

# وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE MOHAMED SEDDIK BENYAHIA JIJEL

Faculté des sciences et de la technologie

Département de Electronique

N° :...../2023

## **MEMOIRE DE MASTER**

**DOMAINE : Sciences et Technologies**

**FILIERE : Electronique**

**SPECIALITE : Electronique des Systèmes Embarqués**

## **Thème**

**Application de l'apprentissage automatique embarqué (EML) et l'Internet  
des Objets (IoT) pour le diagnostic des générateurs photovoltaïques**

**Présenté Par : Zakaria Ksira**

**Encadré Par : Pr. A. Mellit**

**Youcef Kirat**

**Date de soutenance : 26/06/2023**

### **Jury de Soutenance**

**Président : M. Bouzerdoum**

**Grade : Docteur**

**Univ MSB jijel**

**Encadreur : A. Mellit**

**Grade : Professeur**

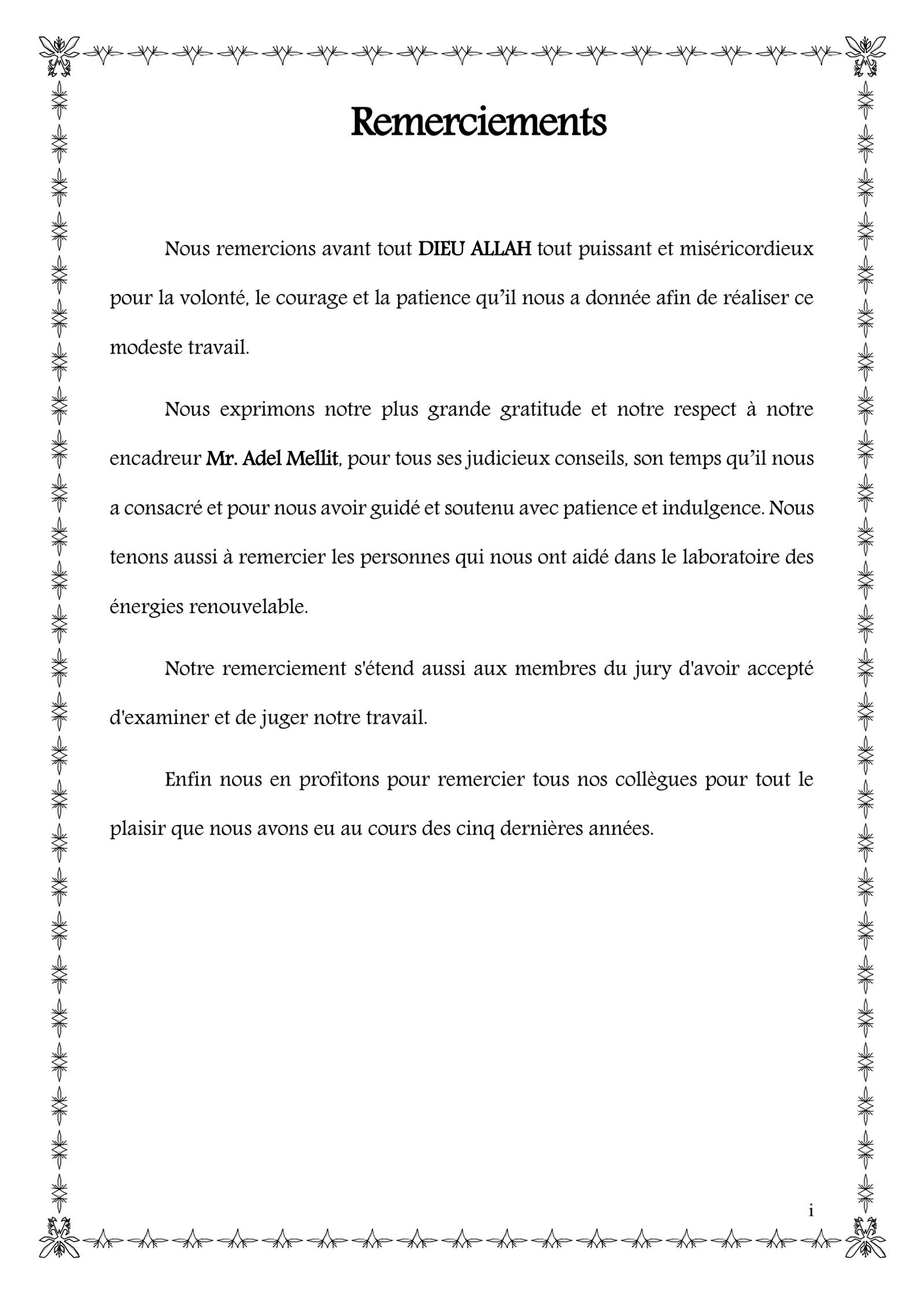
**Univ MSB jijel**

**Examineur : Z. Santouh**

**Grade : Docteur**

**Univ MSB jijel**

**Promotion : 2022 /2023**



# Remerciements

Nous remercions avant tout **DIEU ALLAH** tout puissant et miséricordieux pour la volonté, le courage et la patience qu'il nous a donnée afin de réaliser ce modeste travail.

Nous exprimons notre plus grande gratitude et notre respect à notre encadreur **Mr. Adel Mellit**, pour tous ses judicieux conseils, son temps qu'il nous a consacré et pour nous avoir guidé et soutenu avec patience et indulgence. Nous tenons aussi à remercier les personnes qui nous ont aidé dans le laboratoire des énergies renouvelable.

Notre remerciement s'étend aussi aux membres du jury d'avoir accepté d'examiner et de juger notre travail.

Enfin nous en profitons pour remercier tous nos collègues pour tout le plaisir que nous avons eu au cours des cinq dernières années.

A decorative border with repeating floral motifs surrounds the page. The motifs are stylized flowers and leaves, arranged in a continuous line along the top, bottom, and sides.

# Dédicaces

Je dédie ce modeste travail

A mes chers parents, ma mère et mon père  
pour tous leurs sacrifices, leur patience, leur amour,  
et leurs encouragements.

A mon cher frère et mes chères sœurs, pour leur appui  
et leur encouragement.

A mon binôme **Youcef** qui a partagé avec moi ce travail.

Sans oublier tous mes amis qui  
m'ont toujours encouragé.

**Zakaria**

# Dédicaces

Je dédie cet ouvrage

A ma chère mère qui m'a soutenu et encouragé durant mes années d'étude  
qu'elle trouve ici le témoin de ma grande reconnaissance.

A mon père qui m'a soutenu et encouragé tout au long de mon parcours.

A ma petite sœur qui a toujours su comment provoquer de la joie et du  
bonheur pour m'égayer dans les moments difficiles. Je profite de cette occasion  
pour lui souhaiter une bonne réussite à son examen de BAC : Incha-Allah.

A mes grands-parents qui m'ont été d'un soutien moral indéniable

A mes tantes ainsi qu'à mes oncles pour leurs encouragements sans oublier  
mes amis pour lesquels j'exprime tout ma gratitude.

A mon ami **Houari Emad Eddine**

Aux examinateurs, et membres du jury qui auront cette noble tâche  
d'appréciation de mon travail

Sans oublier mon binôme **Zakaria** auquel j'exprime toute ma gratitude.

A tous, mes vifs remerciements et mon profond respect.

**Youcef**

في هذه المذكرة، نستعرض نظام مدمج أوتوماتيكي لتصنيف أعطال المولدات الكهروضوئية يعتمد على صور حرارية و شبكة تعلم عميق (MobileNetV2). تم تطوير نظام مصنف لأعطال المولدات الكهروضوئية. الأعطال التي تم دراستها هي تراكم الغبار، الأوساخ، وتدهور وحدات التوليد الكهروضوئية. نظام التصنيف المطور تم استعماله (إدماجه) داخل متحكم دقيق (Nano 33 BLE Sense). يتم تخزين نتائج التصنيف في قاعدة بيانات، يمكن الإطلاع عليها عن بعد (IoT) للتحقق من حالة المولدات الكهروضوئية. النتائج العملية التي تم الحصول عليها تؤكد جدوى النظام المضمن لتصنيف أعطال المولدات الكهروضوئية ، بمتوسط دقة تصنيف تصل إلى % 93.4.

**الكلمات المفتاحية :** نظام مدمج، مولد كهروضوئي، متحكم دقيق، IOT، MobileNetV2، Nano 33 BLE.

---

## Résumé

Dans ce mémoire, un système automatique embarqué de classification des défauts dans un générateur photovoltaïque (PV) basé sur des images infrarouges et un réseau neuronal conventionnel profond, type (MobilNetV2), est présenté. Un classificateur a été développé pour la classification du type de défaut qui s'est survenu sur le générateur PV. Les défauts étudiés sont l'accumulation de poussière, la saleté, la dégradation des modules PVs. Le classificateur développé a été intégré dans un microcontrôleur (Nano 33 BLE Sense). Les résultats de classification sont sauvegardés dans une base de données, et accessible à distance (IoT) afin de vérifier l'état des générateur PVs. Les résultats pratique obtenus ont montré la faisabilité du système embarqué proposé pour le diagnostic des générateur PVs, avec une précision de classification moyenne de l'ordre de 93.4 %.

**Mots clés :** Photovoltaïque, Embarqué, MobilNetV2, Classification, IOT, microcontrôleur, Nano 33 BLE.

---

## Abstract

In this thesis, an automatic embedded system for photovoltaic generator faults classification based on infrared images and a deep conventional neutral network (MobileNetV2) is presented. A classifier is developed to classify the fault type present in the PV generator. The studied faults are the accumulation of dust, dirt, and degradation of PVs modules. The developed model was embedded in a microcontroller (Nano 33 BLE Sense). The classification results are saved in a database and can be accessed remotely (IoT) to verify the state of the PVs generators. The practical results obtained prove the feasibility of the proposed embedded system for PVs generators diagnosis, with an average classification accuracy of the order of 93.4%.

**Keywords :** Photovoltaic, Embedded, MobilNetV2, Classifier, IOT, microcontroller, Nano 33 BLE.

---

# Sommaire

Remerciements.....	i
Dédicaces.....	ii
Résumé.....	iv
Sommaire.....	v
Liste des figures.....	viii
Liste des tableaux.....	x
Liste des abréviations.....	xi
Introduction générale.....	1
<b>CHAPITRE 1</b> Diagnostic de défauts des générateurs photovoltaïques.....	3
1.1. Introduction.....	3
1.2. Généralités.....	3
1.2.1. Générateur photovoltaïque.....	3
1.2.2. Cellule photovoltaïque.....	3
1.2.3. Module photovoltaïque.....	4
1.3. Les défauts des modules photovoltaïques.....	5
1.3.1. Les bulles d'air.....	5
1.3.2. Fissure dans les cellules.....	5
1.3.3. Délaminage.....	6
1.3.4. Points chauds.....	7
1.3.5. Défaut de mismatch et d'ombrage partiel.....	7
1.3.6. Défauts de la diode by-pass et diode anti-retour.....	7
1.4. Diagnostic des modules photovoltaïques.....	8
1.4.1. Méthodes électriques.....	8
1.4.2. Méthodes non-électriques.....	9
1.5. Conclusion.....	10

---

<b>CHAPITRE 2</b>	<b>Réseaux neurones et apprentissage profond (CNN)</b>	11
2.1.	Introduction	11
2.2.	L'apprentissage automatique (Machine Learning)	12
2.3.	L'apprentissage profond (Deep Learning)	15
2.3.1.	Définition	15
2.3.2.	Pourquoi le l'apprentissage profond ?	15
2.4.	Les réseaux de neurones	16
2.4.1.	Définition	16
2.4.2.	Fonctionnement	17
2.4.3.	Les types de réseaux neurones	17
2.4.3.1.	Réseaux neurones retro-propagation (Feed-Forward)	18
2.4.3.2.	Réseau de neurones récurrents (RNN)	18
2.4.3.3.	Les Réseaux neurones convolutifs (CNN)	19
2.4.3.3.1.	Définition	19
2.4.3.3.2.	Les types de couche d'un réseau neurones convolutifs	19
2.4.3.3.3.	Fonctionnement du réseau de neurones convolutif	22
2.4.3.3.4.	Avantage des réseaux neurones convolutifs	23
2.4.3.3.5.	Les inconvénients des réseaux neuronaux convolutifs	23
2.5.	Conclusion	23
<b>CHAPITRE 3</b>	<b>Méthodologie et matériels</b>	24
3.1.	Introduction	24
3.2.	Ressources	24
3.2.1.	Ressources logicielles	24
3.2.2.	Ressources matérielles	25
3.3.	Méthodologie	27
3.3.1.	Préparation de la base de données	27
3.3.2.	Développement de modèle	28
3.3.2.1.	Le réseau de neurones utilisé (MobileNetV2)	29

---

3.3.2.2.	Procédure de développement d'un model sur Edge Impulse .....	29
3.3.3.	Développement et réalisation du circuit électronique .....	31
3.3.4.	Déploiement du modèle .....	33
3.3.5.	Test du modèle sur les deux cartes .....	34
3.3.6.	Base de données (Firebase) .....	35
3.3.7.	Développement d'une interface graphique.....	36
3.4.	Conclusion.....	37
<b>CHAPITRE 4</b>	<b>Résultats et interprétation .....</b>	<b>38</b>
4.1.	Introduction .....	40
4.2.	Résultats de la simulation.....	40
4.2.1.	Performances du modèle durant l'entraînement.....	41
4.2.2.	Performance du modèle durant la phase de validation.....	41
4.2.3.	Optimisation du modèle .....	43
4.3.	Résultats pratique .....	44
4.3.1.	Système réalisé .....	44
4.3.2.	Etude comparative.....	46
4.4.	Conclusion.....	47
	Conclusion générale .....	48
	Références .....	49

## Liste des figures

<b>Figure 1.1.</b> Evolution de la répartition des technologies de cellule industrialisées.....	4
<b>Figure 1.2.</b> Module photovoltaïque [14] .....	5
<b>Figure 1.3.</b> Défaut de bulles d'air.....	5
<b>Figure 1.4.</b> Fissures dans les cellules PV [15].....	6
<b>Figure 1.5.</b> Défaut de délaminage .....	6
<b>Figure 1.6.</b> Défaut de point chaud [15] .....	7
<b>Figure 1.7.</b> Défaut de la diode de by-pass [15] .....	8
<b>Figure 1.8.</b> Exemples de la localisation de défauts par une caméra thermique [12].....	10
<b>Figure 2.1.</b> Le lien entre l'intelligence artificielle et l'apprentissage automatique et l'apprentissage profond .....	11
<b>Figure 2.2.</b> Les différents types d'apprentissage automatique .....	12
<b>Figure 2.3.</b> Apprentissage supervisé.....	13
<b>Figure 2.4.</b> Apprentissage non supervisé.....	14
<b>Figure 2.5.</b> Apprentissage par renforcement .....	14
<b>Figure 2.6.</b> Schéma illustratif de l'apprentissage profond.....	15
<b>Figure 2.7.</b> La différence entre ML et DL .....	16
<b>Figure 2.8.</b> Structure d'un réseau neuronal (multicouches) .....	17
<b>Figure 2.9.</b> Réseau neurones (Feed-Forward) .....	18
<b>Figure 2.10.</b> Réseau de neurones récurrents.....	19
<b>Figure 2.11.</b> Architecture de réseau de neurones convolutif.....	20
<b>Figure 2.12.</b> Couche de convolution .....	20
<b>Figure 2.13.</b> Exemple d'opération Pooling-Max.....	21
<b>Figure 2.14.</b> Allure de la fonction ReLU.....	21
<b>Figure 2.15.</b> Les couches entièrement connectées (Fully Connected Layers) .....	22
<b>Figure 3.1.</b> La plateforme Edge Impulse .....	24
<b>Figure 3.2.</b> L'interface de l'IDE Arduino .....	25
<b>Figure 3.3.</b> La carte a microcontrôleur : a) Arduino Nano 33 BLE Sense, et b) Raspberry PI pico .....	26
<b>Figure 3.4.</b> La caméra thermique.....	26
<b>Figure 3.5.</b> La carte Arduino Nano 33 IoT.....	27
<b>Figure 3.6.</b> IR images de thermographie : a) état bonne (N), b) sale (D2), c) dégradé (D3) et d) défaut de poussière sur le module PV (D1) .....	28

---

<b>Figure 3.7.</b> La procédure d'intégration [25].....	28
<b>Figure 3.8.</b> La structure de base de MobileNetV2 [26].....	29
<b>Figure 3.9.</b> Schéma de fonctionnement de système embarqué développé .....	31
<b>Figure 3.10.</b> Le schéma électrique du système (à base de l'Arduino Nano 33 BLE Sense) ...	32
<b>Figure 3.11.</b> Le circuit électrique réalisé sur une Breadboard (à base de l'Arduino Nano 33 BLE Sense).....	32
<b>Figure 3.12.</b> Le schéma électrique du système (à base de la Raspberry PI pico).....	33
<b>Figure 3.13.</b> Le circuit électrique réaliser sur une Breadboard (à base de la Raspberry PI pico) .....	33
<b>Figure 3.14.</b> La page de déploiement de la plate-forme Edge Impulse.....	34
<b>Figure 3.15.</b> Résultats de classification pour l'image IR choisie.....	34
<b>Figure 3.16.</b> Résultats de la classification sur les deux cartes.....	35
<b>Figure 3.17.</b> La base de données (Realtime Database).....	35
<b>Figure 3.18.</b> L'interface graphique développée.....	36
<b>Figure 4.1.</b> Explorateur des caractéristiques des 4 classes durant la phase de la validation ...	42
<b>Figure 4.2.</b> L'image IR sélectionner pour tester le modèle .....	43
<b>Figure 4.3.</b> Circuit imprimée (PCB).....	45
<b>Figure 4.4.</b> Photo de circuit réalisé.....	45
<b>Figure 4.5.</b> Test direct des résultats de classification .....	46

## Liste des tableaux

<b>Tableau 4.1.</b> Paramètres du modèle .....	40
<b>Tableau 4.2.</b> Matrice de confusion (les performances du modèle durant l'entraînement).....	41
<b>Tableau 4.3.</b> Matrice de confusion (performance du modèle durant la validation) .....	42
<b>Tableau 4.4.</b> Validation du modèle avec une image inconnue.....	43
<b>Tableau 4.5.</b> Comparaison entre le modèle optimiser et non optimiser.....	44
<b>Tableau 4.6.</b> Caractéristiques principales de chaque code .....	47

## Liste des abréviations

PV : photovoltaïque.

