans un systèmes IoT. Nous pouvons constater que les données représentent un point commun entre les domaines de l'IoT et l'IA où :

- **L'IoT** : permet de collecter suffisamment de données en temps réel à partir de sources diverses ;
- **L'IA** : permet d'exploiter ces données par l'extraction de nouvelles connaissances à l'aide de modèles de classification ou de régression (prédiction).

Mais pour faire une utilisation commune de ces technologies, il faudra veiller à plusieurs critères tels que :

- **La latence** : nécessaire pour transférer les données collectées à partir de capteurs IoT vers des infrastructures cloud permettant d'entraîner et tester des modèles IA ;
- **La performance** : afin de fournir des temps d'entraînement et test des modèles rapides, il faudra utiliser des ressources de calcul à haute performances tels que les processeurs

¹⁰ <https://www.pyimagesearch.com/2014/06/09/get-deep-learning-bandwagon-get-perspective/>

graphiques GPU offrant un grand nombre de processeurs et par conséquent un calcul parallèle, rapide et moins couteux en énergie ;

- **Le volume** : des données qui ne cessent d'augmenter suite à la diversité des capteurs ce qui complique et ralentit le processus de transmission des données vers des ressources de calcul sur cloud.

Pour faire face à toutes ces contraintes, un nouveau domaine a vu le jour qui est le « **Edge Computing** », et qui propose une alternative au Cloud. L'idée est de déplacer le calcul vers des ressources proches des capteurs de collecte de données ce qui permettra d'éviter le besoin de transférer les données vers le cloud, et aussi de permettre une analyse des données en temps réel. Certaines données (ne nécessitant pas un traitement temps réel ou moins confidentielles) pourront être transférées vers le cloud pour appliquer un traitement ultérieur (non-temps réel).

L'Edge Computing présente plusieurs avantages [17, 18] :

- Réduction de la quantité de données transférée vers le cloud ;
- Communication rapide entre capteurs IoT et ressources Edge ;
- Confidentialité : grâce aux traitement local de données sans devoir les transférer au cloud.

Par ailleurs, il faudra avoir des ressources Edge permettant d'offrir un calcul suffisamment rapide et plus particulièrement lorsqu'il s'agit d'applications en intelligence artificielle connues par leur grande intensité de calcul. Généralement, les modèles sont entraînés dans le cloud à partir de données précédemment collectées puisque l'entraînement des modèles est une tâche très gourmande en calcul et en mémoire, ce qui rend quasiment impossible sa portabilité sur ressources Edge. Cependant, l'inférence ou le test peut être exécutée sur des ressources Edge qui contiennent déjà un modèle IA entraîné précédemment sur le cloud.

Parmi les ressources Edge, on peut citer les appareils mobiles tels que les smartphones, les Raspberry Pi¹¹ ou encore les cartes Jetson de Nvidia¹². Dans ce travail, nous avons choisi de travailler avec la carte Jetson Nano pour les raisons suivantes :

- Carte embarquée équipée d'un processeur CPU quad-core et d'un processeur graphique GPU à 128 cœurs CUDA ;
- Carte équipée d'un système d'exploitation semblable à Linux facilitant sa programmation ;
- Prix d'environ 90 euros qui est largement inférieur aux autres ressources du même type ;
- Présence de plusieurs ports de connexion tels que les ports USB permettant de connecter la carte à une caméra qui capturera la scène du poulailler.

¹¹ Raspberry Pi. <https://www.raspberrypi.org/>

¹² Nvidia Jetson. <https://developer.nvidia.com/embedded/jetson-modules>

Ces caractéristiques permettent à cette carte d'être bien adaptée à l'une des problématiques de notre système de poulailler intelligent : la localisation et le comptage de poules et œufs.

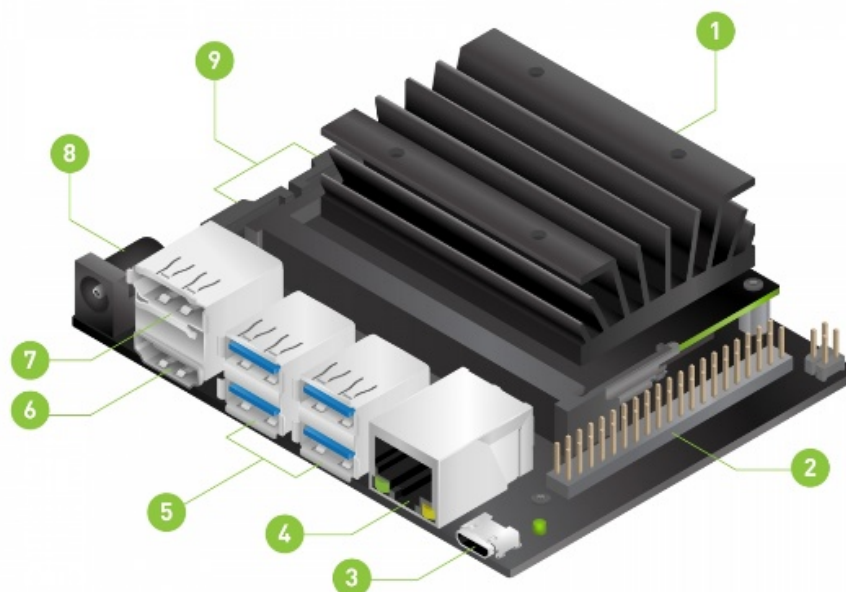


Figure 12 : ressource Edge IA "Jetson Nano" ¹³

Conclusion :

Dans ce chapitre, nous avons présenté les principes de base de l'intelligence artificielle et plus particulièrement les approches d'apprentissage machine et automatique (Machine et Deep Learning). La dernière partie de ce chapitre a permis de mettre en avant la relation entre les domaines d'intelligence artificielle et celui de l'internet des objets, ce qui nous a permis de sélectionner une ressource Edge IA sont mieux adaptés au traitement des données issues de la caméra pour notre problème de poulailler intelligent.

¹³ Jetson Nano. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

CHAPITRE III

Approche proposée

Introduction :

Après avoir vu et compris les principes fondamentaux des domaines d'internet des objets et d'intelligence artificielle, nous proposons une approche combinant les deux technologies (IA et IoT) pour résoudre notre problème de poulailler intelligent (Fig. 13). Ce chapitre décrit les étapes de de la réalisation du prototype de poulailler intelligent.

Dans ce projet nous nous sommes intéressés à la réalisation d'un poulailler intelligent dont la tâche principale est d'assurer le bien-être et la sécurité des poules. D'une part nous utiliserons les technologies IoT à l'aide d'une série de capteurs connectés pour acquérir les variables environnementales nécessaires à la prise de décision dans le poulailler. Aussi, des actionneurs permettront d'agir sur les conditions environnementales du poulailler. Les valeurs collectées seront affichées sur un écran LCD, elles sont également disponibles sur une interface web. D'autre part, une caméra sera également raccordée à un micro-ordinateur « carte Jetson Nano » en vue de déterminer l'activité et les comportements des volatiles. De plus l'analyse des images permettra de compter le nombre de poules grâce à un algorithme d'intelligence artificielle.

