



Mémoire de fin de cycle  
Pour l'obtention du diplôme master  
Option : Intelligence Artificielle

---

# Classification des arythmies cardiaques à partir d'images d'EKG en utilisant des techniques de machine learning

---

*Réalisé par :*  
Mlle. AFIA Sihem  
Mlle. ABBACI Mayssa

*Encadré par :*  
Pr. KACHENOURA Aldjia (CHU Bejaia)  
Pr. AMROUN Kamal (Université de Bejaia)

*Soutenu le 03 Juillet 2023, Devant le jury composé de :*  
Mme Samira BOUKERRAM (Présidente)  
Mme FatimaAIT HETRIT (Examineur)

Promotion : 2022/2023

# *Remerciements*

Tout d'abord, nous remercions le bon dieu qui nous a aidé à surmonter toutes les difficultés rencontrées au long de cette période pour mener à terme ce travail.

Nous tenons à remercier notre encadreur **Professeur Kamal Amroune** ; Professeur au département d'informatique à l'université de Béjaïa, pour ce grand honneur qu'il nous a accordé en acceptant de nous encadrer.

Nos remerciements aux membres de jury **Mme Boukerrame Samira** et **Mme Ait hetrit Fatima** qui nous font l'honneur d'examiner et juger notre travail.

Enfin, nous tenons à remercier également nos familles et toutes personnes qui de près ou de loin ont contribué à ce modeste travail.

# *Dédicaces*

Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime et qu'on remercie en exprimant la gratitude et la reconnaissance durant toute notre existence :

Je dédie ce modeste travail : A mes parents qui m'ont guidé durant les moments les plus pénibles de ce long chemin, mes parents qui ont été à mes cotés et m'ont soutenu durant toute ma vie, mes parents qui ont sacrifié toute leur vie afin de me voir devenir ce que je suis, merci papa, merci maman..

A mes chers frères et soeurs : Nassim ,karim, karima et anissa.

A mes chers neveux et nièces.

A membres du club scientifique et informatique.

A tous mes proches et amis.

En fin, je remercie mon amie mon binôme Mayssa qui a contribué à la réalisation de ce modeste travail.

# *Dédicaces*

Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime et qu'on remercie en exprimant la gratitude et la reconnaissance durant toute notre existence :

Je dédie ce modeste travail : A ma mère qui m'a guide durant les moments les plus pénibles de ce long chemin, ma mère qui a été a mes cotes et ma soutenu durant toute ma vie, ma mère qui a sacrifié toute sa vie afin de me voir devenir ce que je suis, merci ma maman merci pour ton amour. ..

A mon très cher père pour sa patience et tous ses efforts au guide qu'il à été à ses encouragements,ses valeurs inculqués,son soutien, son amour merci papa .

à mes grands frères Amine et Salah d'être toujours était là pour moi de m'avoir soutenu. et encourager.

merci à mes grands parents. Au membre de ma famille à mes cousins et cousines, à amis,

merci. A mon prommoteur Pr Kamal Amroune.

En fin, je remercie mon amie mon binôme Sihem qui a contribué à la réalisation de ce modeste travail.