AC : Alternative Current.

DC : Direct Current.

CNN : Convolutional Neural Network.

ML : Machine Learning.

DL : Deep Learning.

IDE : Integrated Development Environment.

IR : Infrarouges.

I2C : Inter-Integrated Circuit.

UART : Universal Asynchronous Receiver-Transmitter.

PCB : Printed Circuit Board.

# **Introduction générale**

Le photovoltaïque (PV) est certainement l'une des technologies les plus prometteuses de production d'énergie renouvelables, en raison de sa capacité à transformer directement la lumière du soleil en électricité [1]. Ces dernières années, l'industrie photovoltaïque a fait des progrès significatifs dans le monde entier. Leur fiabilité s'est également améliorée en raison de l'accent accru mis sur la qualité et la production des matériaux, et de la capacité à résister à des conditions météorologiques extrêmes.

Cependant, plusieurs autres facteurs pourraient affecter la production PV tels que la dégradation thermique, l'encrassement et l'ombrage qui pourraient conduire à des effets de points chauds. Pour ces raisons, plusieurs études se sont concentrées sur la surveillance du système PV et la détection des pannes pour assurer une grande fiabilité et réduire les coûts et les délais de maintenance. La plupart des défauts sont survenues sur le générateur PV, comme l'accumulation de poussière sur la surface PV, le vieillissement de cellules, l'effet d'ombrage partiel et les défauts internes. Plusieurs études récentes de la littérature ont été confrontées à ce type de problèmes basés sur le traitement des images infrarouge (IR) dans le diagnostic des installations PV [2-7].

L'objectif principale de ce mémoire est de développer un système embarqué, à base de l'apprentissage automatique (EML), de diagnostic des générateurs/modules PV en utilisant un système à base de microcontrôleur (Nano 33 BLE Sense [8], et Raspberry Pi pico [9]) et des images thermographiques. Le système proposé peut détecter, classer et visualiser les résultats en ligne, sur une page web. Ce travail a été réalisé au niveau du laboratoire des énergies renouvelables, Université de Jijel.

Il est à noter que ce travail c'est la continuité du mémoire Master réalisé par C. Moussaoui [10,11] en 2022, dont la thématique est de réaliser un système semi-automatique à base d'une carte Raspberry Pi 4. Dans ce mémoire nous allons essayer d'améliorer et rendre ce système automatique en utilisant une caméra thermique et des cartes a microcontrôleurs à faible consommation.

Le mémoire est organisé comme suit :

Le chapitre 1 est consacré au diagnostic des défauts des générateurs photovoltaïques. En introduisant des généralités sur ces générateurs PV, et présentant les défauts et quelques méthodes de diagnostic.

Le chapitre 2 est consacré aux méthodes de classification, en abordant les types d'apprentissage et une brève introduction aux réseaux de neurone convolutifs.

Dans le chapitre 3 nous allons présenter la réalisation pratique de notre système embarqué, et le développement de l'interface graphique avec un vu d'ensemble du matériel et logiciels utilisés.

Dans le chapitre 4 nous allons présenter les résultats obtenus, à savoir ceux de la phase d'apprentissage, de la validation et de déploiement et le tester sur un microcontrôleur (Arduino Nano 33 BLE Sense et Raspberry PI pico).

# **CHAPITRE 1**

Diagnostic de défauts des générateurs  
photovoltaïques

## **1.1. Introduction**

Dans ce chapitre, nous présentons des généralités sur les générateurs PVs, et nous allons présenter certains défauts, puis nous verrons les méthodes utilisées pour le diagnostic des générateurs PVs et la procédure de détection de défauts.

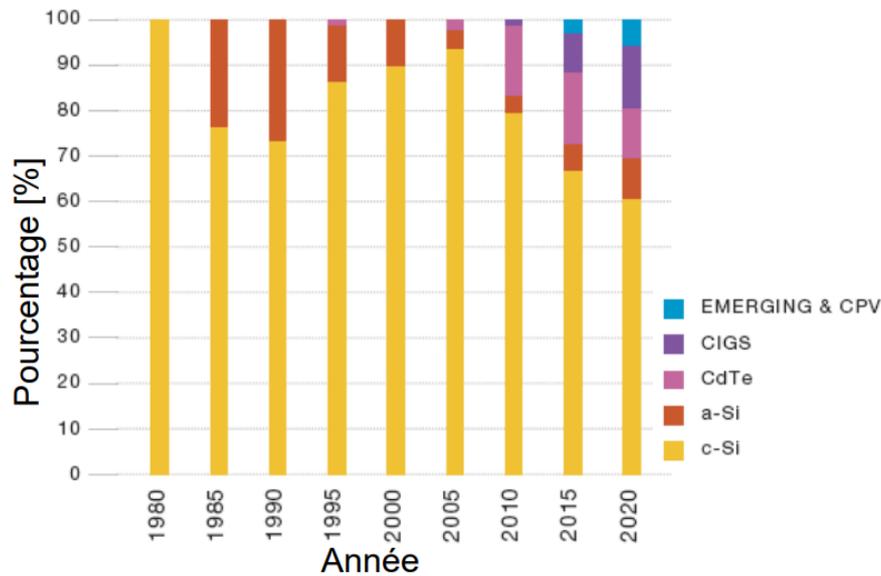
## **1.2. Généralités**

### **1.2.1. Générateur photovoltaïque**

Le générateur PV est l'unité de production d'énergie électrique sous forme de courant continu. Le composant élémentaire de cette unité qui convertit l'énergie solaire en l'énergie électrique est la cellule photovoltaïque [12].

### **1.2.2. Cellule photovoltaïque**

Il existe un grand nombre de technologies permettant la fabrication de cellules photovoltaïques mais beaucoup sont encore en phase de recherche [12]. La figure 1.1 donne l'évolution historique de la répartition des technologies industrialisées. Les cellules actuellement industrialisées en grande quantité sont produites à partir de silicium (environ 80 % en 2010 pour le silicium cristallin "c-Si"). La technologie amorphe (a-Si) qui a été une des technologies préférées a connu une baisse dans le marché actuel par rapport à la technologie couche mince. Par exemple, le marché de la technologie CdTe (Cadmium Telluride) a augmenté de 2 % en 2005 à 13 % en 2010. Une autre technologie de type couche mince comme CIGS (Copper Indium Gallium Selenide) et les autres technologies émergentes sont en voie de se développer.

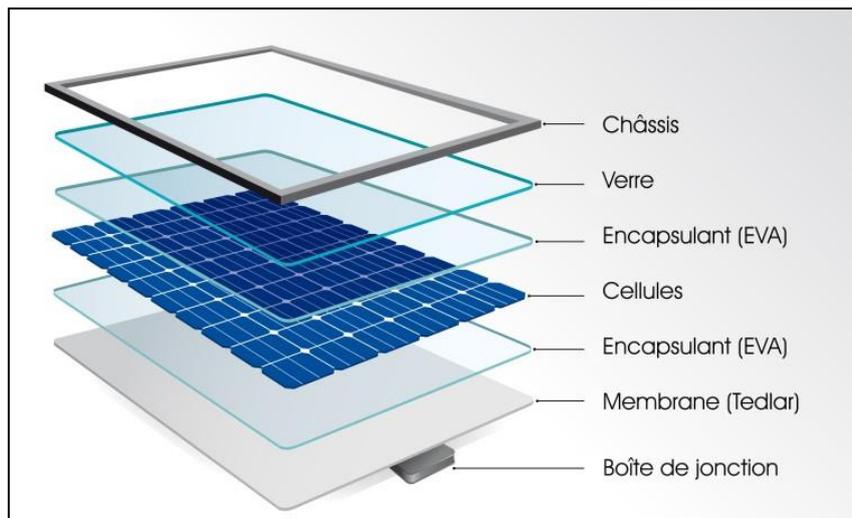


**Figure 1.1.** Evolution de la répartition des technologies de cellule industrialisées

La tension générée par une cellule photovoltaïque est limitée à la valeur du gap du matériau dont elle est issue. Elle est de l'ordre de 0.6 V pour les cellules de type cristallin et amorphe. Pour le courant, il est en fonction de la surface de la cellule et pour une même surface, il dépend du rendement de la cellule. Le rendement des cellules monocristallines est de l'ordre de 15 à 22 %, tandis celui des cellules multi cristallines est de l'ordre de 10 à 15 % [12].

### 1.2.3. Module photovoltaïque

Un module photovoltaïque est un générateur électrique qui transforme l'énergie issue des photons de la lumière en électricité. Un module est composé de plusieurs cellules photovoltaïques, généralement fabriquées à base de silicium, mais plusieurs autres matériaux sont aujourd'hui à l'étude. La puissance électrique continue créée dans une cellule s'additionne à la puissance générée par les autres cellules du module. Les panneaux photovoltaïques sont ensuite reliés électriquement entre eux, pour former des strings, que l'on raccorde sur un onduleur [13] (Voir Fig.1.2).



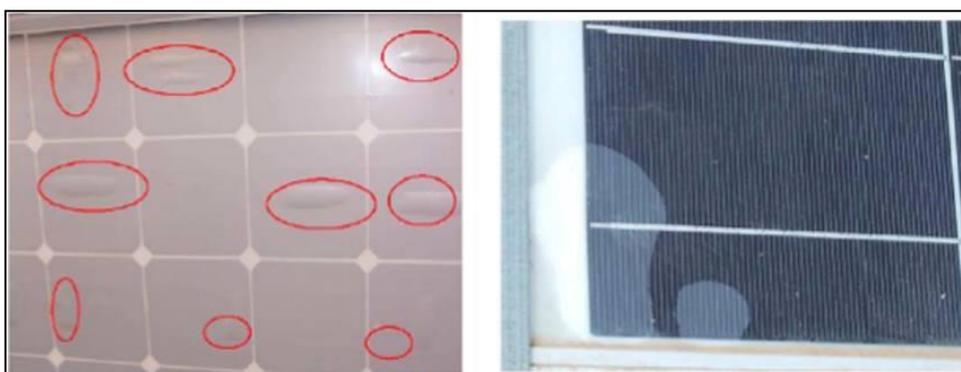
**Figure 1.2.** Module photovoltaïque [14]

### 1.3. Les défauts des modules photovoltaïques

Les performances des modules PVs sont affectées par plusieurs variables environnementales, provoquant différents défauts qui réduisent la production d'énergie des panneaux PV. Dans cette partie on va parler des défauts des MPV les plus fréquents [15,16].

#### 1.3.1. Les bulles d'air

Une bulle est généralement due à une réaction chimique où des gaz sont libérés. Quand une bulle se produit, un gonflement apparaît encapsulant le polymère ou le couvercle arrière (Voir Fig.1.3). La formation d'une bulle rend la dissipation de la chaleur des cellules plus difficile ce qui minimise la durée de vie des PVs.

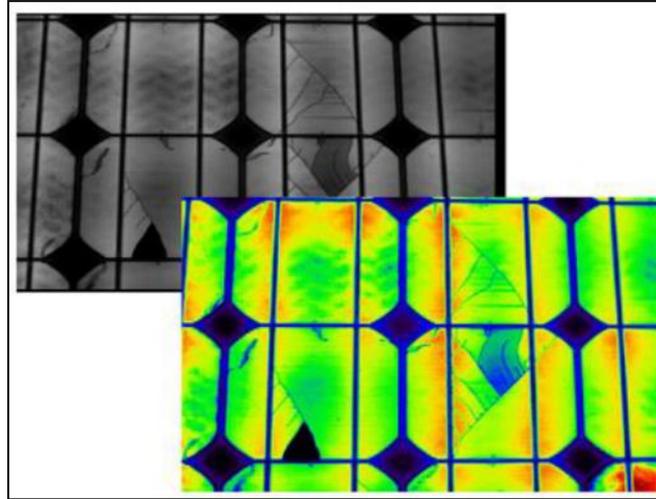


**Figure 1.3.** Défaut de bulles d'air

#### 1.3.2. Fissure dans les cellules

La réduction d'épaisseur, de 300  $\mu\text{m}$  à moins de 200  $\mu\text{m}$ , et l'augmentation de la superficie, à 210 mm x 210 mm, rendait les cellules plus fragiles et sensibles

aux fractures au cours de leur manipulation et au cours du stockage. Les microfissures se composent de petites fissures dans les cellules. Habituellement, elles ne sont pas visibles par la vue, mais peuvent affecter les deux côtés de la cellule. Elles produisent une perte de consistance de la cellule [15] (Voir Fig.1.4).



**Figure 1.4.** Fissures dans les cellules PV [15]

### 1.3.3. Délaminage

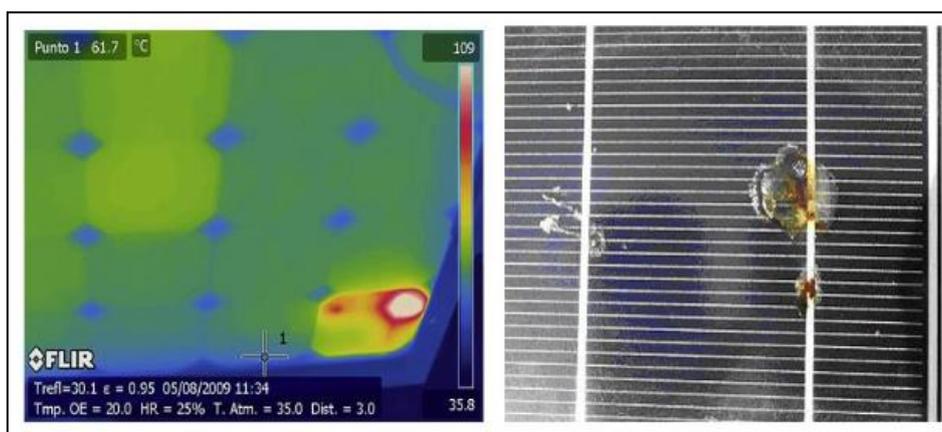
Le délaminage est la conséquence de la perte d'adhérence entre les différentes couches du module PV et le détachement subséquent de ces couches (Voir Fig.1.5). Il peut apparaître entre encapsulant et cellules polymères ou entre les cellules et la vitre avant. Le délaminage est un problème majeur, car il peut conduire à deux effets : une diminution de la lumière, où la réflexion augmente, ainsi que la pénétration de l'eau à l'intérieur de la structure de module [15].