Grâce à cette collecte et interprétation d'informations, le poulailler est capable de gérer intelligemment la fermeture et l'ouverture de la porte principale à des heures différentes suivant les saisons. Le choix et les caractéristiques de chaque composant utilisé sont donnés ci-dessous. Le reste du chapitre mettra en avant la description globale du matériel et projet accompagné d'un schéma bloc de son fonctionnement

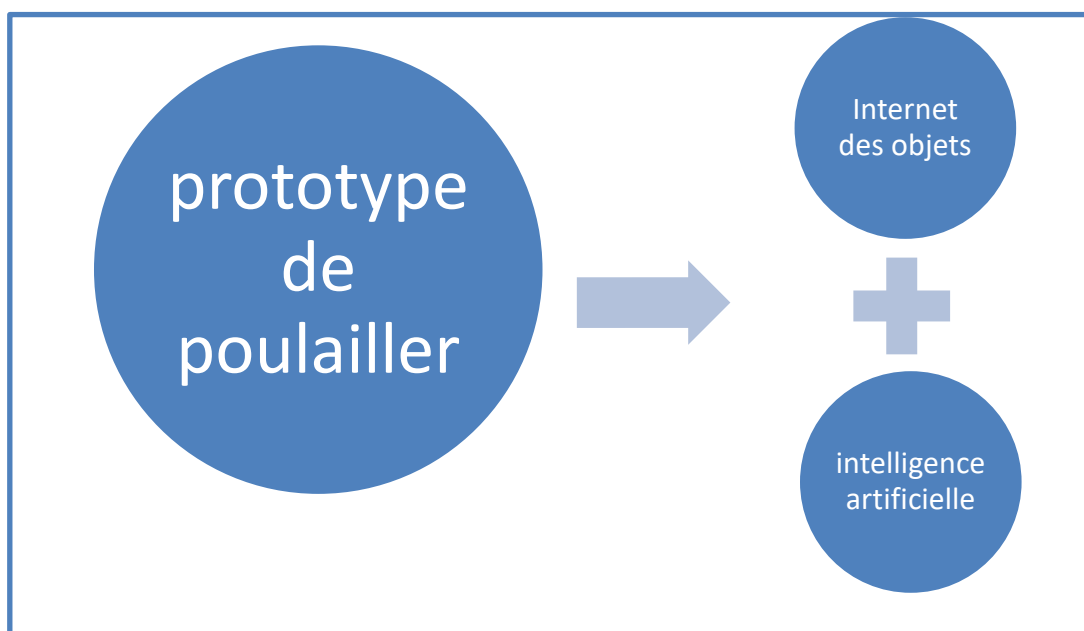


Figure 13 : structure du poulailler

Cette section est présentée en trois parties :

1. Description du matériel utilisé (capteurs d'acquisition IoT sélectionnés, micro-ordinateurs, etc.)
2. Présentation schématique de notre système de poulailler intelligent
3. Résultats expérimentaux :
 - 3.1. Partie 01 : dispositif d'acquisition des données environnementales
 - 3.2. Partie 02 : dispositif Edge IA d'acquisition et traitement de données visuelles

1. Description du matériel utilisé

1.1. La carte Arduino : c'est une carte électronique créée par une équipe de développeurs italiens, basée sur le microcontrôleur ATMEL. C'est un outil qui permet de créer des systèmes électroniques plus ou moins complexes. Elle est programmée en langage Arduino C.

1.2. NVIDIA Jetson Nano: Le kit de développement NVIDIA® Jetson Nano est un mini-ordinateur puissant qui permet d'exécuter plusieurs tâches en parallèles y compris des réseaux de neurones pouvant être utilisés dans différentes applications telles que la classification d'images, la détection d'objets, la segmentation et le traitement de la parole. Cette plateforme est facile d'utilisation grâce à son système for semblable à Linux, elle consomme très peu d'énergie (environ 5 watts). La carte « Jetson Nano » est très utile dans le domaine de l'intelligence artificielle et plus particulièrement d'apprentissage profond (Deep Learning) car elle possède toutes les fonctionnalités nécessaires pour développer des projets d'IA grâce aux composants suivants (Fig. 12) :

- Emplacement pour carte microSD (stockage principal)
- Connecteur d'extension à 40 broches
- Port micro-USB pour une entrée d'alimentation 5 V.
- Port Ethernet Gigabit
- 4 Ports USB 3.0
- Port de sortie HDMI
- Connecteur DisplayPort
- Barrel jack pour une entrée d'alimentation 5V
- Connecteurs de caméra MIPI CSI-2.

1.3. ESP32 : Le module WiFi contenant le microcontrôleur ESP32 peut être programmé comme une carte Arduino, et peut communiquer en Wifi avec d'autres appareils (ordinateurs, Smartphones, etc.). ESP32 est une série de systèmes de microcontrôleur sur puce (Soc) à faible coût et à faible consommation avec fonctions WIFI et Bluetooth, cette puce est Dual Core. Il offre deux unités de traitement 32 bits qui fonctionnent à une fréquence plus rapide. Ce module utilise le protocole de communication WIFI 802.11 b/g/n, qui permet au microcontrôleur interne de se connecter à un réseau Wi-Fi et de faire des connexions TCP/IP simples, ou d'être détectable par d'autres périphériques qui utilisent le même protocole WIFI.

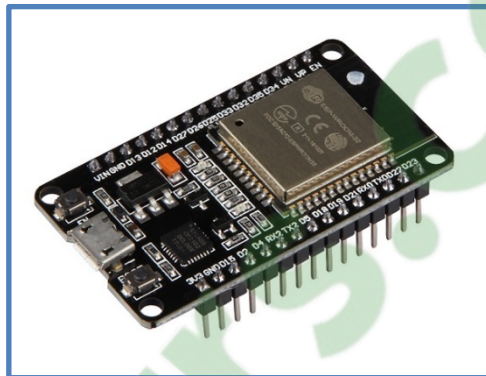


Figure 14 : Le module ESP32

▪ **Caractéristiques du module ESP32 :**

- ESP32 est dual core, cela signifie qu'il dispose de 2 processeurs.
- Wifi et Bluetooth intégré.
- Exécutions des programmes de 32 bits.
- La fréquence d'horloge jusqu'à 240 MHz.
- 512 Ko de mémoire vive (RAM) [24].

▪ **ESP32 PINS :** le module ESP32 dispose d'un total de 32 E/S, dont 26 numériques et 18 analogiques. L'écran tactile peut également être utilisé pour créer une interface homme-machine basée sur ESP32.