# Table des matières

<b>Introduction générale</b>	<b>9</b>
<b>1 Cardiologie</b>	<b>10</b>
1.1 Introduction . . . . .	10
1.2 Physiologie cardiovasculaire . . . . .	10
1.2.1 Le coeur . . . . .	10
1.2.2 Fonctionnement électrique du coeur . . . . .	11
1.3 Principales pathologies cardiovasculaire . . . . .	11
1.3.1 Arythmie cardiaque . . . . .	12
1.3.2 Causes et facteurs de risque des arythmies . . . . .	14
1.3.3 Mécanisme d'arythmies cardiaques . . . . .	15
1.3.4 Symptômes et conséquences des arythmies . . . . .	16
1.3.5 Diagnostic . . . . .	16
1.4 Électrocardiographe . . . . .	16
1.5 Electrocardiogramme . . . . .	17
1.5.1 Types d'électrocardiogrammes . . . . .	17
1.6 Interprétation d'électrocardiogramme . . . . .	17
1.6.1 Ondes d'électrocardiogramme (ECG) . . . . .	17
1.6.2 Segments et intervalles qui caractérisent un ECG normal. . . . .	18
1.7 Conclusion . . . . .	20
<b>2 Deep Learning pour la classification des arythmies cardiaques</b>	<b>21</b>
2.1 Introduction . . . . .	21
2.2 Intelligence artificielle . . . . .	21
2.3 Apprentissage automatique . . . . .	21
2.3.1 Définition . . . . .	21
2.3.2 Approches de l'apprentissage automatique . . . . .	22
2.4 Apprentissage profond . . . . .	23
2.5 Différence entre le deep learning et le machine learning . . . . .	23
2.6 Réseau de neurones convolutif (CNN) . . . . .	23
2.6.1 Les avantages des CNN . . . . .	28
2.6.2 Les domaines d'applications des CNN . . . . .	28
2.6.3 Quelques modèles de CNN . . . . .	28
2.6.4 CNN pour la classification d'images médicales . . . . .	29
2.7 Transfert learning . . . . .	30
2.8 Conclusion . . . . .	30

<b>3</b>	<b>Etat de l'art de classification des arythmies cardiaques</b>	<b>31</b>
3.1	Introduction . . . . .	31
3.2	synthèse bibliographique . . . . .	33
3.3	Conclusion . . . . .	34
<b>4</b>	<b>Nos approches</b>	<b>35</b>
4.1	Introduction . . . . .	35
4.2	Base de données ECG image data . . . . .	35
4.2.1	La base de données MIT-BIH Arrhythmia . . . . .	39
4.2.2	La base de données PTB Diagnostic ECG . . . . .	39
4.3	Prétraitement des données . . . . .	39
4.3.1	Entraînement des modèles . . . . .	42
4.3.2	Métriques d'évaluation . . . . .	45
4.3.3	Evaluation des modèles . . . . .	48
4.3.4	Mobilenet . . . . .	50
4.4	Présentation de notre application . . . . .	51
4.4.1	Présentation des interfaces . . . . .	52
4.5	Conclusion . . . . .	55
<b>5</b>	<b>Conclusion générale</b>	<b>56</b>

# Table des figures

1.1	l'appareil cardio vasculaire	11
1.2	activité électrique du cœur	12
1.3	ECG représentatif de tachycardie auriculaire	12
1.4	ECG représentatif de fibrillation auriculaire	13
1.5	ECG représentatif de fibrillation auriculaire	13
1.6	ECG représentatif de bradycardie	14
1.7	ECG représentatif de extrasystole	14
1.8	Ondes d'ECG	19
1.9	Intervalles d'ECG	20
2.1	Différentes approches de l'apprentissage automatique [1]	22
2.2	Architecture d'un CNN	24
2.3	Illustration d'une couche de convolution	25
2.4	Illustration du stride	25
2.5	Exemple d'application de zero-padding	26
2.6	Fonction d'activation Relu	26
2.7	Action de la couche de pooling	27
2.8	Couche fully-connected	27
2.9	Architecture de Resnet50	29
2.10	Architecture de MobileNet	29
4.1	Processus de construction de notre modèle	35
4.2	Exemple d'une image ECG de battement normal	36
4.3	Exemple d'une image ECG d'extrasystole supraventriculaire	36
4.4	Exemple d'une image ECG de contraction ventriculaire prématurée	37
4.5	Exemple d'une image ECG de fusion entre un battement ventriculaire et normal	37
4.6	Exemple d'une image ECG de battement non classifiable	38
4.7	Exemple d'une image ECG d'infarctus du myocarde	38
4.8	Contenu de la base MIT-BIH Arrhythmia	39
4.9	Contenu de la base PTB Diagnostic ECG	39
4.10	Processus de traitement et de préparation de notre dataset	40
4.11	Redimensionnement des données de test	40
4.12	Redimensionnement des données d'entraînement	41
4.13	Partitionnement des données	41
4.14	Chemins vers les dossiers contenant les images redimensionnées pour l'entraînement et le test	42
4.15	Générateur d'images pour les données d'entraînement avec augmentation des données.	42