**Figure 1.5.** Défaut de délaminage

### 1.3.4. Points chauds

C'est une zone d'un module PV à très haute température, et qui pourrait endommager les cellules (l'ombrage partiel, la différence entre cellules...) ou n'importe quel autre élément du module dans un court laps de temps de fonctionnement. Si le point chaud est causé par un court-circuit d'une cellule, ou une erreur dans l'interconnexion entre les cellules générant également un court-circuit d'une cellule. En outre, si une cellule génère un courant plus faible que le reste des cellules (appelée mésappariement) Un point chaud apparaît sous la forme d'une cellule entière fonctionnant à une température plus élevée que le reste des cellules (Voir Fig.1.6).



**Figure 1.6.** Défaut de point chaud [15]

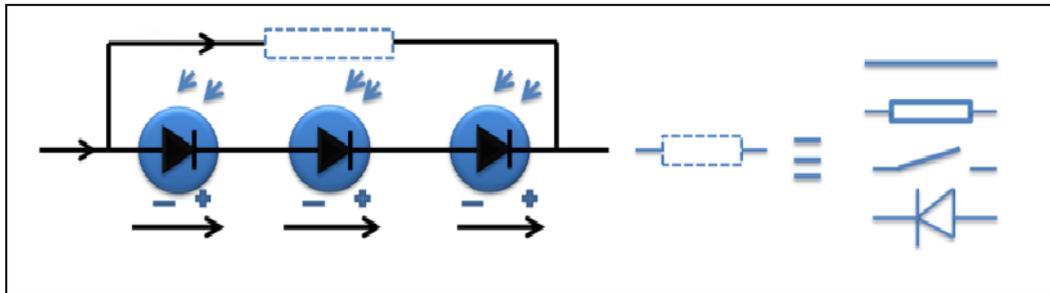
### 1.3.5. Défaut de mismatch et d'ombrage partiel

Le défaut de mismatch est causé par le groupement de cellules possédant une caractéristique I-V non identique, cette différence est à cause des propriétés physiques différentes suite à une tolérance dans la fabrication, ou à cause des conditions de fonctionnement différents causés par des défauts (comme l'ombrage partiel). Le défaut d'ombrage partiel est un cas particulier de défaut de mismatch, car sa présence réduit l'ensoleillement reçu par des cellules [12].

### 1.3.6. Défauts de la diode by-pass et diode anti-retour

La diode de by-pass se place, en général, en parallèle avec une branche de dix-huit cellules (Voir Fig.1.7) [16]. On peut constater plusieurs défauts de cette dernière, ces défauts sont en général des défauts de fabrication. Donc dans un panneau photovoltaïque on peut trouver une diode déconnectée, une diode équivalente à une impédance quelconque, une diode court-circuitée et une diode inversée. De la même manière que pour la diode de by-pass, la

diode anti retour est exposée aux mêmes types de défauts envisagés pour la diode anti-retour : court-circuit, impédance quelconque, circuit ouvert et diode inversée [15].



**Figure 1.7.** Défaut de la diode de by-pass [15]

## 1.4. Diagnostic des modules photovoltaïques

Dans la partie précédente nous avons parlé des défauts les plus fréquents, et dans cette partie nous parlerons du diagnostic des modules PV. Généralement on distingue deux grandes catégories de méthodes de diagnostic, méthodes visuelles et thermiques (non-électriques) et méthodes électriques [16].

### 1.4.1. Méthodes électriques

Les systèmes de surveillance actuels d'une installation PV sont essentiellement intégrés à des onduleurs, les données mesurées sont les mêmes d'un système à l'autre [16, 12]. Parmi Les grandeurs mesurées les plus courantes :

- Le courant débité par le champ PV
- La tension aux bornes du champ PV
- La résistance d'isolement entre les bornes positive et négative du champ PV

Il est aussi possible d'ajouter les grandeurs complémentaires tell que la température ambiante du site et l'ensoleillement aux mesures électriques. Ces données nécessitent un capteur spécifique.

Les mesures du côté AC sont directement liées à l'énergie qui sera vendue donc elles sont plus importantes en nombre. Il est courant de relever :

- Le courant AC
- La tension AC
- La fréquence
- L'impédance du réseau vue par l'onduleur

On peut déduire d'autres grandeurs :

- La puissance instantanée DC
- La puissance instantanée AC
- L'énergie produite sur différentes périodes (suivant la capacité de stockage des données) côtés DC et AC

Ces données sont calculées directement par le microcontrôleur embarqué dans l'onduleur ou par l'automate associé.

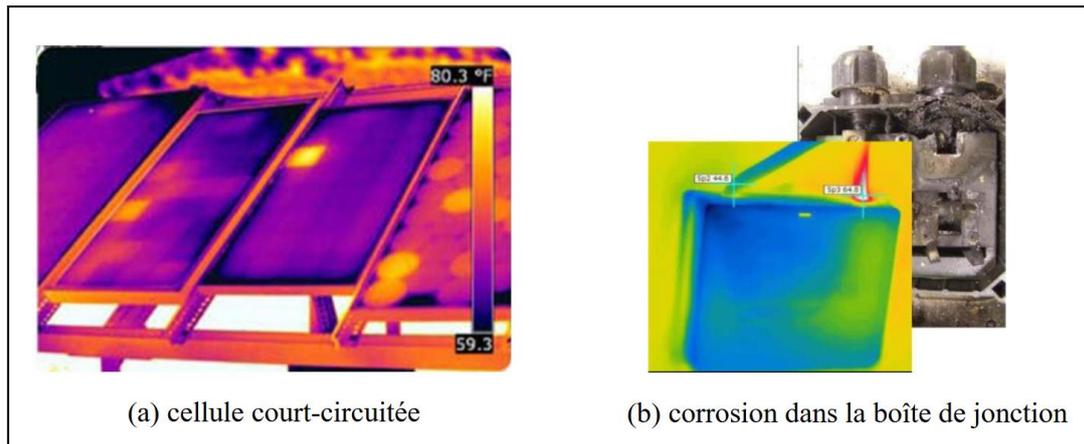
En termes de diagnostic pour de la maintenance préventive ou curative, il existe très peu d'informations et de fonctionnalités. Au mieux, seule une mesure tension/courant string par string permet d'identifier un éventuel déséquilibre de production et/ou une défaillance d'un onduleur.

#### **1.4.2. Méthodes non-électriques**

Il existe plusieurs méthodes non-électriques du diagnostic des défauts au niveau de cellule PV. On peut citer comme méthodes :

- Les essais mécaniques de flexion.
- L'imagerie par photoluminescence et électroluminescence.
- Tests de thermographie.

Au niveau du module PV, la méthode de l'imagerie d'infrarouge est la plus utilisée. Cette méthode baser sur le fait que tous les matériels émettent un rayonnement infrarouge sur une plage de longueur d'onde qui dépend de la température du matériau. En examinant la distribution de la température au niveau du module, des anomalies peuvent être localisées [12,16].



**Figure 1.8.** Exemples de la localisation de défauts par une caméra thermique [12]

## 1.5. Conclusion

Dans ce chapitre nous avons commencé par une brève introduction au générateurs PVs, ensuite nous avons présenté les différents défauts des modules PVs, ainsi que les méthodes de diagnostics existants. Dans le chapitre suivant nous allons présenter une introduction à l'apprentissage automatique et profond.

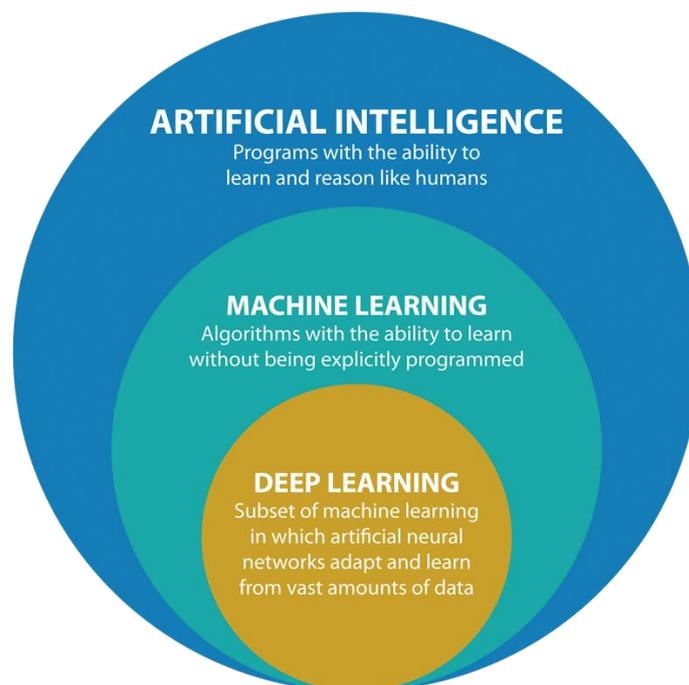
# **CHAPITRE 2**

Réseaux neurones et apprentissage  
profond (CNN)

## 2.1. Introduction

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage automatique (Machine Learning) champ d'étude de l'intelligence artificielle. Par conséquent, L'apprentissage profond (en anglais Deep Learning, Deep Structured Learning, Hierarchical Learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires [17].

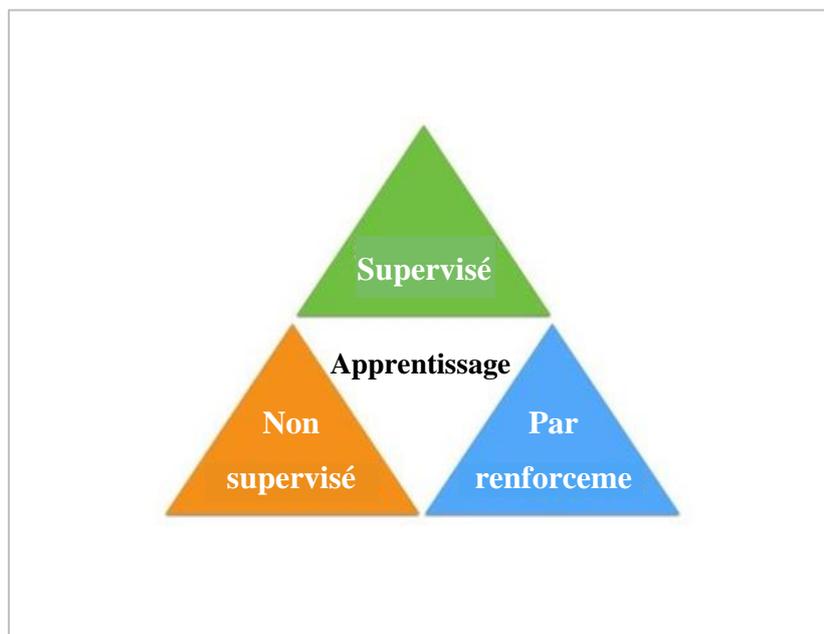
Après avoir défini très brièvement la notion de l'intelligence artificielle et de l'apprentissage automatique, dans ce chapitre nous développerons le rôle de l'apprentissage profond et son champ d'application dans l'architecture d'étude de l'intelligence artificielle et une vision générale sur les réseaux neurones convolutifs [18].



**Figure 2.1.** Le lien entre l'intelligence artificielle et l'apprentissage automatique et l'apprentissage profond

## 2.2. L'apprentissage automatique (Machine Learning)

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle qui implique la formation d'algorithmes pour faire des prédictions ou des décisions basées sur des données. L'objectif de l'apprentissage automatique est de permettre aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sur une tâche spécifique au fil du temps, sans être explicitement programmés. Il existe trois principaux types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

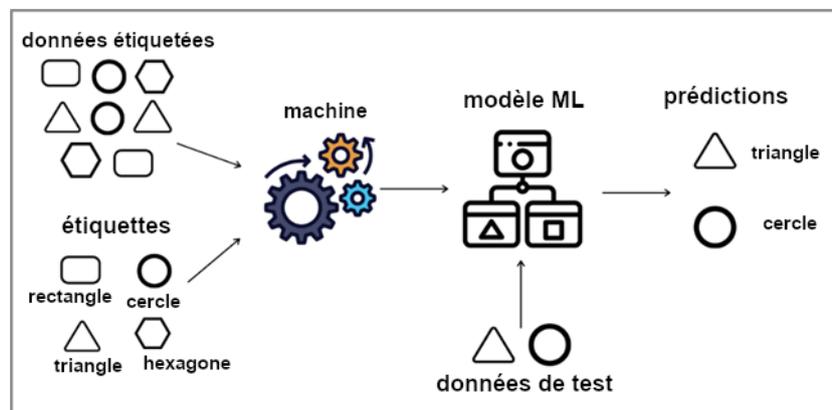


**Figure 2.2.** Les différents types d'apprentissage automatique

- **L'apprentissage supervisé** implique la formation d'un modèle sur des données étiquetées, où la sortie correcte est connue. Le modèle apprend à faire des prédictions basées sur les données d'entrée et les étiquettes de sortie correspondantes.

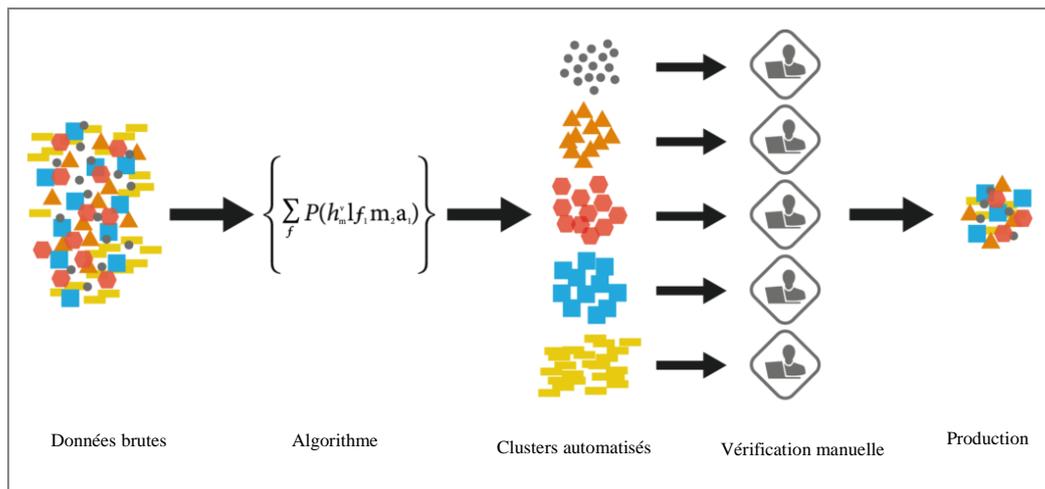
Les algorithmes d'apprentissage supervisé conviennent aux tâches suivantes :

- **Classification binaire** : Diviser les données en deux catégories.
- **Classification multi-classes** : Choix entre plus de deux types de réponses.
- **Modélisation de régression** : Prédire des valeurs continues.
- **Assemblage** : combinaison des prédictions de plusieurs modèles d'apprentissage automatique pour produire une prédiction précise [18]



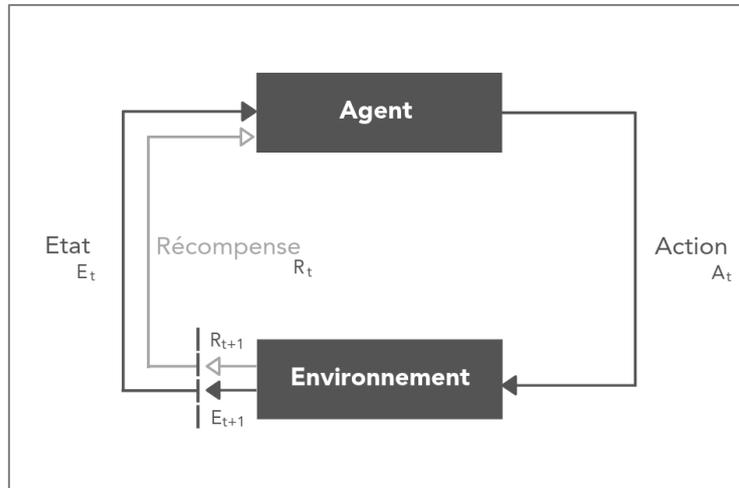
**Figure 2.3.** Apprentissage supervisé

- **L'apprentissage non supervisé** implique la formation d'un modèle sur des données non étiquetées, où la sortie correcte n'est pas connue. Le modèle apprend à identifier des modèles et des relations dans les données sans aucune aide.