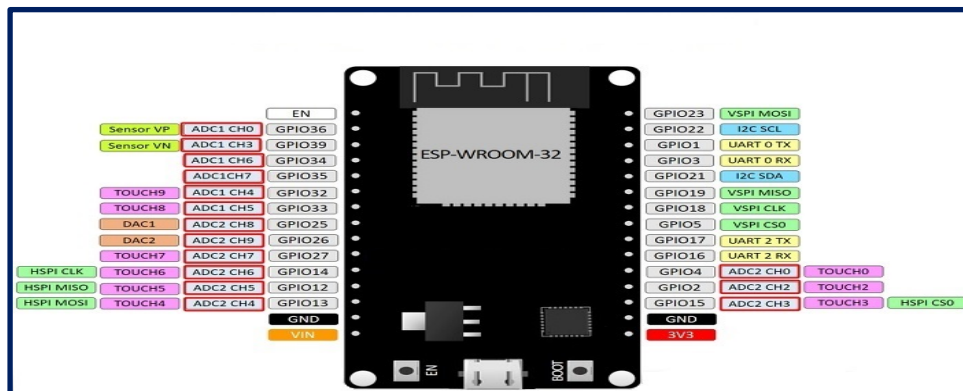


Figure 15 : repérage des broches des modules ESP32

1.4. Les capteurs : les capteurs représentent des dispositifs qui envoient des informations sur l'environnement, ils convertissent les informations physiques en signaux électriques. Ces derniers sont les éléments de base des systèmes d'acquisition de données (Fig. 16).

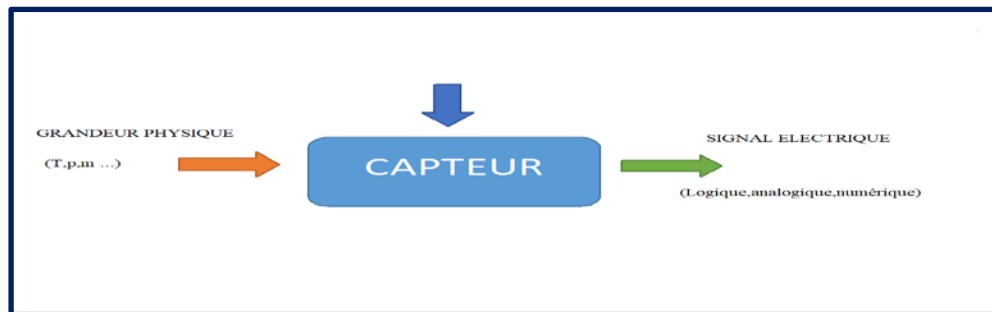


Figure 16 : fonctionnement du capteur

Afin de mener à bien notre projet de poulailler, nous avons sélectionné les capteurs suivants :

1.4.1. DHT11 : c'est un capteur de température et d'humidité, facile à utiliser et à faible coût. Il n'a besoin que d'une résistance de tirage et d'une alimentation 3V ou 5V pour fonctionner. Il est facile de programmer à l'aide de l'Arduino. Ce capteur peut mesurer des températures de 0 à + 50 °C avec une précision de +/- 2 °C, un taux d'humidité relative de 20 à 80% et une précision de +/- 5%. La mesure peut être effectuée une fois par seconde. Les Caractéristiques de DHT11 sont (Fig. 17) :

- Alimentation : 3-5 V
- Signal de sortie : Signal numérique
- Plage de mesure : Humidité 20-80%RH ; Température : 0-50°C
- Précision : Humidité +/- 5%RH ; Température : +/- 2 °C
- Résolution : Humidité 1%RH ; Température : 0.1°C
- Fréquence de mesure : 1Hz (1 mesure par seconde)
- Stabilité à long terme : +/- 1% RH/an
- Dimension : 12x15.5x5.5

Le brochage du capteur DHT11 est représenté par :

- La broche n°1 : broche d'alimentation
- La broche n°2 : broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms.
- La broche n°3 n'est pas utilisée et ne doit pas être câblée.
- La broche n°4 : masse du capteur (GND).

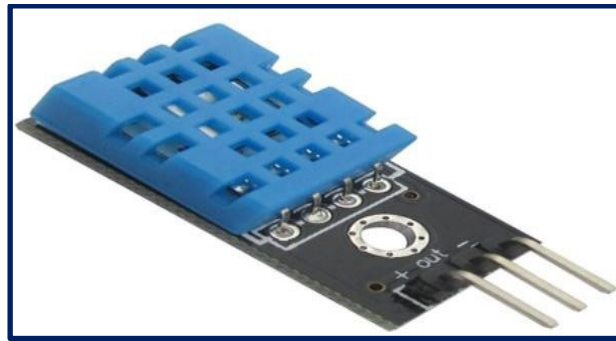


Figure 17 : le capteur DHT11

La communication avec le DHT11 : le micro-ordinateur réveille le capteur en mettant la ligne de données à l'état bas pendant au moins 18 ms. Pendant ce temps, le capteur se réveillera et se préparera à mesurer la température et l'humidité. Une fois le temps écoulé, le contrôleur libère la ligne de données et continue à surveiller. Après avoir libéré la ligne de données, le capteur répond au microcontrôleur en gardant la ligne de données basse pendant 80 μ s, puis haute pendant 80 μ s (indiquant qu'il est prêt à prendre une mesure).

Ensuite, le DHT11 enverra une série de 40 bits (5 octets), les deux premiers octets contiennent la valeur de mesure d'humidité. Les deux octets suivants contiennent la mesure de la température et le cinquième octet contient la somme de contrôle permettant de vérifier que les données lues sont correctes (Fig. 18).

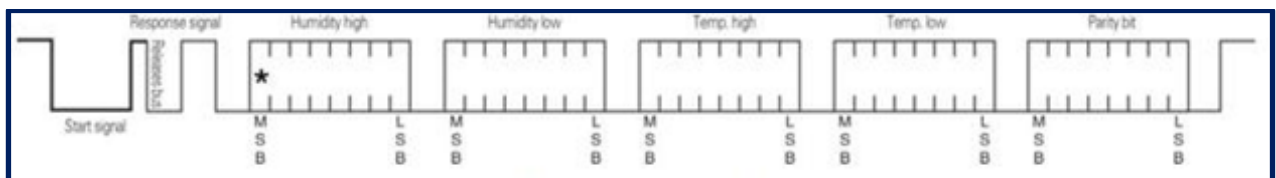


Figure 18 : format de trame de communication de DHT11

1.4.2. Les capteurs MQ-xxx : Les capteurs MQ sont des capteurs physicochimiques, utilisés pour détecter divers gaz, polluants et fumées dans l'atmosphère. Avant de fonctionner et de fournir des valeurs de mesure correctes, certains capteurs doivent être préchauffés. Nous utilisons deux capteurs de cette série : le MQ-135 et le MQ-136.