4.16	Générateur d'images pour les données de test sans augmentation des données . .	43
4.17	Chargement de Resnet50. . . . .	43
4.18	Entraînement du modèle . . . . .	44
4.19	Précision et perte tout au long de l'exécution du modèle. . . . .	45
4.20	Précision et perte tout au long de l'exécution du modèle mobilenet. . . . .	46
4.21	matrice de confusion . . . . .	46
4.22	Graphe d'accuracy du modèle Resnet50 . . . . .	48
4.23	Graphe de loss du modèle Resnet50 . . . . .	49
4.24	Matrice de confusion du modèle Resnet50 . . . . .	49
4.25	Graphe d'accuracy loss du modèle mobilnet . . . . .	50
4.26	Graphe de loss du modèle mobilenet . . . . .	50
4.27	Matrice de confusion du modèle mobilenet . . . . .	51
4.28	Processus de classification d'ECG à l'aide de notre application . . . . .	52
4.29	Interface principale . . . . .	52
4.30	Interface de d'inscription . . . . .	53
4.31	Interface de connexion . . . . .	53
4.32	Interface de classification . . . . .	54
4.33	Interface d'affichage . . . . .	55

# Introduction générale

Selon l’OMS (Organisation Mondiale de la Santé), les maladies cardiovasculaires sont la cause de mortalité numéro un dans le monde, près de trois quarts de ces décès se produisent dans des pays à revenu faible ou intermédiaire.

La classification des maladies cardiaques dont les arythmies cardiaques est une tâche primordiale dans le domaine de la cardiologie, elle permet l’identification et le diagnostic des irrégularités du rythme cardiaque à partir d’enregistrements d’électrocardiogrammes (ECG). Elle est traditionnellement effectuée par les médecins et les experts en cardiologie à l’aide de méthodes manuelles pour analyser les signaux ECG et classifier les différents types d’arythmie. Toutefois, grâce aux récentes avancées en matière d’apprentissage automatique, notamment les techniques de deep learning, il est désormais possible d’automatiser cette classification en utilisant des images d’ECG.

L’électrocardiogramme (ECG) quant à lui est le tracé de l’activité électrique du cœur, enregistré grâce aux électrodes collées sur la peau du thorax. Le tracé de l’ECG est une suite de variation de signal appelée : ondes P, QRS et T et U, qui correspondent à différentes phases du cycle cardiaque, Les arythmies cardiaques se manifestent par des variations anormales de ces motifs (tracé d’onde), indiquant un dysfonctionnement électrique du cœur.

Les méthodes de deep learning, spécifiquement les réseaux de neurones convolutionnels (CNN), ont prouvé leur efficacité dans la classification d’images dans plusieurs domaines, tels que la vision par ordinateur et la détection de motifs. En utilisant ces techniques pour les arythmies cardiaques, il est possible de transformer les signaux ECG en images et de recourir à des modèles de deep learning pour automatiser la classification des arythmies.

## Problématique

Notre problématique se concentre sur la nécessité de développer une solution automatisée pour l’analyse et l’interprétation des ECG, et la difficulté d’effectuer une classification précise des arythmies cardiaques à partir d’images d’ECG en raison de la variation des signaux électriques, de la présence de bruit et de la similitude entre certaines formes d’arythmies, surtout pour les analyses de longue durée comme l’ECG de Holter et les cas ambulatoire.