**Figure 2.4.** Apprentissage non supervisé

-L'**apprentissage par renforcement** consiste à entraîner un modèle à prendre des décisions en fonction des commentaires de l'environnement. Le modèle apprend à prendre des mesures qui maximisent un signal de récompense, comme gagner un jeu ou accomplir une tâche.



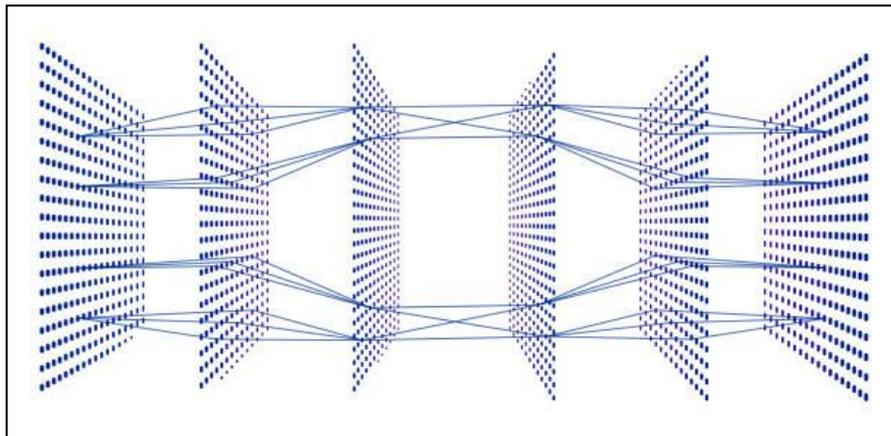
**Figure 2.5.** Apprentissage par renforcement

L'apprentissage automatique a de nombreuses applications pratiques, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel, la détection des fraudes et les systèmes de recommandation.

## 2.3. L'apprentissage profond (Deep Learning)

### 2.3.1. Définition

L'apprentissage en profondeur est un sous-ensemble de l'apprentissage automatique qui implique la formation de réseaux de neurones artificiels à plusieurs couches pour apprendre et faire des prédictions ou des décisions basées sur des données complexes. Elle est dite « profonde » car les réseaux de neurones comportent de nombreuses couches, leur permettant d'apprendre des représentations hiérarchiques des données. L'apprentissage en profondeur a été couronné de succès dans un large éventail d'applications, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel et la conduite autonome.

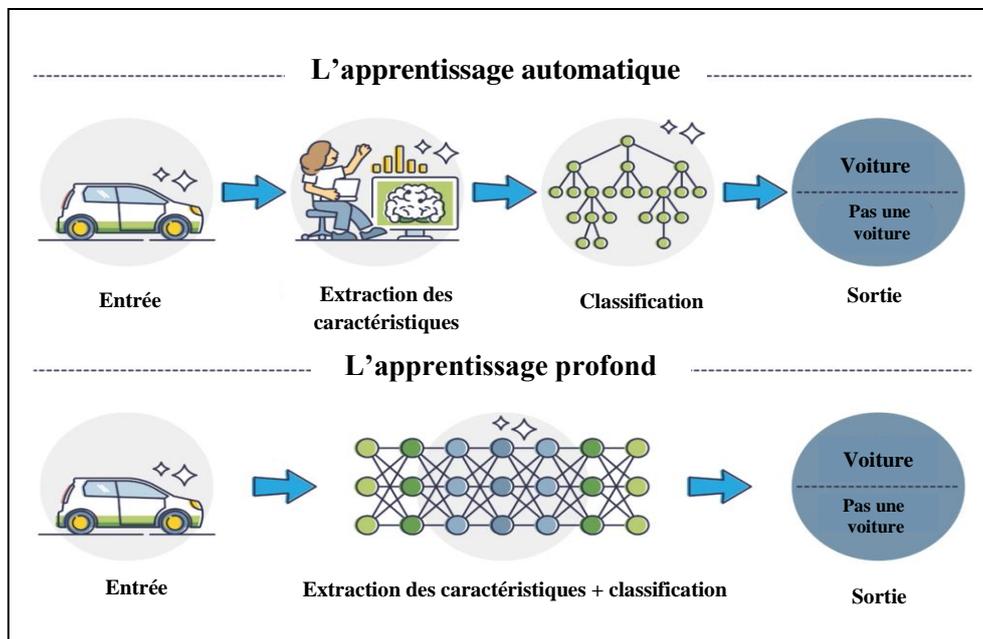


**Figure 2.6.** Schéma illustratif de l'apprentissage profond

### 2.3.2. Pourquoi le l'apprentissage profond ?

Dans les dernière année l'apprentissage en profondeur est devenu de plus en plus populaire en raison de sa capacité à apprendre et à s'améliorer à partir de grandes quantités de données, et de son succès dans la résolution d'un large éventail de problèmes tels que la reconnaissance d'images, le traitement du langage naturel et la reconnaissance vocale. L'un des principaux avantages de l'apprentissage en profondeur est sa capacité à apprendre et à extraire automatiquement des fonctionnalités à partir de données brutes, sans avoir besoin d'une ingénierie manuelle des fonctionnalités. Cela le rend particulièrement utile dans les applications où les données sont complexes et de grande dimension, telles que la reconnaissance d'images et de la parole. L'apprentissage en profondeur a également réussi à résoudre des problèmes qui étaient auparavant considérés comme difficiles ou impossibles à résoudre avec les techniques traditionnelles d'apprentissage automatique aussi pour sa capacité à gérer de grandes quantités

de données. Les modèles d'apprentissage en profondeur peuvent être formés sur des ensembles de données massifs, ce qui leur permet d'apprendre des modèles et des relations complexes qui peuvent ne pas être apparents pour les humains

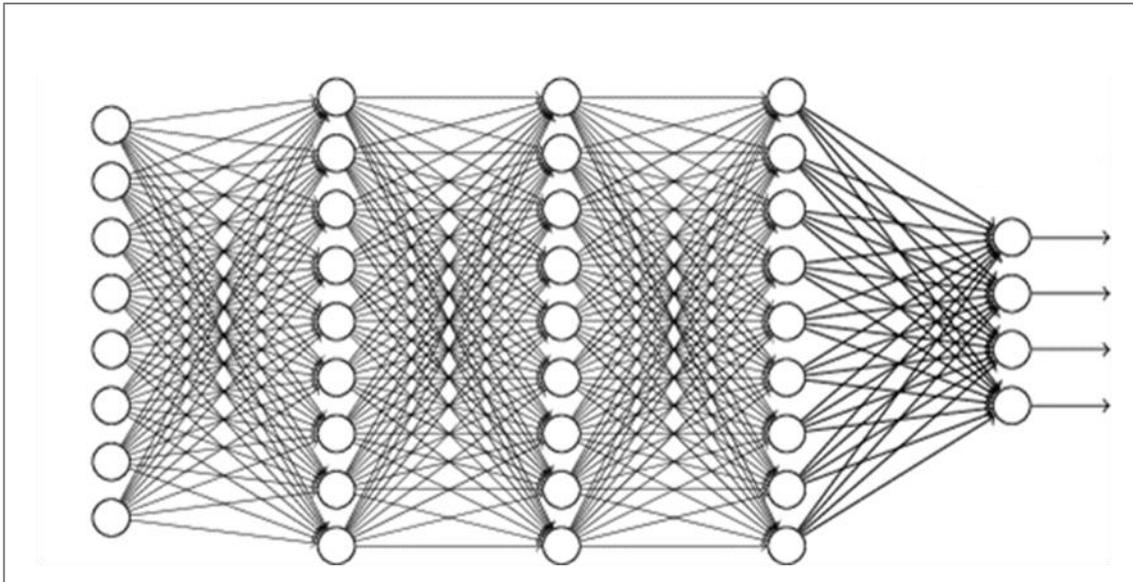


**Figure 2.7.** La différence entre ML et DL

## 2.4. Les réseaux de neurones

### 2.4.1. Définition

Un réseau de neurones, également sous le nom réseau neurones artificiel (ANN : Artificial Neural Network) ou réseau de neurones à impulsion (SNN : Spike Neural Network) est un type d'algorithme d'apprentissage automatique qui s'inspire de la structure et de la fonction du cerveau humain. Il se compose de couches de nœuds interconnectés, ou neurones, qui traitent et transmettent les informations. Les réseaux de neurones sont utilisés dans un large éventail d'applications, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel et l'analyse prédictive. Ils sont particulièrement utiles dans les applications où les données sont complexes et de grande dimension, et où les techniques d'apprentissage automatique traditionnelles peuvent ne pas être efficaces. Les réseaux de neurones peuvent automatiquement apprendre et extraire des fonctionnalités à partir de données brutes, et peuvent être entraînés sur de grands ensembles de données pour améliorer la précision et les performances.



**Figure 2.8.** Structure d'un réseau neuronal (multicouches)

### 2.4.2. Fonctionnement

Le réseau neuronal reçoit des données d'entrée qui sont transmises à la couche d'entrée, chaque entrée est représentée par un neurone. Ensuite les données d'entrée seront traitées à travers une ou plusieurs couches cachées (Hidden layers) constituées de neurones interconnectés. Chaque neurone de la couche cachée reçoit une entrée des nœuds de la couche précédente et applique une fonction mathématique à l'entrée pour produire une sortie.

La couche finale du réseau neuronal est la couche de sortie qui produit la sortie en fonction d'entrée et du traitement qui s'est produit dans la couche cachée, la couche de sortie peut avoir un ou plusieurs neurones selon le type de problème à résoudre.

Avant qu'un réseau de neurones puisse être utilisé pour faire des prédictions ou des classifications, il doit être formé sur un ensemble de données. Pendant la formation, le réseau de neurones ajuste les poids et les biais des neurones dans les couches cachées pour minimiser la différence entre la sortie prédite et la sortie réelle. Une fois le réseau de neurones est formé, il peut être utilisé pour faire des prédictions ou des classifications sur de nouvelles données.

### 2.4.3. Les types de réseaux neurones

- Il existe différents types de réseaux neuronaux nous mentionnons :
  - Réseaux neurones retro-propagation (Feed-forward)
  - Réseaux neurones récurrents (RNN)

- Réseaux neurones convolutifs (CNN)

### 2.4.3.1. Réseaux neurones retro-propagation (Feed-Forward)

Est un type de réseau de neurones où les informations circulent dans une direction directe c'est à dire de l'entrée aux nœuds de traitement (couche cachée) puis aux sorties.

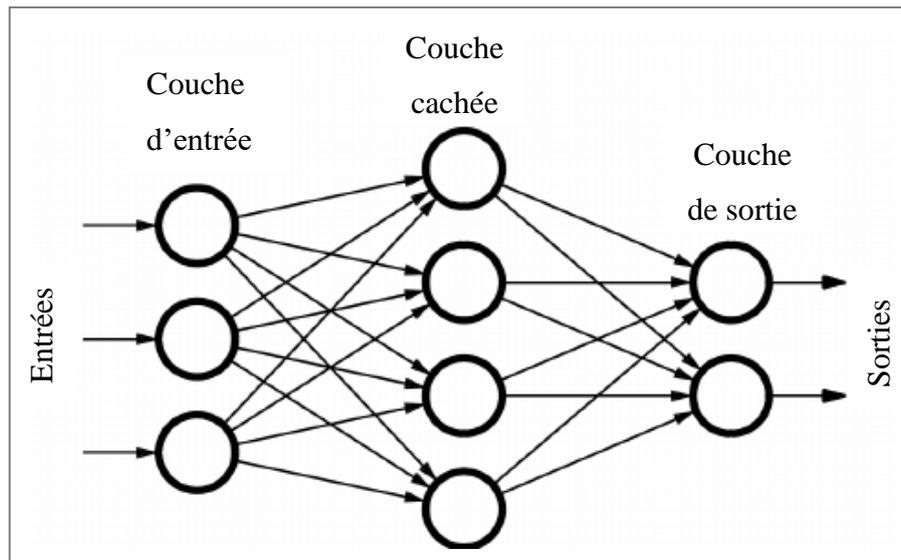
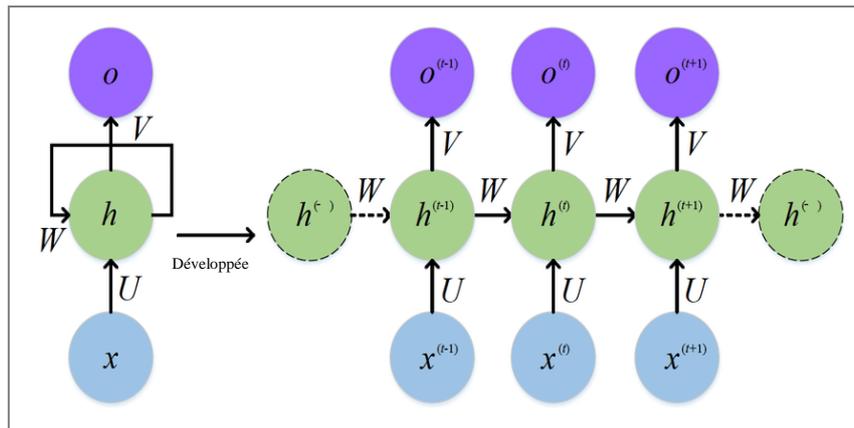


Figure 2.9. Réseau neurones (Feed-Forward)

### 2.4.3.2. Réseau de neurones récurrents (RNN)

Un RNN a une architecture qui admet des boucles. Ainsi, contrairement aux réseaux de neurones à propagation avant, où les entrées des neurones d'une couche ne proviennent que des couches précédentes, la sortie de certains neurones d'un réseau récurrent sera donnée en entrée à des neurones des couches précédentes. Les réseaux récurrents conviennent en particulier pour l'analyse de données séquentielles ou temporelles [19].



**Figure 2.10.** Réseau de neurones récurrents

### 2.4.3.3. Les réseaux de neurones convolutifs (CNN)

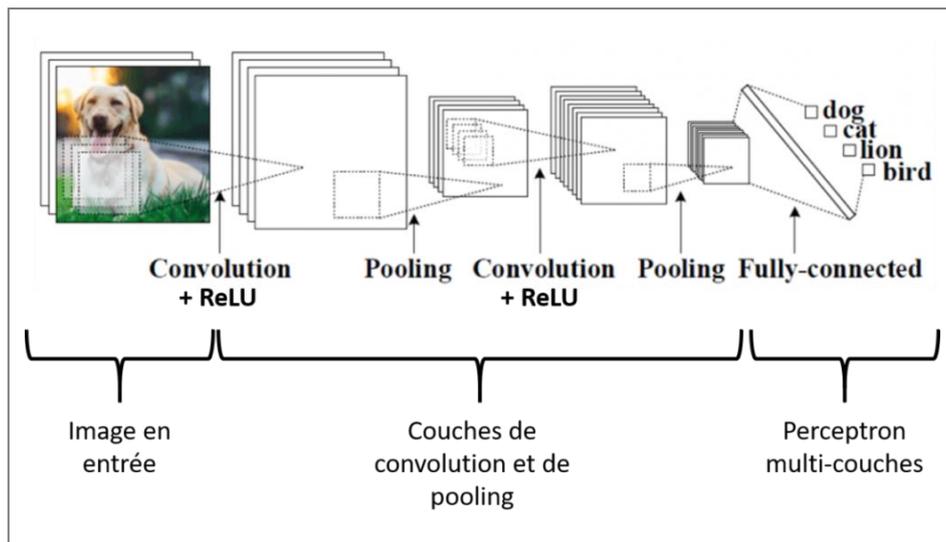
#### 2.4.3.3.1. Définition

Un réseau de neurones convolutifs ou réseau de neurones à convolution (en anglais ConvNet pour Convolutional Neural Networks) est un type de réseau de neurones artificiels à propagation directe (feed-forward) dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités d'informations [20].

#### 2.4.3.3.2. Les types de couche d'un réseau neurones convolutifs

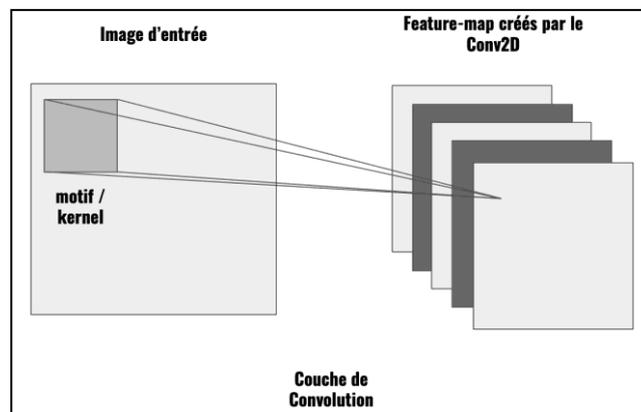
- Dans un réseau de neurones convolutif nous trouvons quatre types de couches :

- La couche de **convolution**
- La couche de **pooling**
- La couche de **correction ReLU**
- La couche **fully-connected**.