1.4.2.1. MQ-136 : le capteur de gaz MQ-136 détecte la concentration de H₂S dans l'air et affiche le résultat comme une tension analogique. La concentration de détection varie de 1 ppm à 200 ppm. Le capteur peut fonctionner à des températures allant de -10 à 50°C et consomme moins de 180 mA à 5 V) [18]. Il est utilisé dans les équipements de contrôle de la qualité de l'air pour les bâtiments et les bureaux

Les caractéristiques du capteur MQ-136 sont :

- Gaz de détection : sulfure d'hydrogène
- Plage de détection : 1-200ppm
- Tension de fonctionnement : DC 5V
- Courant de travail : 180mA
- Tension de sortie : 0-5V
- Air pur : tension $\leq 1.5V$
- Sensibilité : $\geq 3\%$
- Résistance de charge : 10K (réglable)
- Consommation d'énergie des composants : $\leq 0,9 W$
- Température de fonctionnement : -10 à 50 ° C (température nominale 20 ° C)
- Humidité de fonctionnement : 95% HR (humidité nominale 65% HR)
- Temps de réponse : $\leq 1S$
- Préchauffage : 24 heures



Figure 19 : Le capteur de gaz MQ136

1.4.2.2. MQ-135 : le capteur de gaz MQ-135 est sensible au CO₂, à l'alcool, au Benzène, à l'oxyde d'azote (NO_x) et à l'ammoniac (NH₃), etc. Le MQ-135 utilise le dioxyde d'étain (SnO₂) dont la conductivité électrique varie en fonction de la présence des polluants. Il est adapté à différentes applications et caractérisé par sa longue durée de vie et le faible coût [21].

Les pins de MQ-135 sont :

- VCC ↔ 2.5V ~ 5.0V
- GND ↔ Ground
- AOUT ↔ Analog output
- DOUT ↔ Digital output



Figure 20 : Le capteur de gaz MQ135

2. Présentation schématique de notre système de poulailler intelligent

En raison de l'impossibilité d'ESP32 à la carte « Jetson Nano » (pour des raisons purement matérielles), nous avons développé deux systèmes séparés. Le premier utilise les capteurs DHT11, MQ135, MQ136 et MQ137 pour surveiller en temps réel les informations de température, humidité, quantité de gaz et d'ammoniac. Ces informations serviront à manipuler automatiquement le système de ventilation afin de garantir les meilleures conditions de vie (Fig. 21). Le deuxième système utilise les images collectées à l'aide d'une caméra USB qui sont ensuite analysées avec la carte Jetson Nano pour détecter et compter le nombre d'œufs et de poules. Le comptage du nombre de poules permettra de savoir si toutes sont rentrées et par conséquent fermer automatiquement la porte (Fig. 22).

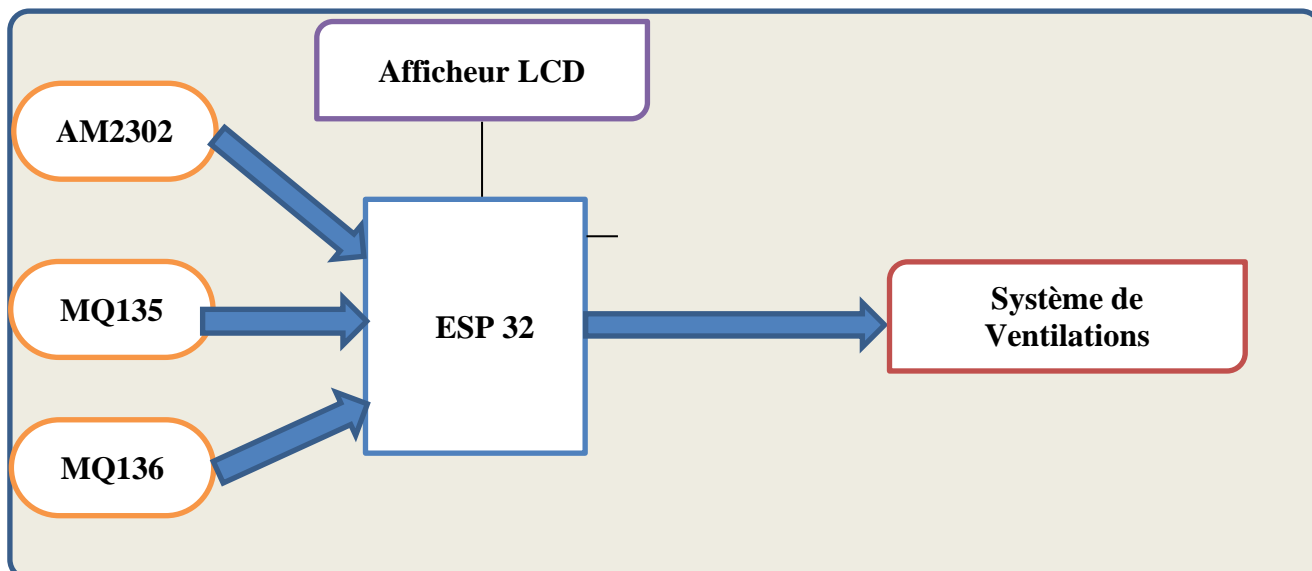


Figure 21 : schéma de fonctionnement du poulailler intelligent à l'aide de l'ESP32 et capteurs