Aussi les techniques classiques de classification, se basant sur l’extraction manuelle de caractéristiques, peuvent être limitées dans leur capacité à représenter la complexité des signaux

---

d'ECG.

Par conséquent, il est nécessaire de développer des approches plus avancées, robustes, précises, évolutives, et automatisées pour améliorer la classification des arythmies cardiaques, ainsi d'aider les médecins à faire face au grand nombre de diagnostics auxquels ils sont confrontés chaque jour.

## Objectif

L'objectif de notre travail est de développer une approche précise et conviviale pour l'interprétation des anomalies de rythmes cardiaques en utilisant le deep learning. La classification des arythmies cardiaques est une tâche complexe qui nécessite une expertise médicale approfondie et une analyse minutieuse des signaux ECG. En exploitant les avantages du deep learning, nous cherchons à créer un modèle de réseau neuronal capable d'apprendre les motifs distincts des différentes arythmies cardiaques et de les classer avec précision. Cette approche vise à faciliter le processus d'interprétation des anomalies de rythmes cardiaques, en offrant une méthode automatisée qui peut aider les professionnels de la santé dans leurs diagnostics et leur prise de décision clinique.

## Structure du mémoire

Ce mémoire est organisé comme suit :

Le premier chapitre est consacré aux notions de cardiologie plus particulièrement l'électrocardiogramme . Nous présentons quelques définitions des différents types d'arythmies cardiaques.

Dans le deuxième chapitre, nous présenterons en détails le CNN , quelques modèles de CNN , leur utilisation pour la classification d'images médicales ainsi la notion du transfert learning qui vont être implémenté dans le 4ème chapitre dans le but de résoudre le problème en question.

Le troisième chapitre représente l'état de l'art de classification des arythmies cardiaques.

Le quatrième chapitre est, quant à lui, dédié à la mise en œuvre de notre approche. Commencant par une vue d'ensemble des outils et de l'environnement. Nous passons à la description de l'ensemble de données utilisés et à leurs pré-traitement puis Nous présenterons les deux modèles adoptés. Ensuite, nous les appliquons sur l'ensemble de données préparé. Puis, nous discuterons et comparerons les résultats obtenus. Ensuite nous passerons à la présentation de notre application de classification des arythmies cardiaques "Arrytmia".

Nous concluons ainsi ce mémoire par une conclusion générale.

# Cardiologie

## 1.1 Introduction

La discipline de la cardiologie se spécialise dans l'étude et le traitement des affections cardiaques. Les professionnels de la cardiologie font usage de l'électrocardiogramme (ECG) pour détecter et prendre des décisions cruciales pour leurs patients. L'utilisation de l'automatisation pour l'analyse de l'ECG est devenue un outil précieux dans ce domaine, permettant une détection rapide et exacte des anomalies cardiaques. Toutefois, l'expertise clinique des cardiologues reste indispensable pour interpréter les résultats automatisés et prendre des décisions individualisées. La coopération entre la technologie et l'expertise médicale garantit des soins de qualité en cardiologie.

## 1.2 Physiologie cardiovasculaire

### 1.2.1 Le coeur

Cet organe est considéré comme une double pompe musculaire (droit et gauche), le cœur droit qui reçoit le sang veineux du corps via les veines caves, puis le projette dans les poumons grâce à l'artère pulmonaire. Et le cœur gauche qui reçoit le sang oxygéné qui vient des poumons via les veines pulmonaires et le propulse finalement vers l'entièreté du corps grâce à l'aorte.

L'aorte contient quatre cavités répartie également sur le cœur droit et gauche, la partie droite comporte : l'oreillette droite et le ventricule, séparés par la valvule tricuspide. Et la partie de gauche contient l'oreillette gauche et le ventricule gauche séparés par la valvule mitrale.

Les oreillettes sont les chambres supérieures et les ventricules sont les chambres inférieures.