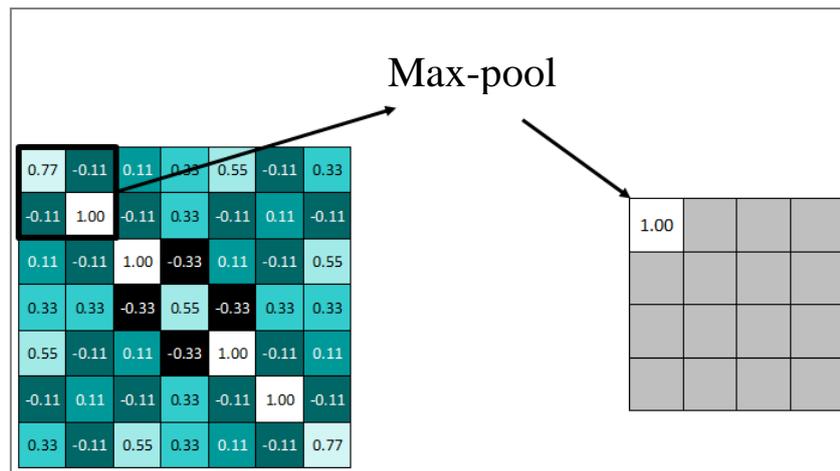
**Figure 2.11.** Architecture de réseau de neurones convolutif

**Les couches de convolution (Convolutional Layers) :** Les couches de convolution sont un ensemble de filtres qui sont appris au cours de l'entraînement. La taille et le nombre de ces filtres sont définis à priori [21].



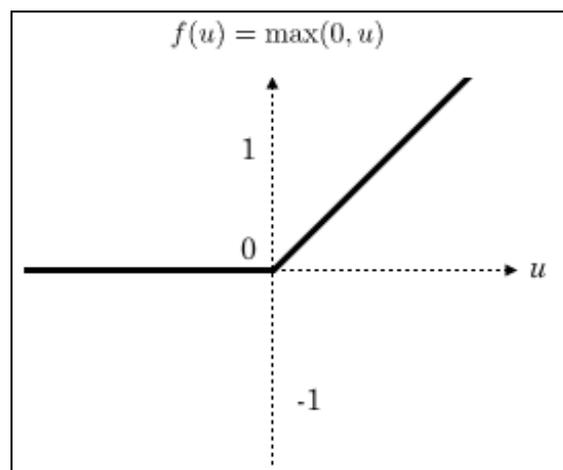
**Figure 2.12.** Couche de convolution

**Les couches de regroupements (Pooling Layers) :** Il est commun d'insérer périodiquement après les couches de convolutions des couches de Pooling. Ces fonctions sont prédéfinies et réduisent pour les couches ultérieures le nombre de paramètres à apprendre tout en élargissant le champ de réception (receptive field). Elles opèrent indépendamment aux différentes profondeurs du réseau et ne nécessitent pas de poids à entraîner [21].



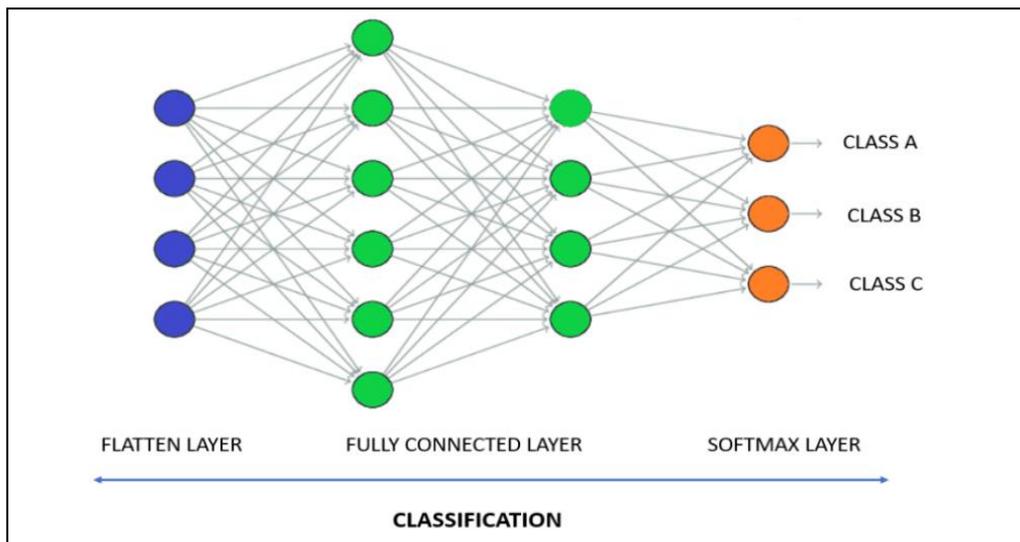
**Figure 2.13.** Exemple d'opération Pooling-Max

**La couche d'activation ReLU :** C'est une fonction simple qui renvoie l'entrée si elle est positive et zéro si non, elle peut être mathématiquement définie comme :  $f(x) = \max(0, x)$ .



**Figure 2.14.** Allure de la fonction ReLU

**Les couches entièrement connectées (Fully Connected Layers) :** Les neurones dans ces couches sont tous connectés à l'ensemble des neurones des cartes d'activations précédentes [21].



**Figure 2.15.** Les couches entièrement connectées (Fully Connected Layers)

#### 2.4.3.3.3. Fonctionnement du réseau de neurones convolutif

Le premier bloc fait la particularité de ce type de réseaux de neurones, puisqu'il fonctionne comme un extracteur des caractéristiques (features). Pour cela, il effectue du modèle correspondant en appliquant des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de convolution, et renvoie des features-maps, qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées. Ce procédé peut être réitéré plusieurs fois, on filtre les features-maps obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles features-maps à normaliser et redimensionner, et qu'on peut filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières features-maps sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second. Le second bloc n'est pas caractéristique d'un CNN : il se retrouve en fait à la fin de tous les réseaux de neurones utilisés pour la classification. Les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires et fonctions d'activation) pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur contient autant d'éléments qu'il y a de classes : l'élément  $i$  représente la probabilité que l'image appartienne à la classe  $i$ . Chaque élément est donc compris entre 0 et 1, et la somme de tous vaut 1. Ces probabilités sont calculées par la dernière couche de ce bloc (et donc du réseau), qui utilise une fonction logistique (classification binaire) ou une fonction softmax (classification multi-classe) comme fonction d'activation [22].

#### **2.4.3.3.4. Avantage des réseaux neurones convolutifs**

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du réseau de neurones convolutifs par rapport au perceptron multicouches, qui, lui, considère chaque neurone indépendant et affecte donc un poids différent à chaque signal entrant [23].

#### **2.4.3.3.5. Les inconvénients des réseaux neuronaux convolutifs**

- Les CNN peuvent être coûteux en calcul et nécessiter une grande puissance de traitement, ce qui peut les rendre difficiles à former et à déployer sur des appareils bas de gamme.
- Ils peuvent être sujets au sur-ajustement, en particulier lorsqu'ils sont entraînés sur de petits ensembles de données ou avec une régularisation insuffisante.
- Les CNN peuvent être difficiles à interpréter, car les caractéristiques apprises sont souvent abstraites et difficilement compréhensibles par les humains.
- Ils nécessitent une grande quantité de données étiquetées pour la formation, ce qui peut être long et coûteux à obtenir.
- Les CNN peuvent ne pas fonctionner correctement sur des données très différentes des données d'entraînement, ce qui peut limiter leur capacité de généralisation.

## **2.5. Conclusion**

Dans ce chapitre nous avons vu que les réseaux de neurones convolutifs (CNN) sont de puissants algorithmes d'apprentissage en profondeur qui ont révolutionné le domaine de la vision par ordinateur. Ils sont capables d'extraire des caractéristiques d'images et de vidéos et peuvent être utilisés pour diverses tâches telles que la classification d'images, la détection d'objets et la segmentation. Les CNN se sont avérés très efficaces dans de nombreuses applications et sont maintenant largement utilisés dans l'industrie et la recherche.

# **CHAPITRE 3**

Méthodologie et matériels

## 3.1. Introduction

Notre objectif est de développer un système basé sur un microcontrôleur capable de détecter automatiquement les défauts dans les générateurs photovoltaïques à partir des images infrarouges. Dans ce chapitre, nous allons présenter les différentes étapes de conception et de la réalisation pratique de ce système embarqué.

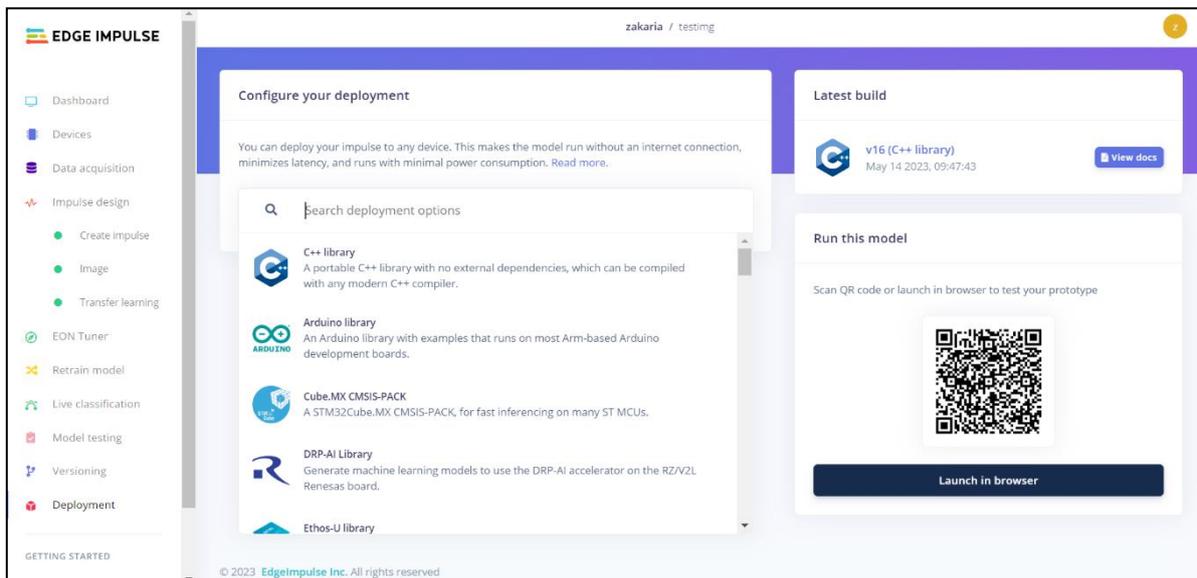
## 3.2. Ressources

Nous avons utilisé une combinaison d'outils, de logiciels et de matériels qui rend possible la conception de notre système.

### 3.2.1. Ressources logicielles

- **La plate-forme Edge Impulse**

Pour développer notre modèle nous avons utilisé la plate-forme Edge Impulse (Voir Fig.3.1), C'est une plateforme de développement pour le ML sur les systèmes embarqués à base d'un microcontrôleur, fondée en 2019 par Zach Shelby et Jan Jongboom [24]. Edge Impulse offre une efficacité et une vitesse maximales sur une large gamme de matériels, des microcontrôleurs aux processeurs.



**Figure 3.1.** La plateforme Edge Impulse

- **IDE Arduino**

L'IDE Arduino (Environnement de développement intégré), c'est un logiciel open source qui facilite l'écriture de code et son téléchargement sur la carte, et facilite l'utilisation et l'organisation des bibliothèques (Voir Fig.3.2). L'IDE Arduino est fournie

avec une interface utilisateur conviviale qui facilite le développement pour les débutants, il prend en charge une variété de langages de programmation, y compris C et C++ et est compatible avec une large gamme de cartes Arduino.

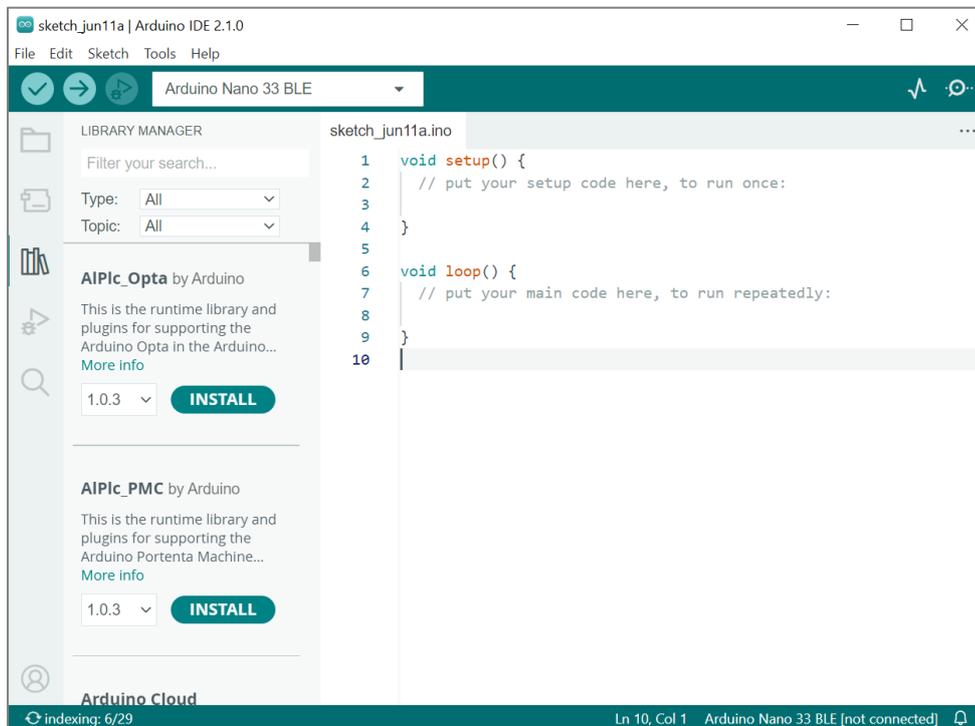


Figure 3.2. L'interface de l'IDE Arduino

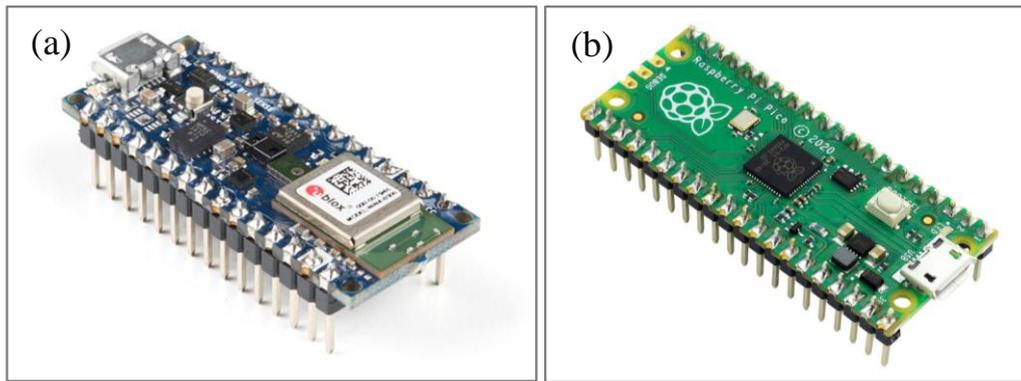
### 3.2.2. Ressources matérielles

Notre système se compose de deux parties, une partie qui prend des images IR et effectue le diagnostic, dans cette partie nous avons utilisé :

- **Une carte à microcontrôleur**

Nous pouvons déployer notre modèle Edge Impulse sous la forme d'une bibliothèque Arduino qu'il peut exécuter sur presque toutes les cartes Arduino Arm, nous pouvons également le déployer sous la forme d'une bibliothèque C/C++ qu'il peut exécuter dans n'importe quel microcontrôleur. Mais il faut choisir un microcontrôleur avec au moins 64 Ko de RAM.

Nous avons choisi la carte Arduino Nano 33 BLE Sense (256 Ko de RAM, et un processeur ARM Cortex-M4 32bit cadencé à 64MHz) pour tester la bibliothèque Arduino, et nous avons choisi la carte Raspberry PI pico (264 Ko de RAM, et un processeur ARM Cortex M0+ dual-cœur cadencé à 133MHz) pour tester la bibliothèque C/C++ (Voir Fig.3.3).