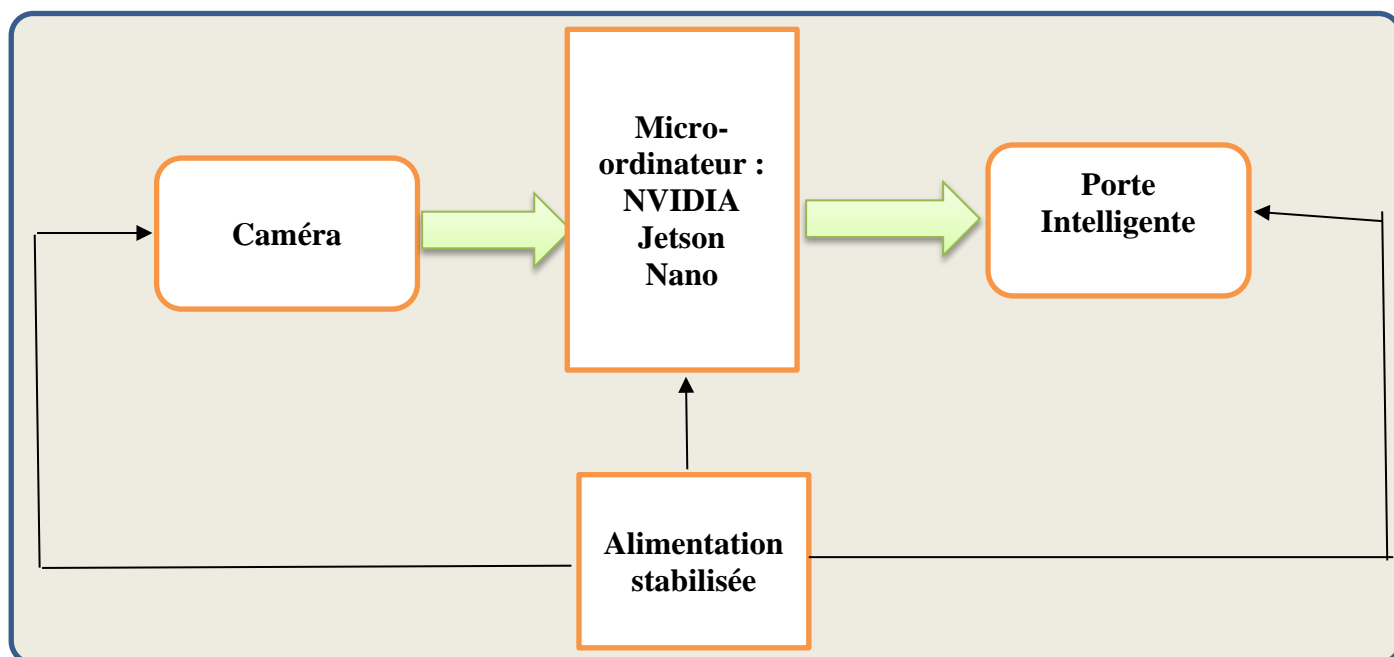


Figure 22 : schéma du poulailler intelligent à l'aide de la Jetson Nano et caméra

3. Résultats expérimentaux

3.1. Partie 01 : dispositif d'acquisition des données environnementales

L'élément central du dispositif concret implémenté est un microcontrôleur Expressif esp32 qui est connectable en Bluetooth et en Wi-Fi. Ce microcontrôleur à faible coût a été sélectionné d'une part pour son coût modique, sa connectique (I2C, I2S, digitale et analogique) et d'autre part pour sa facilité de programmation à l'aide d'Arduino IDE dans une version allégée du C (C-like). Il supporte également la plupart des bibliothèques d'Arduino et son écosystème de capteurs ce qui facilite l'implémentation de différents types de capteurs.



Figure 23 : ESP32 Expressif

Les capteurs que nous avons décidé de connecter au microcontrôleur en analogique sont : le MQ135 qui mesure la teneur en CO_2 dans l'air, le MQ136 qui est un capteur qui mesure la teneur des composés sulfuré, principalement le H_2S . Un troisième capteur est l'AM2302 qui est connecté sur une entrée digitale et qui mesure la température avec une précision d'un demi-degré et l'humidité relative avec une précision de 3%.



Figure 24 : De gauche à droite, le MQ135, MQ136 et AM2302

Les capteurs MQ135 et MQ136 sont alimentés avec une tension de 5V DC, mais les entrées digitales n'admettent des valeurs que sur 3V3 DC. Nous avons donc dû diviser les tensions mesurer pour les ramener de 5V en 3V3 et être compatible avec les valeurs mesurables par le microcontrôleur grâce à un pont diviseur.

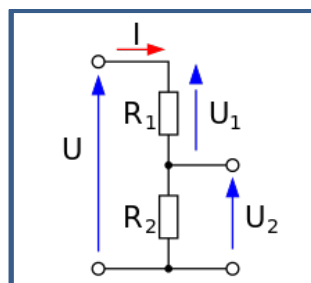


Figure 25 : Pont diviseur

Les valeurs de R_1 et de R_2 sont respectivement de $17\text{k}\Omega$ et $33\text{k}\Omega$. U vaut 5V et U_1 : 3V3 et U_2 : 1V7. Les pins data des capteurs MQ135 et MQ136 sont respectivement reliés aux pin G34 et G35 de l'ESP32. Ces pins disposent d'un convertisseur analogique digital : ADC. Le capteur

AM2302 est alimenté en 3V3 et fourni une sortie digitale sur 3V3. Une résistance pull-up de 1KΩ a été intégré entre la sortie data et le +3V3. La sortie data est reliée à la sortie G4 de l'ESP32. Le contrôle de ces paramètres est essentiel pour le bien être des volatiles et le maintien de ces paramètres dans un intervalle de valeur correcte assure une productivité optimale des volailles. Un écran LCD I2C est alimenté en 5V permet de voir l'adresse IP acquise lors de la connexion sur le réseau Wi-Fi et ensuite de visualiser l'évolution de la température, de l'humidité relative, du taux de CO₂ et du taux de NH₃.



Figure 26 : Valeurs des capteurs sur l'écran LCD I2C

Les valeurs des capteurs peuvent également être consultées via la consultation de la page web délivrée par le serveur web hébergé sur l'ESP32.

| MEASUREMENT | VALUE |
|---------------|----------|
| Temp. Celsius | 21.00 °C |
| Humidity | 44.40 % |
| CO2 | 2374 ppm |
| Ammoniac | 494 ppm |

Figure 27 : Serveur web hébergé sur l'ESP32.

Notons que le dispositif est alimenté par une adaptateur secteur 220V – 5VDC 1A. Le code de cette partie se trouve en annexe 1

3.2. Partie 02 : dispositif Edge IA d'acquisition et traitement de données visuelles

Nous avons fait le montage décrit dans la figure 21 regroupant une carte Jetson Nano et une caméra USB tous les deux alimentés à l'aide d'un chargeur USB C (similaire à celui qu'on utilise pour recharger les smartphones) branché à la Jetson Nano. La caméra est alimentée depuis la carte Jetson nano via le port USB. Grâce à ce dispositif, nous pouvons collecter les images en temps réel à l'aide d'une librairie de traitement d'image « OpenCV » installée sur la carte Jetson Nano. Le langage de programmation utilisé pour cette partie est Python car ce dernier est le plus utilisé dans le domaine de l'IA. Nous avons ensuite développé trois fonctions :

3.2.1. Détection et comptage des œufs : pour chaque image de la vidéo, nous appliquons une méthode de détection de contours, fournie par « OpenCV ». Cette méthode s'appuie sur le détecteur de Canny implémenté par les fonction « Canny » et « findContour » permettant de détecter les contours séparément. Pour savoir que le contour détecté correspond à un œuf, on calcul sa taille à l'aide de la fonction « contourArea ». Si elle dépasse un seuil, on distingue qu'il s'agit d'un œuf et il nous rester plus qu'à compter le nombre de contours restants. La Fig. 27 illustre le résultat de cette détection. Le code de cette partie se trouve en annexe 2.