**L'oreillette droite** : Reçoit le sang pauvre en oxygène qui circule dans tout le corps grâce aux veines cave supérieure et inférieure, quand elle arrive à sa capacité elle se contracte et le sang alors se dirige vers le ventricule droit car manque de place.

**Le ventricule droit** : C'est donc la cavité inférieure droite qui après avoir reçu le sang faible en oxygène le pompe vers les poumons via l'artère pulmonaire ou il sera oxygéné.

**L'oreillette gauche** : Reçoit le sang oxygéné des poumons via les veines pulmonaires et le conduit vers le ventricule gauche.

**Le ventricule gauche** : Quant à lui il reçoit le sang oxygéné et le dirige vers tout le corps

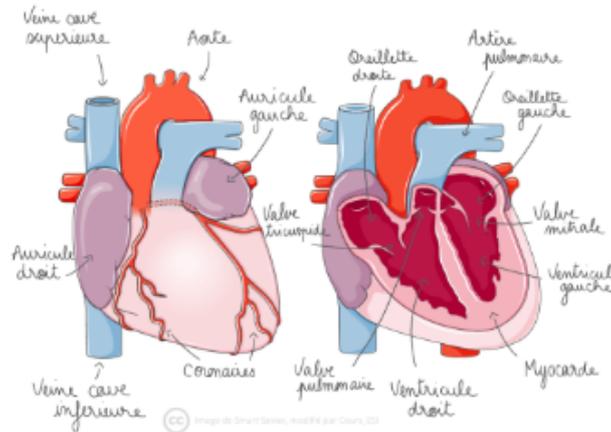


FIGURE 1.1 – l'appareil cardio vasculaire

via l'aorte, afin d'apporter les besoins nécessaires au corps aux tissus et organes. [51]

### 1.2.2 Fonctionnement électrique du cœur

Le courant électrique du cœur naît dans le nœud sinusal (ou nœud sino-auriculaire) situé dans l'oreillette droite. Il traverse les oreillettes, provoquant leur contraction et l'éjection du sang vers les ventricules. Le courant traverse dans le corps grâce aux cellules conductrices, une fois que le courant arrive vers le nœud auriculo-ventriculaire (AV), il (AV) agit comme un retardateur pour permettre aux oreillettes de se contracter complètement avant que les ventricules ne commencent leur contraction. Enfin, le courant est transmis aux ventricules par le faisceau de fibres His et les fibres de Purkinje, entraînant une contraction synchronisée des ventricules et l'éjection du sang du cœur vers le corps. [52]

## 1.3 Principales pathologies cardiovasculaire

Il existe plusieurs pathologies cardiovasculaires qui trouble le fonctionnement correct du cœur et les vaisseaux sanguins, voici quelques-unes des pathologies [46]

- Insuffisance cardiaque.[46]
- Coronaropathie.[46]
- Valvulopathies.[46]

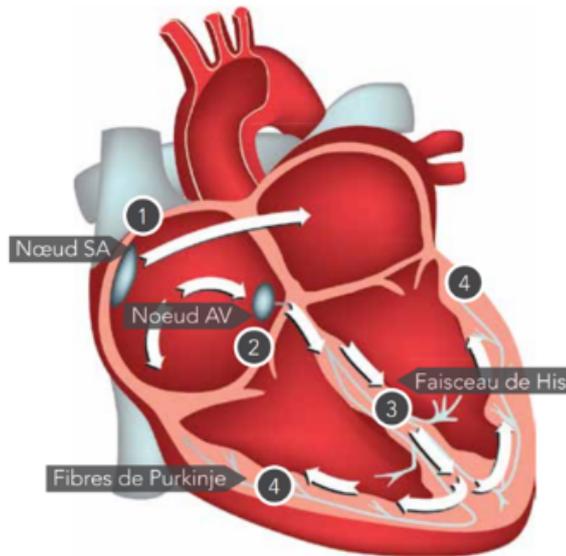


FIGURE 1.2 – activité électrique du cœur

### 1.3.1 Arythmie cardiaque

Fait référence à une anomalie du système électrique du cœur, qui entraîne des battements irréguliers ou anormaux. Dans un cœur sain, des impulsions électriques sont créées selon un modèle précis et se propagent à travers le cœur, synchronisant ainsi ses contractions. Cependant, dans les cas de dysrythmie, ce dispositif électrique ne fonctionne pas correctement, ce qui peut causer des battements trop rapides, trop lents ou irréguliers.[46, 16]