**Figure 3.3.** La carte a microcontrôleur : a) Arduino Nano 33 BLE Sense, et b) Raspberry PI pico

- **Une caméra IR**

Dans notre application, nous voulons avoir une détection automatique en temps réel, pour ce faire, nous avons besoin d'une caméra thermique que nous pouvons utiliser avec les deux cartes sélectionnées. Nous avons choisi la caméra thermique développée par Melexis (Voir Fig.3.4), elle est conçue pour détecter le rayonnement infrarouge émis par les objets et le convertir en relevés de température. Elle dispose un réseau des capteurs thermiques  $32 \times 24$  capables de mesurer des températures de  $-40^{\circ}\text{C}$  à  $300^{\circ}\text{C}$  avec une précision de  $\pm 1^{\circ}\text{C}$ . Elle peut communiquer avec la carte a microcontrôleur par le protocole I2C.

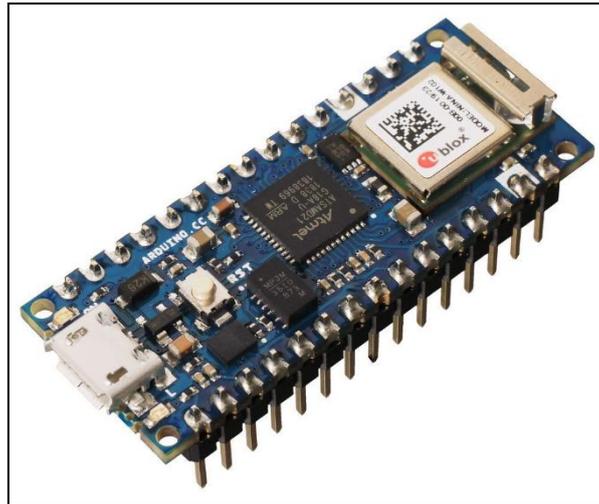


**Figure 3.4.** La caméra thermique

La deuxième partie qui enregistre les résultats du diagnostic dans une base de données, dans cette partie que nous avons utilisée :

- **Arduino Nano 33 IoT**

L'Arduino Nano 33 IoT est une carte de développement conçu pour les applications Internet des objets (IoT) (Voir Fig.3.5), elle est compatible avec l'écosystème Arduino, ce qui le rend facile à utiliser pour les utilisateurs débutants et avancés. Nous avons choisi cette carte car elle dispose d'un module WiFi u-blox NINA-W102 qui nous permettra de sauvegarder les résultats des diagnostics dans une base de données.



**Figure 3.5.** La carte Arduino Nano 33 IoT

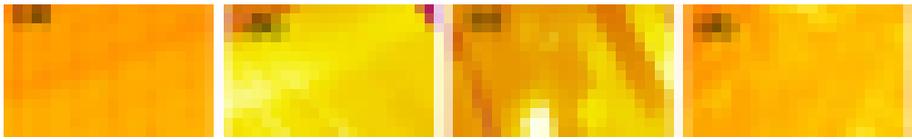
### 3.3. Méthodologie

Pour construire notre application, nous allons suivre les étapes suivantes :

- 1) Préparation de la base de données.
- 2) Développement du model de la détection et la classification des problèmes.
- 3) Développement et construction du circuit électronique.
- 4) Déploiement du modèle.
- 5) Développement d'une interface graphique (GUI).

#### 3.3.1. Préparation de la base de données

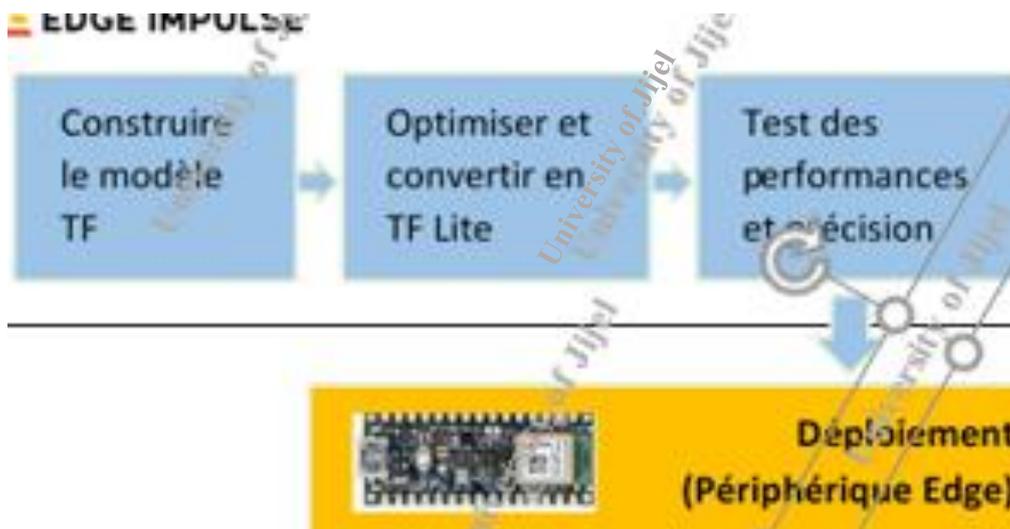
La base de données utilisée se compose de 2000 IR images de thermographie collectées pour les modules PV dans des conditions normales et défectueuses [7]. Les modules PV présentent quatre classes de défauts : état bonne (N), sale (D2), dégradé (D3) et défaut de poussière sur le module PV (D1), comme illustré à la Figure 3.6.



**Figure 3.6.** IR images de thermographie : a) état bonne (N), b) sale (D2), c) dégradé (D3) et d) défaut de poussière sur le module PV (D1)

### 3.3.2. Développement de modèle

Notre system vise à appliquer un réseau neuronal convolutif profond (CNN) pour classer certains défauts sur les modules photovoltaïques et construire le modèle développé pour une application en temps réel. Pour ce faire, nous utilisons une plate-forme ouverte de ML nommée Edge Impulse. Pour développer le modèle, nous utilisons un MobilNetV2 de type CNN. En ce qui concerne les périphériques Edge, figure 3.7 représente un flux de travail de la procédure d'intégration sur la carte Arduino Nano 33 BLE Sense [25].



**Figure 3.7.** La procédure d'intégration [25]

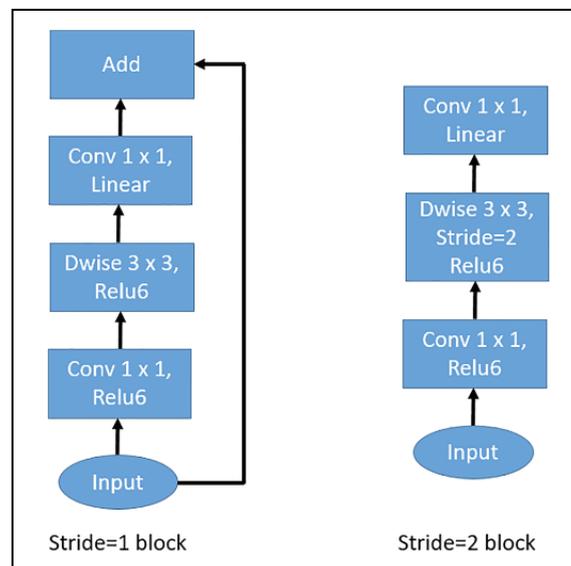
Cela comprend quatre blocs [25] :

- i) Construire le modèle TensorFlow (TF),
- ii) Optimiser et convertir le modèle en TF Lite,

- iii) Tester les performances du modèle,
- iv) Intégrer le modèle à l'intérieur du périphérique Edge mentionné et les déployer.

### 3.3.2.1. Le réseau de neurones utilisé (MobileNetV2)

MobileNetV2 est l'un des réseaux de neurones convolutifs (CNN) les plus couramment utilisés et les plus légers (Voir Fig.3.8). Il est conçu pour les images et peut être utilisé à la fois pour la classification et la génération de fonctionnalités génériques. Dans ce réseau, des convolutions  $3 \times 3$  et  $1 \times 1$ , des opérations Bottleneck et une mise en commun moyenne sont appliquées. MobileNetV2 comporte 154 couches. Le MobileNetV2 utilise 3.4 Millions de paramètres, ce qui est inférieur à d'autres modèles CNN couramment préférés [26].



**Figure 3.8.** La structure de base de MobileNetV2 [26]

### 3.3.2.2. Procédure de développement d'un modèle sur Edge Impulse

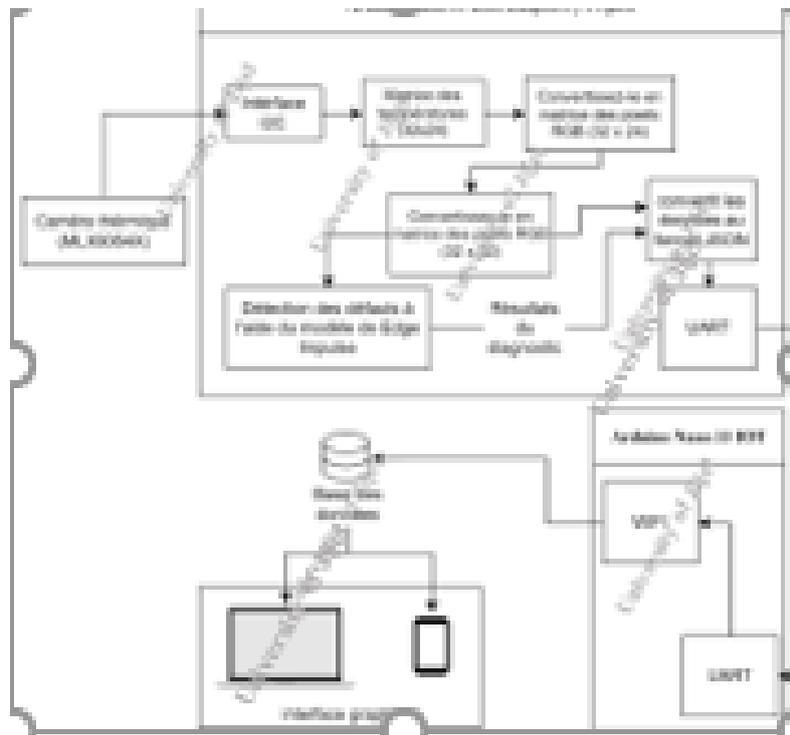
Nous pouvons résumer la procédure de développement d'un modèle sur la plate-forme Edge Impulse dans les étapes suivantes :

- 1) Acquisition de données : elle consiste à télécharger et étiqueter nos images IR en quatre classes (D1, D2, D3 et N).
- 2) Conception de l'Impulsion : elle comprend trois étapes :
  - a) Créer une impulsion : paramétrez la taille des images, ajoutez un bloc de traitement et un bloc d'apprentissage.

- b) Image : dans cette étape, nous sélectionnons le type de l'image IR (RGB ou Grayscale), et nous générons les caractéristiques des classes.
  - c) Apprentissage par transfert : ici, nous définissons les paramètres CNN (nombre de cycle d'entraînement, taux d'apprentissage, ensemble de validation, nombre de neurones dans la couche finale et valeur d'abandon). Une fois ces paramètres définis, nous entraînons le modèle.
- 3) Réapprentissage du modèle : cette étape vise à ré-entraîner le modèle avec des paramètres connus.
  - 4) Classification en direct : dans cette étape, nous testons le modèle avec de nouvelles données (images IR).
  - 5) Test de modèle : définissez le résultat attendu pour chaque image IR sur le résultat souhaité pour attribuer automatiquement un score pour l'impulsion.
  - 6) Déploiement : cette étape permet au modèle de s'exécuter sans connexion Internet, minimise la latence et s'exécute avec une consommation d'énergie minimale. Ensuite, nous devons sélectionner la carte matérielle (par exemple, Nano 33 BLE Sense) pour construire un modèle optimisé pour une application en temps réel (firmware). Autrement, nous pouvons créer une bibliothèque et cela peut transformer notre impulsion en code source optimisé pour être exécuté sur n'importe quel appareil.

### 3.3.3. Développement et réalisation du circuit électronique

Comme la figure 3.9 montre le principe de fonctionnement de notre système développé.

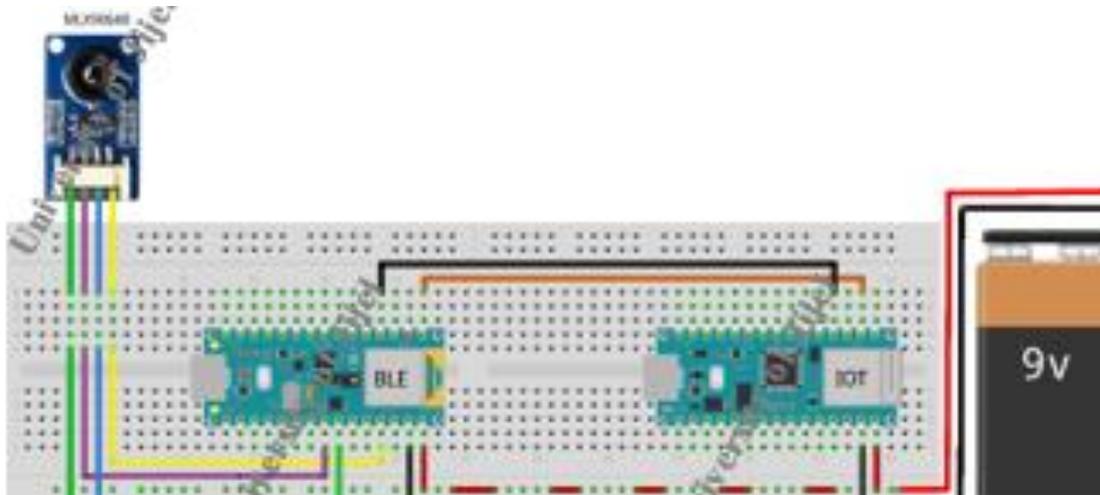


**Figure 3.9.** Schéma de fonctionnement de système embarqué développé

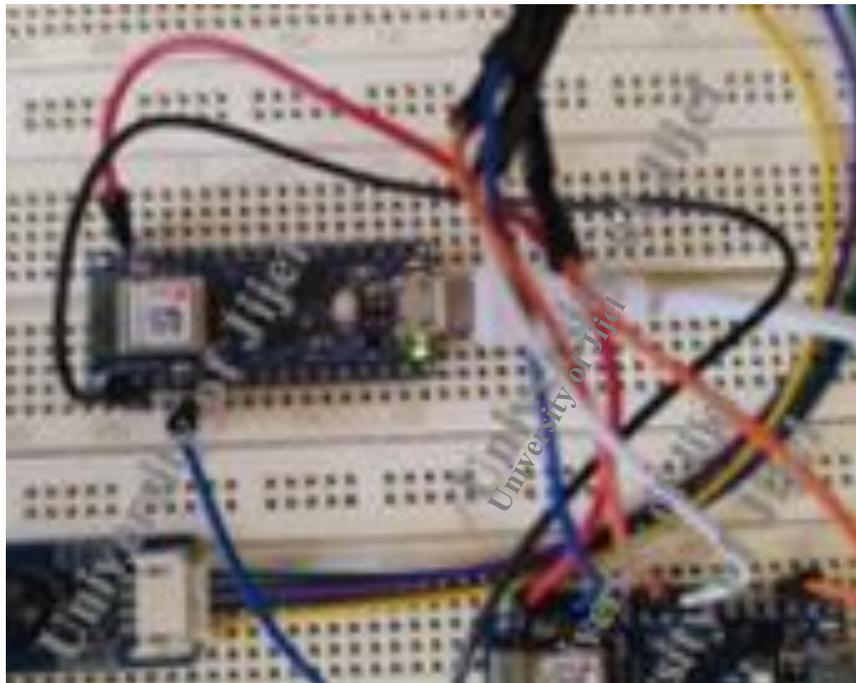
Nous pouvons résumer le fonctionnement de notre système dans les étapes suivantes :

- **Etape 1** : lit les données des capteurs IR via le protocole I2C.
- **Etape 2** : génère une matrice de températures  $32 \times 24$  (les calculs sont effectués selon la fiche technique de la caméra).
- **Etape 3** : convertissons les températures en couleurs en utilisant l'interpolation linéaire.
- **Etape 4** : redimensionnons la matrice des couleurs à  $32 \times 32$  (nous avons simplement ajouté des pixels noirs) pour l'adapter au model générer par Edge Impulse.
- **Etape 5** : exécuter le model générer par Edge Impulse avec la matrice des couleurs comme entrée.
- **Etape 6** : convertir les résultats de diagnostic générer par le model au format JSON, et les envoyer par Serial a la carte Arduino Nano 33 IoT.
- **Etape 7** : sauvegarder les résultats dans une base de données.