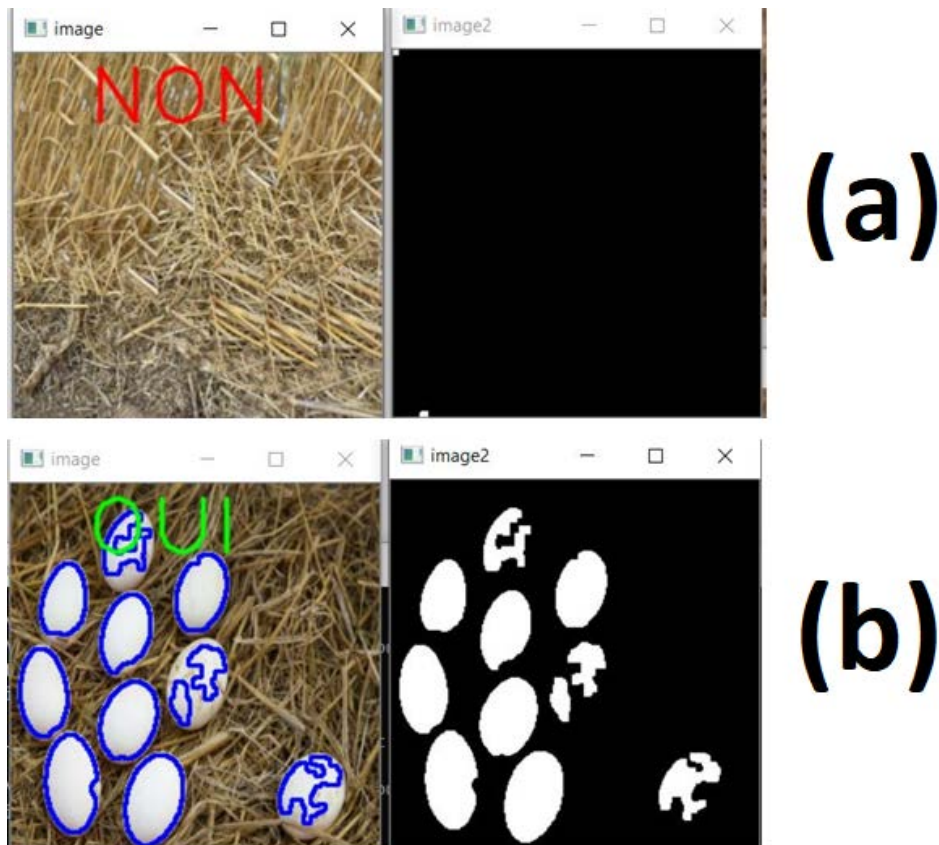


Figure 28 : (a) détection de 0 œufs. (b) détection de 10 œufs

3.2.2. Détection et comptage des poules : la détection des poules est plus compliquée que les œufs en raison de leur forme complexe et variant d'une poule à l'autre. Afin de garantir une bonne précision, nous avons pris le choix de travailler avec un réseau de neurones profond permettant la localisation d'objets tel que Yolo¹⁴. Ce dernier permet de détecter et localiser en temps réel différents types d'objets (Fig. 28).

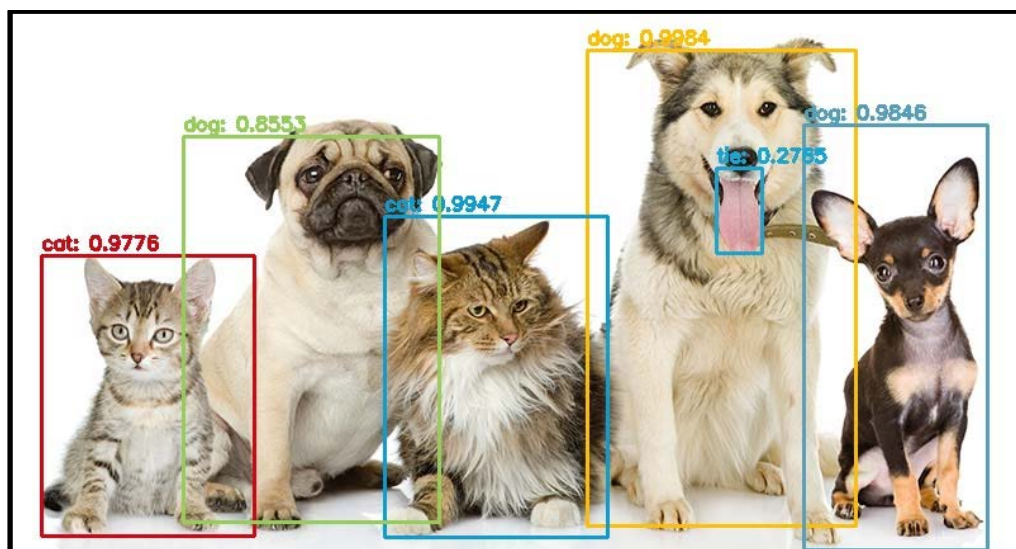


Figure 29 : exemple de localisation d'animaux à l'aide du réseau de neurones profond Yolo¹⁵

Afin d'exploiter ce réseau de neurones pour le comptage des poules, nous suivons trois phases :

- Collecte d'une base de données annotée :** nous avons collecté une base contenant 200 images de poules que nous avons téléchargées à l'aide de Google Images.
- Annotation des images :** afin d'exploiter un réseau de neurones de localisation, il a fallu annoter chaque image en précisant la présence ou pas d'une poule et aussi les positions (coordonnées) où se trouvent ses poules. Nous avons utilisé l'outil LabelMe¹⁶ pour réaliser cette annotation.
- Téléchargement du modèle de base de Yolo :** nous avons récupéré un modèle pré-entraîné de Yolo permettant de détecter différents types d'objets (voiture, personne, chat, chien, vélo, etc.).
- Réentraînement du modèle avec notre base d'images de poules :** étant donné que le modèle téléchargé ne permet pas de détecter les poules, il a fallu réentraîner Yolo avec notre base d'images pour espérer reconnaître et compter les poules. Malheureusement, l'entraînement de ce type de réseau nécessite une ressource de calcul à haute performance

¹⁴ Yolo. <https://pjreddie.com/darknet/yolo/>

¹⁵ Yolo. <https://pjreddie.com/darknet/yolo/>

¹⁶ LabelMe. <https://github.com/wkentaro/labelme>

(GPU) pour permettre au modèle de passer plusieurs itérations. Cette tâche représente l'une des premières perspectives de ce travail.

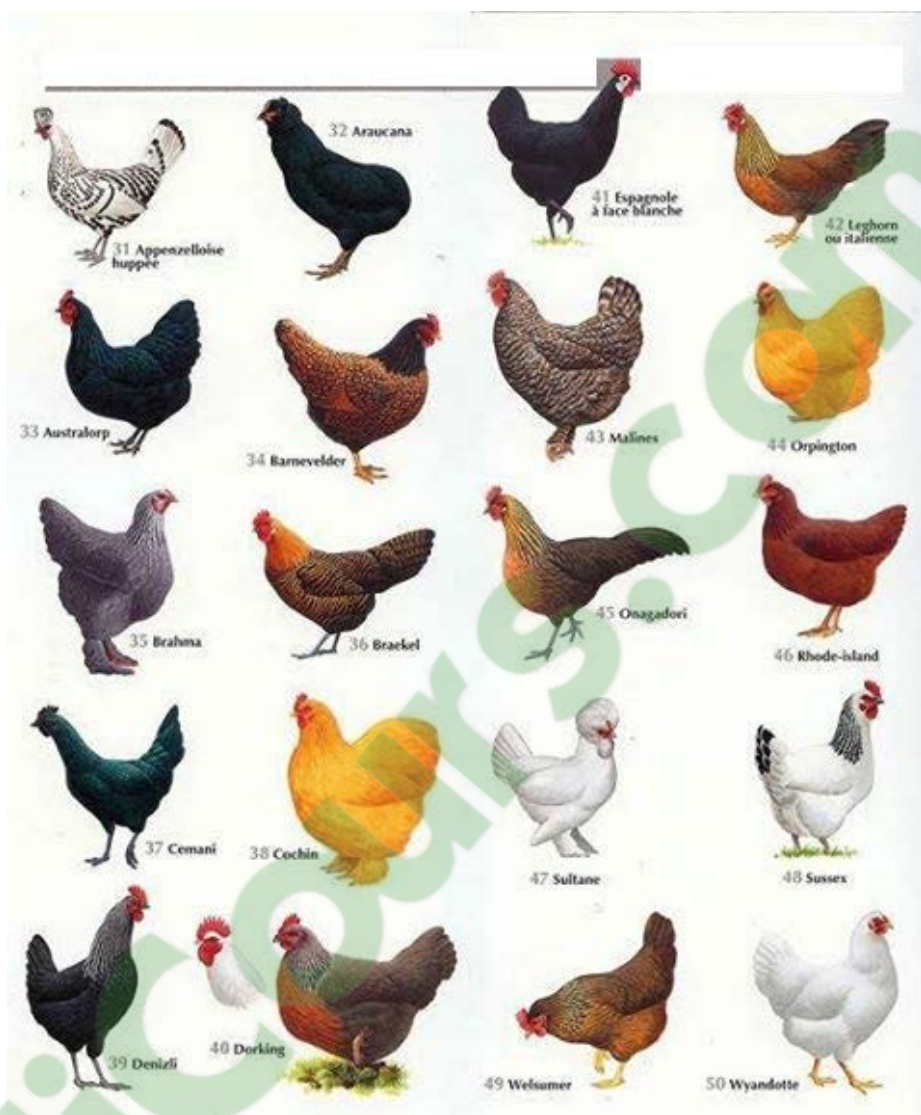


Figure 30 : exemples d'images de poules collectées à partir de Google Image

CONCLUSION

Conclusion et Perspectives

Ce travail de fin d'études m'a permis d'explorer les domaines d'internet des objets (IoT) et d'intelligence artificielle (IA) dans le cadre du développement d'un système de poulailler intelligent. Malgré la pandémie Covid-19 qui a causé un délai considérable pour la réception des capteurs et matériaux commandés (qu'on a commencé à recevoir seulement en mois de Septembre 2020), nous avons tout de même pu atteindre les objectifs du travail de fin d'étude en réalisant :

- Etude théorique du domaine de l'IoT qui a permis d'identifier les matériaux et capteurs les mieux adaptés à notre problématique.
- Etude théorique du domaine de l'IA qui a permis d'identifier les différents algorithmes à appliquer pour détecter et compter les œufs (et poules prochainement)
- Développement d'un système temps réel de collecte d'informations de température, humidité, quantité de gaz et ammoniac à l'aide de l'ESP32 et différents capteurs.
- Développement d'une solution IA embaquée sur la ressource Edge IA « Jetson Nano » pour détecter et compter les œufs.

Comme perspectives, nous envisageons de développer les points suivants :

- Combiner les deux systèmes utilisant l'ESP32 et la Jetson Nano en un seul système ce qui nécessite le changement de configuration matérielle ;
- Finalisation de l'algorithme de détection et comptage des poules en lançant les entraînements à l'aide ressources GPU avec des bases de données plus grandes.
- Développement du système de contrôle automatique des portes et ventilateurs en fonction des résultats collectées et calculées à partir de nos deux dispositifs décrits plus haut.

Bibliographie :

- [1] Pierre-Jean Benghozi, Sylvain Bureau et Françoise Massit-Folléa, L'Internet des objets, Edition MSH.
- [2] Yahi amina, Kouri Loubna, "contrôle et suivi d'une maison intelligente via internet", Mémoire de Master en Electronique, Université de bouira, Algérie, 2018.
- [3] La NFC, comment Ça marche ? [en ligne], (consulté le : 22 mai 2020), disponible sur le lien : <https://www.paragon-rfid.com/la-nfc-comment-ca-marche/>
- [4] Bluetooth, [en ligne], consulté le : 22 mai 2020, disponible sur le lien : <https://www.futura-sciences.com/tech/definitions/electronique/bluetooth-16527/>
- [5] IPv6 : Tout savoir sur ce nouveau standard Internet [en ligne], consulté le : 20 Mars 2020, disponible sur le lien : <https://www.ionos.fr/digitalguide/serveur/know-how/quels-sont-les-avantages-de-ipv6/>
- [6] 5 raisons pour lesquelles la sécurité de l'IoT est difficile [en ligne], consulté le : 30 Mars 2020) disponible sur : <https://www.xerox.com/fr-cg/infos-complementaires/securite-iot>
- [7] Y.Ait Mouhoub et F.Bouchebbah. Proposition d'un modèle de confiance pour l'Internet des objets. Mémoire master de l'université Abderrahmane Mira Bejaia, 21 Juin 2015.
- [8] Quelle est la place de la gestion de l'alimentation dans l'IoT [en ligne], consulté le : 31 Mars 2020, disponible sur : <https://www.rs-online.com/designspark/quelle-est-la-place-de-la-gestion-d'alimentation-dans-liot-fr>
- [8] Russel, S. Et Norvig, P., « Artificial Intelligence : A Modern Approach » 3rd edition, Pearson, 2010.
- [9] A. M. Turing. « Computing Machinery and Intelligence ». *English. In : Mind. New Series* 59.236 (1950), p. 433-460
- [10] Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." *Neural networks for perception*. Academic Press, 1992. 65-93.
- [11] S. Kotsiantis. "Supervised machine learning: A review of classification techniques". *Informatica Journal*, 31 :249–268 (2007)
- [12] R. O. Duda et al, "Pattern Classification", chapter : Unsupervised Learning and Clustering. Wiley Inter science (2001)
- [13] O. Chapelle, "Semi-supervised Learning". MIT Press (2006) [5] R. Sutton et al, "Reinforcement Learning - An Introduction", MIT Press (2012)
- [14] Yan, L. C., B. Yoshua, and H. Geoffrey. "Deep learning." *nature* 521.7553 (2015): 436-444.
- [15] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9 (8), 1735–1780.
- [16] Hitaj, Briland, Giuseppe Ateniese, and Fernando Perez-Cruz. "Deep models under the GAN: information leakage from collaborative deep learning." *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017.
- [17] J. Chen et X. Ran. « Deep Learning With Edge Computing : A Review ». *In : Proceedings of the IEEE PP (juil. 2019)*, pp. 1-20, (cf. p. 12, 13).
- [18] M. G. Sarwar Murshed et al. « Machine Learning at the Network Edge : A Survey ». *In : arXiv e-prints, arXiv :1908.00080 (juil. 2019), arXiv :1908.00080. arXiv : 1908.00080 [cs.LG] (cf. p. 13).*
- [19] MQ-136 datasheet [en ligne], consulté le : 20 août 2020, disponible sur : <http://www.sensorica.ru/pdf/MQ-136.pdf>
- [20] MQ-137 datasheet [en ligne], (consulté le : 20 août 2020), disponible sur : <http://www.sensorica.ru/pdf/MQ-137.pdf>