La figure 3.10 montre le prototype du circuit électrique proposer pour la réalisation de notre système (à base de l'Arduino Nano 33 BLE Sense), et la figure 3.11 montre la réalisation du circuit sur une plaque d'essai (Breadboard).

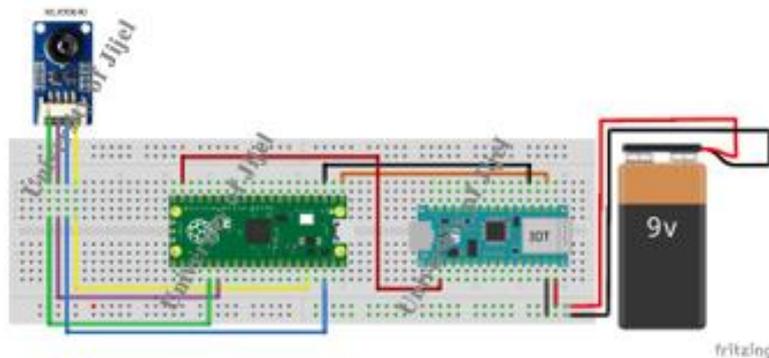


**Figure 3.10.** Le schéma électrique du système (à base de l'Arduino Nano 33 BLE Sense)

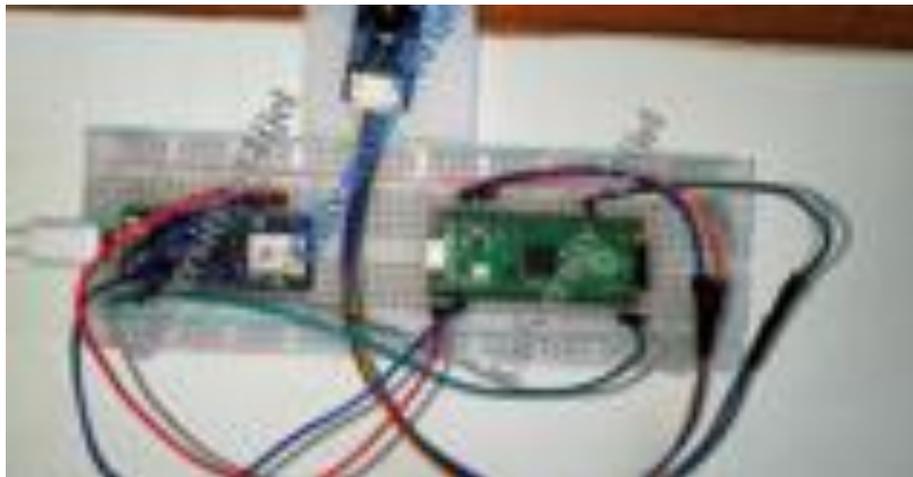


**Figure 3.11.** Le circuit électrique réalisé sur une Breadboard (à base de l'Arduino Nano 33 BLE Sense)

La figure 3.12 montre le prototype du circuit électrique proposé pour la réalisation de notre système (à base de la Raspberry PI pico), et la figure 3.13 montre la réalisation de circuit sur une plaque d'essai (Breadboard).



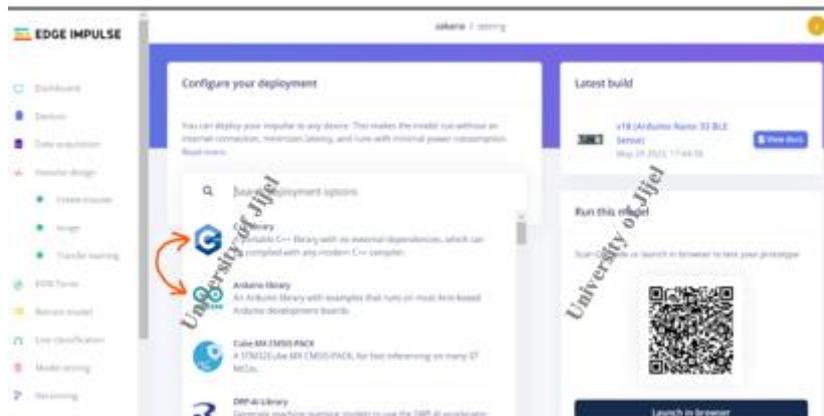
**Figure 3.12.** Le schéma électrique du système (à base de la Raspberry PI pico)



**Figure 3.13.** Le circuit électrique réalisé sur une Breadboard (à base de la Raspberry PI pico)

#### 3.3.4. Déploiement du modèle

Notre objectif est d'exécuter le modèle généré par Edge Impulse sur un microcontrôleur, donc dans la page de déploiement de la plate-forme Edge Impulse, nous choisissons la bibliothèque Arduino (pour l'exécuter dans l'Arduino Nano 33 BLE Sense), et la bibliothèque C/C++ (pour l'exécuter sur le Raspberry PI pico). Après avoir téléchargé les bibliothèques, nous pouvons les utiliser dans notre code comme n'importe quelle autre bibliothèque régulière (Voir Fig.3.14).



**Figure 3.14.** La page de déploiement de la plate-forme Edge Impulse

Pour compiler et téléverser le code d'Arduino Nano 33 BLE Sense nous utilisons l'IDE Arduino, et pour le Raspberry PI pico nous utilisons l'officiel pico C/C++ SDK, et le compilateur GCC et l'éditeur du code VS Code.

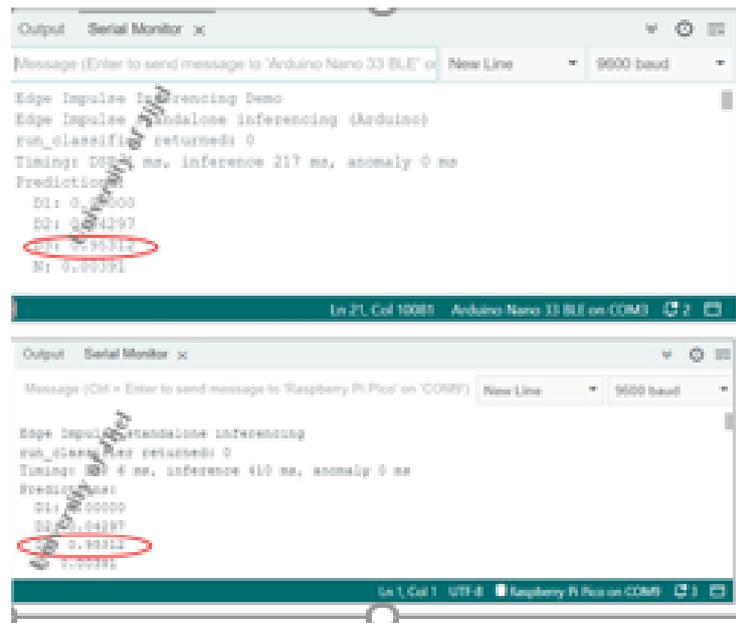
### 3.3.5. Test du modèle sur les deux cartes

Après avoir exporté le modèle développé sur Edge Impulse sous forme d'une bibliothèque, nous devons le tester pour voir s'il donne les mêmes résultats que la simulation. Pour cela, nous sélectionnons une image inconnue (parmi les images utilisées pour la validation de modèles), comme le montre la Fig.3.15, nous utilisons les caractéristiques brutes (raw features) de l'image comme entrée pour notre modèle.



**Figure 3.15.** Résultats de classification pour l'image IR choisie

Après avoir exécuté le code sur les deux cartes (l'Arduino Nano 33 BLE Sense et le Raspberry PI Pico), les résultats de la classification sont présentés dans la figure suivante.



**Figure 3.16.** Résultats de la classification sur les deux cartes

Comme on peut le voir sur la Fig.3.16, les résultats de classification dans les deux cartes sont les mêmes (l'Arduino et le Raspberry), 95.3 % pour D3 et 4.2 % pour D2 et 0.3 % pour D1 et N, donc les deux cartes classent l'image dans la même classe D3 ce qui est le même résultat que la simulation, et ce qui signifie que notre modèle fonctionne correctement.

### 3.3.6. Base de données (Firestore)

Nous avons choisi d'utiliser la base de données offerte par Firebase, Realtime-Database, Ce choix a été fait en raison de grand nombre des plates-formes supportées (mobile, desktop, web), et parce que le processus d'intégration est facile et rapide par rapport aux autres choix.

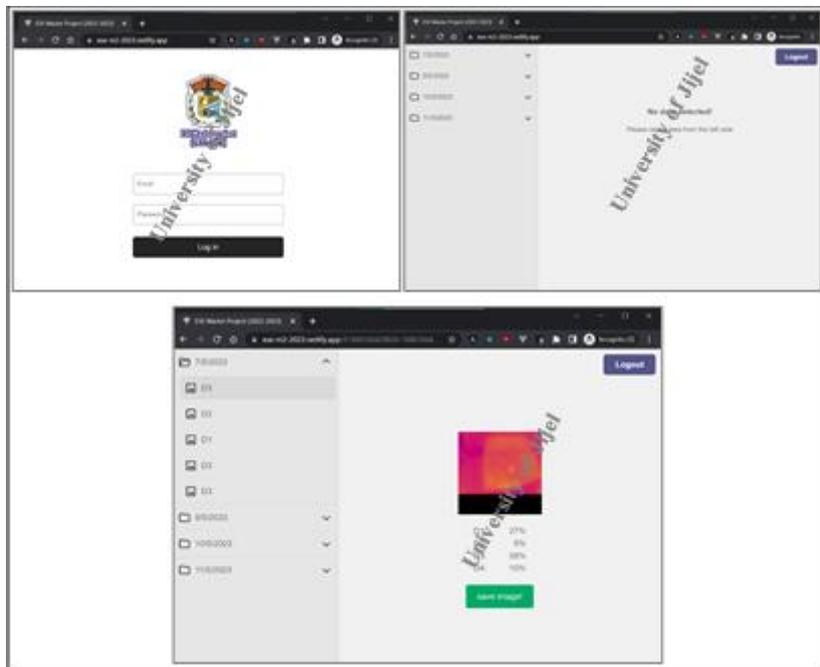


**Figure 3.17.** La base de données (Realtime Database)

Dans la base de données nous allons sauvegarder les images collectées par la caméra IR en temps réel et les résultats de classification de chaque image. Les données sont organisées par la date de collecte.

### 3.3.7. Développement d'une interface graphique

Afin de pouvoir vérifier et visualiser les données collectées par la caméra IR, nous avons besoin d'une interface graphique qui permet de vérifier les données collectées à partir de la base de données. Pour ce faire, nous avons choisi de développer une application Web, qui supporte presque toutes les plateformes (ordinateurs ou smartphones). Ainsi que la possibilité de la consulter à tout moment, sans avoir à installer quoi que ce soit juste un navigateur Web (Voir Fig.3.30).



**Figure 3.18.** L'interface graphique développée

Comme la Fig.3.18 montre, notre Web application est constitué de deux pages, la première page est destinée à l'espace de connexion pour l'administrateur (l'accès aux données est protégé par un mot de passe), et la deuxième page affiche les données collectées de manière organisée et facile pour naviguer les données collectées à partir de la base de données.

Nous avons utilisé Reactjs pour développer la partie front-end de notre application et pour le back-end nous avons utilisé Firebase.

### **3.4. Conclusion**

Dans ce chapitre nous avons présenté les ressources (logiciels et matériels) utilisées, à savoir le Edge Impulse, l'IDE Arduino, les cartes à microcontrôleurs (Raspberry PI pico, Arduino Nano 33 BLE Sense, Nano 33 IoT). Nous avons décrit également notre base de données utilisée pour développer notre modèle. Aussi nous avons exposé les différentes étapes de déploiement du modèle sur les microcontrôleurs en question. Une interface graphique, afin de visualiser les résultats en ligne, a été développée. Dans le chapitre suivant nous allons entamer la partie des résultats obtenus durant ce projet.

# **CHAPITRE 4**

Résultats et interprétation

## 4.1. Introduction

Dans ce chapitre nous allons présenter et interpréter les résultats de simulation, obtenu sur la plate-forme Edge Impulse, ainsi que les résultats pratiques de notre système. Aussi, nous allons faire une petite comparaison entre les deux cartes (l'Arduino Nano BLE Sense et la Raspberry PI pico).

## 4.2. Résultats de la simulation

Pour développer notre modèle nous avons divisé notre base de données en deux parties : la première partie pour l'entraînement du modèle (80%) et la deuxième partie pour tester et valider (20 %). Après plusieurs expériences et variations des hyper-paramètres (c-à-d, nombre de cycle, taux d'apprentissage, dropout, nombre de neurones dans la couche intermédiaire) du modèle MobileNetV2, et cela pour avoir une très bonne performance (matrice de confusion). Le tableau suivant montre les paramètres optimaux obtenu de notre modèle. Il est à noter que notre modèle a été entraîné et développé à l'aide de la plate-forme Edge Impulse.

**Tableau 4.1.** Paramètres du modèle

Paramètres du modèle MobileNetV2	Valeur
Nombre de cycles d'entraînement	30
Taux d'apprentissage	0.0006
Dropout	0.2
Nombre de neurones à la couche intermédiaire	16

Afin d'évaluer les performances du modèle, nous avons calculé la matrice de confusion, y compris, la précision, la perte (loss), et le F1-score. Pour effectuer ces calculs, les équations suivantes sont utilisées :

$$Accuracy = \frac{\sum_i MC(i,i)}{\sum_i \sum_j MC(i,j)} \quad (4.1)$$

$$Precision_i = \frac{MC(i,i)}{\sum_j MC(j,i)} \quad (4.2)$$

$$Recall_i = \frac{MC(i,i)}{\sum_j MC(i,j)} \quad (4.3)$$

$$F1-score = 2 \frac{(sensitivity \times Reacll)}{(sensitivity + Precision)} \quad (4.4)$$

#### 4.2.1. Performances du modèle durant l'entraînement

La matrice de confusion obtenue du modèle durant l'entraînement est présentée dans le tableau suivant :

**Tableau 4.2.** Matrice de confusion (les performances du modèle durant l'entraînement)

La Classe	D1 (%)	D2 (%)	D3 (%)	N (%)	Taux de classification (Accuracy) (%)	Perte (Loss)
<b>D1</b>	98.7	1.3	0	0	96.20	0.11
<b>D2</b>	0	92.6	1.2	6.2		
<b>D3</b>	0	1.5	98.5	0		
<b>N</b>	0	4.4	0	95.6		
<b>F1-score</b>	99	93	98	95		

Comme le montre le tableau 4.2, le taux de classification (Accuracy) est 96.2 % et la perte (Loss) est 0.11, et le F1-score se situe entre 93 % et 99 %. Dans l'ensemble, ces résultats sont bons en termes de précision.

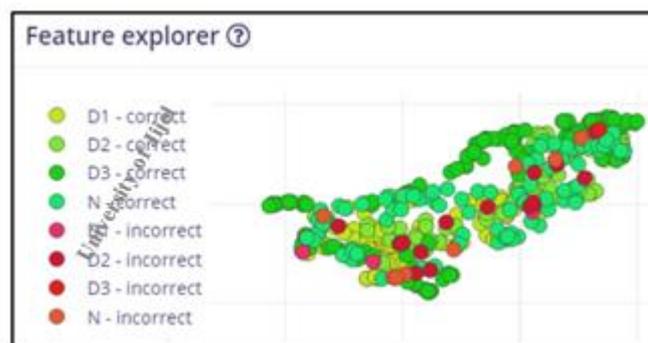
#### 4.2.2. Performance du modèle durant la phase de validation

Pour vérifier l'efficacité de ce modèle, nous avons calculé la matrice de confusion pendant le processus de validation en utilisant des images IR inconnues. Le tableau 4.3 présente la matrice de confusion de la performance pendant le processus de validation du modèle.

**Tableau 4.3.** Matrice de confusion (performance du modèle durant la validation)

La classe	D1 (%)	D2 (%)	D3 (%)	N (%)	Incertitude (%)	Taux de classification (Accuracy) (%)
<b>D1</b>	96.8	1.1	0	0	2.1	93.45
<b>D2</b>	1.0	85.1	1.0	6.9	5.9	
<b>D3</b>	0	0	99.1	0	0.9	
<b>N</b>	0	5.8	0	92.2	1.9	
<b>F1-score</b>	98	89	99	93		

La précision est de 93.45 % et le score F1 se situe entre 93 % et 99 %. Sur la base de l'incertitude et du F1-score, la classe D3 peut être classée correctement avec une précision de 99.1 %. La classe D2 à une valeur d'incertitude élevée, et c'est la classe la plus mal classée avec une précision de 85.1 %. Les caractéristiques explorées sont présentées à la figure 4.1.

**Figure 4.1.** Explorateur des caractéristiques des 4 classes durant la phase de la validation

La figure 4.2 montre un exemple de classification en direct. Une image inconnue, un boîtier de module PV sale (classe D2), a été sélectionnée pour vérifier la capacité du modèle à classer cette image.



**Figure 4.2.** L'image IR sélectionner pour tester le modèle

Les résultats sont présentés dans le tableau 4.4. Comme on peut le voir, le modèle classe l'image IR donnée avec une bonne précision. La classe prédite est D2, correspond aux modules PV sales, comme prévu.

**Tableau 4.4.** Validation du modèle avec une image inconnue

<b>Nom</b>	D4_387
<b>Résultats attendus</b>	D2
<b>Classe</b>	
<b>D1</b>	0
<b>D2</b>	1
<b>D3</b>	0
<b>N</b>	0
<b>Incertain</b>	0

#### 4.2.3. Optimisation du modèle

Afin de minimiser les ressources matérielles et la latence, un processus d'optimisation du modèle proposé a été réalisé. À cet égard, une comparaison entre le modèle optimisé et le modèle non optimisé est présentée dans le tableau 4.5 où l'utilisation de la RAM (KB), l'utilisation de la mémoire flash (MB) et la latence (ms) pour chaque modèle ont été évaluées. Le pourcentage de réduction est calculé entre le modèle optimisé et le modèle non optimisé.

**Tableau 4.5.** Comparaison entre le modèle optimiser et non optimiser

<b>Optimiseur</b>	<b>Utilisation de RAM (KB)</b>	<b>Utilisation de la mémoire flash (MB)</b>	<b>Temps de latence (ms)</b>
<b>Modèle non optimisé (float32)</b>	474.9	1.6	6,600
<b>Modèle optimisé quantifié (Int8)</b>	225.4	0.58	904
<b>Réduction</b>	52 %	63 %	86 %

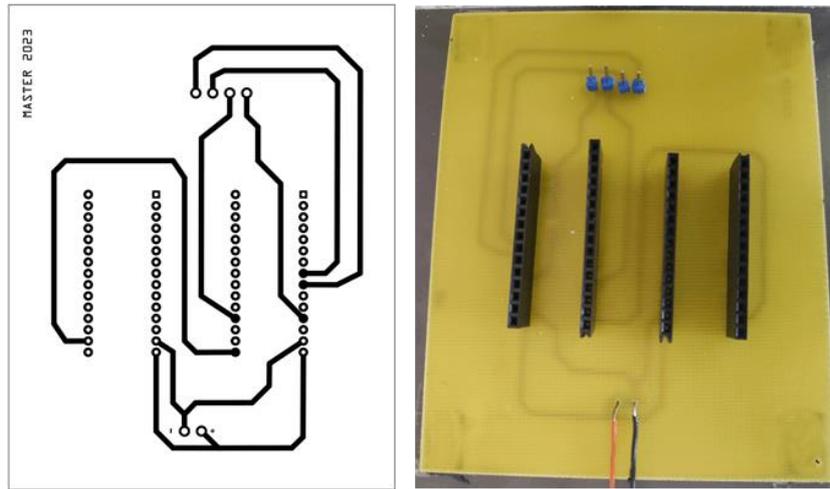
Comme le montre le tableau 4.5, la RAM du modèle optimisé a été réduite par rapport au modèle non optimisé, avec un pourcentage de 52 %. L'utilisation de la mémoire flash est également réduite à 0.58 Mo, soit une réduction en pourcentage de 63 %. La latence est également réduite de manière significative, 904 ms contre 6600 ms pour le modèle non optimisé. Cela signifie que le code optimisé peut être exécuté environ 7 fois plus rapidement que le code non optimisé.

### 4.3. Résultats pratique

Dans cette partie nous allons présenter le circuit final réaliser de notre system et les résultats pratiques, telle que les performances du modèle sur les deux cartes (Arduino nano 33 BLE et la Raspberry PI pico).

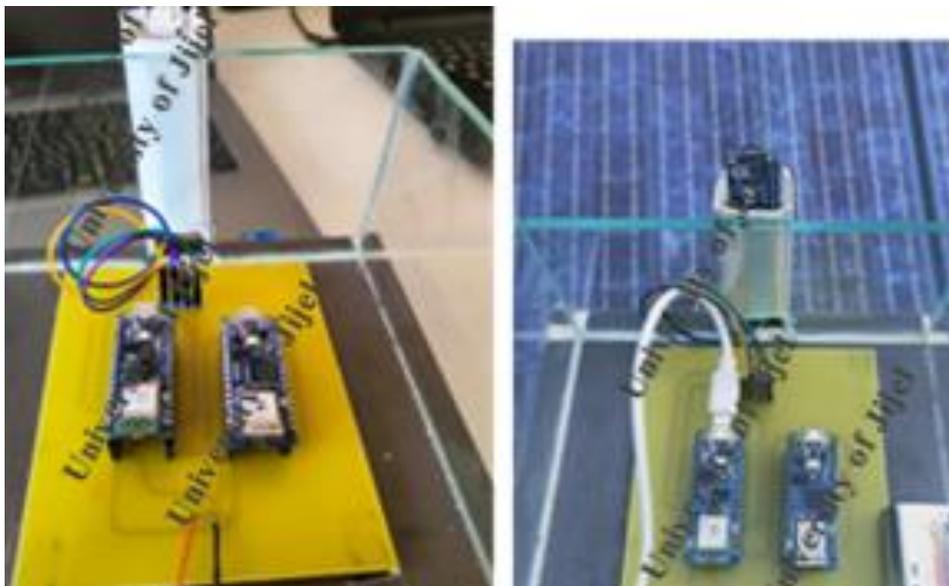
#### 4.3.1. Système réalisé

Pour se débarrasser de tous les fils électriques, comme on peut voir dans la figure 3.11, et pour éliminer tous les problèmes de connexion nous avons développé un circuit imprimé (PCB).



**Figure 4.3.** Circuit imprimée (PCB)

Après la réalisation du circuit imprimé, on a mis les composants à leur place dans le PCB, le résultat final est présenté dans la figure suivante :



**Figure 4.4.** Photo de circuit réalisé

Afin de tester notre système, en temps réel, nous avons développé un serveur Web local à l'aide de l'Arduino Nano 33 IoT (sur l'adresse <http://192.168.190.63>), qui affiche une simple

page Web contenant un image IR direct de la caméra et les résultats de classification générés par le modèle qui est exécuté dans le Arduino Nano 33 BLE sens, les données affichées sur la page sont automatiquement rafraîchies deux fois chaque second. La figure suivante montre le résultat :



**Figure 4.5.** Test direct des résultats de classification

#### 4.3.2. Etude comparative

Dans cette section nous allons présenter une comparaison entre les deux circuits développés, de point de vue, qualité d'image, tempe de réponse, les ressources utilisées (mémoire Flash).

Après avoir téléchargé le modèle développé à partir de la plateforme Edge Impulse sous forme de bibliothèque Arduino et de bibliothèque C/C++, ces deux bibliothèques ont été utilisées pour développer le code principal de notre système. Nous avons créé deux versions, la première utilise la bibliothèque Arduino pour l'Arduino Nano BLE Sense, et la seconde utilise la bibliothèque C/C++ pour la Raspberry PI pico. Le tableau suivant montre les caractéristiques principales de chaque code :

**Tableau 4.6.** Caractéristiques principales de chaque code

<b>Microcontrôleur</b>	<b>Taille de programme (KB)</b>	<b>Temps de réponse (ms)</b>
<b>Arduino Nano 33 BLE Sense</b>	699	218
<b>Raspberry PI pico</b>	1315	410

D'après le tableau 4.6 on peut voir que le temps de réponse (temps de classification) du code de l'Arduino Nano 33 BLE Sense est 218 (ms), c'est presque deux fois plus rapide que la Raspberry PI pico. Cette grande différence en vitesse est due à ce fait que l'Arduino Nano 33 BLE Sense utilise une bibliothèque optimisée spécialement pour les cartes Arduino, par contre la carte Raspberry PI pico utilise une bibliothèque générale n'est pas optimisée à un microcontrôleur ou une carte spéciale.

#### **4.4. Conclusion**

Dans ce chapitre nous avons tout d'abord présenté et interprété les résultats de simulation. On a constaté qu'une précision de 96.20 % a été obtenue durant la phase d'entraînement et de 93.45 % durant la phase de validation, ce qui signifie que ce modèle a besoin d'une éventuelle amélioration.

En ce qui concerne les résultats pratiques, nous avons clairement justifié la faisabilité de notre approche de diagnostic des générateurs photovoltaïques à l'aide d'un système à base d'un microcontrôleur, dédié spécialement aux applications embarquées.

# **Conclusion générale**

Dans ce mémoire, une solution embarquée pour le diagnostic des défauts des modules/générateurs PV, basée sur des images thermographiques (IR) a été réalisée. L'idée est d'intégrer le modèle MobileNetV2 dans un microcontrôleur à faible consommation et à coût raisonnable (~43 \$) (Nano 33 BLE Sense). Ce système embarqué a été développé pour une application en temps réel, dans le but de détecter et de classer quatre classes de défauts différentes.

Les résultats pratiques obtenus ont montré la faisabilité du système pour le diagnostic des générateurs PVs, avec une précision de classification moyenne de 93,4 %, conduisant à une bonne classification des classes de défauts considérées. De plus, le modèle a été optimisé en tenant compte de l'utilisation du matériel disponible et des temps de latence, ce qui se traduit par une réduction de la taille de la RAM du modèle jusqu'à 52 %, une réduction de l'utilisation du flash de 63 % et une réduction de la latence de 86 %. Cela montre comment le processus d'optimisation peut conduire à une amélioration significative des performances d'exécution.

L'utilisation d'une carte Arduino Nano 33 BLE Sense était plus simple et mieux en point de vue des ressources disponibles, par contre la carte Raspberry Pi Pico est moins chère et sa difficulté réside dans la non disponibilité des ressources (bibliothèques optimisées). En effet, la carte Arduino en question dispose d'une bibliothèque spécifique développée par Edge Impulse ce qui fait une énorme amélioration du temps de classification. Cependant, dans le cas de la carte Raspberry Pi Pico on a pris beaucoup de temps afin d'écrire un code pour faire fonctionner la caméra.

Afin d'améliorer les résultats de classification de notre solution embarquée nous proposons comme perspectives d'utiliser une caméra plus performante (résolution de 96x96 minimum) et aussi un microcontrôleur avec une RAM suffisante pour traiter les images obtenues à partir de la caméra IR.

## Références

- [1]. IEA (2022), Solar PV, IEA, Paris <https://www.iea.org/reports/solar-pv>, License: CC BY 4.0.
- [2]. M. Alajmi, K. Awedat, M. S. Aldeen and S. Alwagdani, "IR Thermal Image Analysis: An Efficient Algorithm for Accurate Hot-Spot Fault Detection and Localization in Solar Photovoltaic Systems", 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 162-168.
- [3]. V. Lofstad-Lie, E. S. Marstein, A. Simonsen, and T. Skauli, "Cost-Effective Flight Strategy for Aerial Thermography Inspection of Photovoltaic Power Plants", *IEEE Journal of Photovoltaics*, vol. 12, pp. 1543–1549, 2022.
- [4]. N. Prajapati, R. Aiyar, A. Raj and M. Paraye, "Detection and Identification of faults in a PV Module using CNN based Algorithm", 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-5.
- [5]. D. Rocha, M. Lopes, J. P. Teixeira, P. A. Fernandes, M. Morais and P. M. P. Salome, "A Deep Learning Approach for PV Failure Mode Detection in Infrared Images: First Insights", IEEE 49th Photovoltaics Specialists Conference (PVSC), Philadelphia, PA, USA, 2022, pp. 0630-0632.
- [6]. A. Mellit, "An embedded solution for fault detection and diagnosis of photovoltaic modules using thermographic images and deep convolutional neural networks", *Engineering Applications of Artificial Intelligence*, vol. 116, pp. 105459, 2022.
- [7]. N. Kellil, A. Aissat, and A. Mellit, "Fault diagnosis of photovoltaic modules using deep neural networks and infrared images under Algerian climatic conditions". *Energy*, vol.263, pp.125902, 2023.
- [8]. <https://store.arduino.cc/products/arduino-nano-33-ble-sense> [Consulté le 07 Avril 2023].
- [9]. <https://www.raspberrypi.com/products/raspberry-pi-pico/> [Consulté le 07 Avril 2023].
- [10]. C. Moussaoui, "Conception et réalisation d'un système semi-automatique de diagnostic des générateurs photovoltaïques à l'aide de l'apprentissage profond et de l'imagerie thermique", mémoire de Master, Université Mohamed Seddik Benyahia Jijel, 2022.

- [11]. C. Moussaoui, O. Herrak and A. Mellit, "Fault diagnosis of photovoltaic panels using a low-cost edge device, International Conference on Advanced Renewable Energy Systems", (ICARES'22), Algeria - December 18 -20, 2022.
- [12]. L. Bun, "Détection et localisation de défauts dans un système Photovoltaïque ", Thèse doctorat, Université de Grenoble, 2011.
- [13]. <https://www.orygeen.eu/docs-actus/glossaire/module-photovoltaique> [Consulté le 10 Avril 2023].
- [14]. <https://soleildelarc.com/wp-content/uploads/2021/02/SDA-composition-panneau.png> [Consulté le 10 Avril 2023].
- [15]. A. Mellit, G.M Tina, and S.A Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review ". *Renewable and Sustainable Energy Reviews*, vol.91, pp.1-17,2018.
- [16]. R. Khenfer, " Détection et isolation de défauts combinant des méthodes à base de données appliquées aux systèmes électro énergétique ", thèse doctorat, université de Ferhat Abbas Setif, 2015.
- [17]. L. Ayyoub, "Application des réseaux de neurones profonds pour la prédiction de la puissance produite par un micro-réseau, dédié au chargement des véhicules électriques ", Mémoire de master, Université Mohamed Seddik Ben Yahia-Jijel,2020.
- [18]. A. Mellit, and S.A Kalogirou, Machine learning and deep learning for photovoltaic applications. In *Artificial Intelligence for Smart Photovoltaic Technologies* (pp. 1-14). Melville, New York: AIP Publishing LLC, 2022.
- [19]. S. Soumia Douâa, "Techniques de NLP pour la détection des fausses nouvelles", Mémoire de master, Université Ibn Khaldoun - Tiaret, 2022.
- [20]. K. Sépéli Sodoké, "Extraction de connaissances à partir de traces multidimensionnelles de mouvements oculaires pour un transfert d'expertise perceptivo-décisionnelle dans le domaine médical", Thèse doctorat, Université du QUÉBEC à Montréal, Décembre 2020.
- [21]. K. Abdelmalek, "Représentations d'images pour la recherche et la classification d'images", Thèse de Doctorat, Université M'hamed Bougara-Boumerdes, 2021.
- [22]. P. Blanc-Durand, "Réseaux de neurones convolutifs en médecine nucléaire : applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM", Thèse de doctorat, Université Paris Descartes, 2018.

- [23]. A. Mellit, and S. Kalogirou, *Handbook of Artificial Intelligence Techniques in Photovoltaic Systems: Modeling, Control, Optimization, Forecasting and Fault Diagnosis*. Elsevier, 2022.
- [24]. <https://www.edgeimpulse.com/> [Consulté le 07 Avril 2023].
- [25]. A. Mellit, N. Blasuttigh and A. Massi Pavan, TinyML for fault diagnosis of Photovoltaic Modules using Edge Impulse Platform, ICSmartGrid2023, Paris, France, June 04-07, 2023.
- [26]. C. H., Karadal, M. C., Kaya, T., Tuncer, S., Dogan, and U. R. Acharya,. "Automated classification of remote sensing images using multileveled MobileNetV2 and DWT techniques". *Expert Systems with Applications*, vol.185, pp.115659.2021