- **[21]** MQ-135 datasheet [en ligne], consulté le : 20 août 2020, disponible sur : <http://www.sensorica.ru/pdf/MQ-135.pdf>
- **[22]** Bluetooth, [en ligne], (consulté le : 22 mai 2020), disponible sur le lien : <https://www.futura-sciences.com/tech/definitions/electronique-bluetooth-16527/>
- **[23]** Réseau IOT : De quoi s'agit-il ? comment fonction un réseau IOT ? [en ligne], (consulté le : 31 Mars 2020), disponible sur le lien : <https://www.djvuzone.org/reseau-iot-de-quoi-sagit/>
- **[24]** the internet of things with ESP32[en ligne], (consulté le : 20 août 2020), disponible sur le lien : <http://esp32.net/>

ANNEXE1

Code implémenté sur l'ESP32.

```
#include <WiFi.h>
#include <Wire.h>
#include "DHT.h"
#include <LiquidCrystal_I2C.h>

int lcdColumns = 20;
int lcdRows = 4;

LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);

// Uncomment one of the lines below for whatever DHT sensor type you're using!
// #define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT21 // DHT 21 (AM2301)
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// DHT Sensor;
uint8_t DHTPin = 4;
DHT dht(DHTPin, DHTTYPE);
float Temperature;
float Humidity;
uint8_t MQ135Pin = 34;
int mq135Value = 0;
int mq136Value = 0;
uint8_t MQ136Pin = 35;
//const char* ssid = "sinfo-t580.umons.ac.be";
//const char* password = "Fs6k4JQg";
const char* ssid = "Mi A3";
const char* password = "8e5621032eaf";

WiFiServer server(80);

String header;

void setup() {
  Serial.begin(115200);
  pinMode(DHTPin, INPUT);
  dht.begin();

  lcd.init();
  lcd.backlight();

  lcd.setCursor(0,0);
  lcd.print("Connecting to Wi-Fi");
  Serial.print("Connecting to Wi-Fi Network");
  lcd.setCursor(0,1);
  lcd.print(ssid);
  Serial.println(ssid);
  WiFi.begin(ssid, password);
```

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("Successfully connected to WiFi.");
Serial.println("IP address of ESP32 is : ");
Serial.println(WiFi.localIP());
lcd.setCursor(0,3);
lcd.print(WiFi.localIP());
server.begin();
Serial.println("Server started");
delay(2000);
}

void loop()
{
  Temperature = dht.readTemperature();
  Humidity = dht.readHumidity();
  mq135Value = analogRead(MQ135Pin);
  mq136Value = analogRead(MQ136Pin);
  WiFiClient client = server.available();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Tmp: ");
  lcd.print(dht.readTemperature());
  lcd.print(" C");
  lcd.setCursor(0,1);
  lcd.print("Hum: ");
  lcd.print(dht.readHumidity());
  lcd.print(" %");
  lcd.setCursor(0,2);
  lcd.print("CO2: ");
  lcd.print(mq135Value);
  lcd.print(" ppm");
  lcd.setCursor(0,3);
  lcd.print("NH3: ");
  lcd.print(mq136Value);
  lcd.print(" ppm");

  if (client)
  {
    Serial.println("Web Client connected ");
    String request = client.readStringUntil('\r');
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println("Connection: close");
    client.println();
    client.println("<!DOCTYPE html><html>");
    client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
    client.println("<link rel=\"icon\" href=\"data:;\">");
    client.println("<style>body { text-align: center; font-family: \"Arial\", Arial;}");
  }
}

```

```

client.println("table { border-collapse: collapse; width:40%; margin-left:auto; margin-
right:auto;border-spacing: 2px;background-color: white;border: 4px solid green; }");
client.println("th { padding: 20px; background-color: #008000; color: white; }");
client.println("tr { border: 5px solid green; padding: 2px; }");
client.println("tr:hover { background-color:yellow; }");
client.println("td { border:4px; padding: 12px; }");
client.println(".sensor { color:red; font-weight: bold; background-color: #bcbbc; padding: 1px;
}");
client.println("</style></head><body><h1>ESP32 Web Server Reading sensor values</h1>");
client.println("<h2>AM2302 - MQ135 - MQ136</h2>");
client.println("<table><tr><th>MEASUREMENT</th><th>VALUE</th></tr>");
client.println("<tr><td>Temp. Celsius</td><td><span class=\"sensor\">");
client.println(dht.readTemperature());
client.println(" *C</span></td></tr>");
client.println("<tr><td>Humidity</td><td><span class=\"sensor\">");
client.println(dht.readHumidity());
client.println(" %</span></td></tr>");
client.println("<tr><td>CO2</td><td><span class=\"sensor\">");
client.println(mq135Value);
client.println(" ppm</span></td></tr>");
client.println("<tr><td>Ammoniac</td><td><span class=\"sensor\">");
client.println(mq136Value);
client.println(" ppm</span></td></tr>");
client.println("</body></html>");
client.stop();
client.println();
Serial.println("Client disconnected.");
Serial.println("");
}
delay(5000);
}

```

ANNEXE 2

1. Code implémenté pour l'acquisition d'une vidéo avec la Jetson nano et la caméra :

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

2. Code implémenté pour la détection des poules :

```
import cv2
import numpy as np
import glob
fichiers=glob.glob("images/*.jpg")
for fichier in fichiers:
    img=cv2.imread(fichier, 1)
    width = int(250)
    height = int(250)
    dim = (width, height)
    img = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    edges=~cv2.Canny(img, 80, 150)
    edges=cv2.erode(edges, None, iterations=3)
    edges=cv2.dilate(edges, None, iterations=2)
    presence=0
    elements=cv2.findContours(edges, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)[-2]
    i=0
    for e in elements:
        c=cv2.contourArea(e)
        if c>170:
            cv2.drawContours(img, [e], 0, (255, 0, 0), 2)
            presence=1
    if presence:
        cv2.putText(img, "OUI", (50, 50), cv2.FONT_HERSHEY_PLAIN, 4, (0,
255, 0), 2, cv2.LINE_AA)
    else:
        cv2.putText(img, "NON", (50, 50), cv2.FONT_HERSHEY_PLAIN, 4, (0,
0, 255), 2, cv2.LINE_AA)
    cv2.imshow("image", img)
    cv2.imshow("image2", edges)
    cv2.waitKey()
```