Les arythmies peuvent être classées en différents types voici quelques-uns d'entre eux :

#### 1.3.1.1 Tachycardie

Quand le cœur bat trop vite ou à un rythme supérieur à 100 pulsations à la minute.

Les symptômes peuvent être des étourdissements, des vertiges, fatigue, une faiblesse, une confusion, une perte de conscience, etc.

Cela se produit généralement aux niveaux des oreillettes (tachycardie supraventriculaire) ou dans les ventricules (tachycardie ventriculaire). [16]

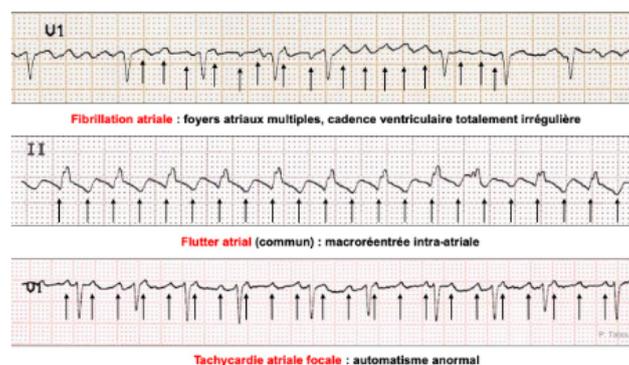


FIGURE 1.3 – ECG représentatif de tachycardie auriculaire

### 1.3.1.2 Fibrillation auriculaire

La fibrillation auriculaire est une arythmie cardiaque courante, généralement observée chez les personnes âgées de plus de 60 ans souffrant d'hypertension artérielle ou de troubles cardiaques. Elle est causée par l'usure du tissu conducteur du cœur. Lorsque l'oreillette est en fibrillation, elle peut se contracter de manière chaotique et rapide, de 350 à 600 fois par minute. Cela peut entraîner des problèmes de circulation sanguine et augmenter le risque de formation de caillots sanguins qui peuvent migrer vers le cerveau et provoquer un accident vasculaire cérébral (AVC). [16]

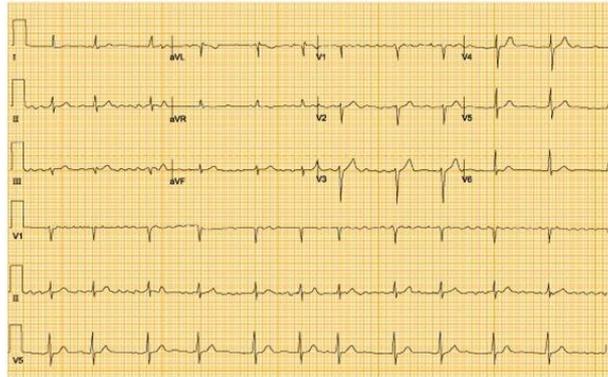


FIGURE 1.4 – ECG représentatif de fibrillation auriculaire

### 1.3.1.3 Fibrillation ventriculaire

C'est une perturbation du rythme cardiaque qui peut être fatale et qui se caractérise par une activité électrique désordonnée et incohérente dans les ventricules. La fibrillation ventriculaire empêche le cœur de pomper le sang efficacement, ce qui entraîne une perte de conscience et un arrêt cardiaque. Un traitement médical urgent, comme la défibrillation, est essentiel pour restaurer un rythme cardiaque normal. [16]

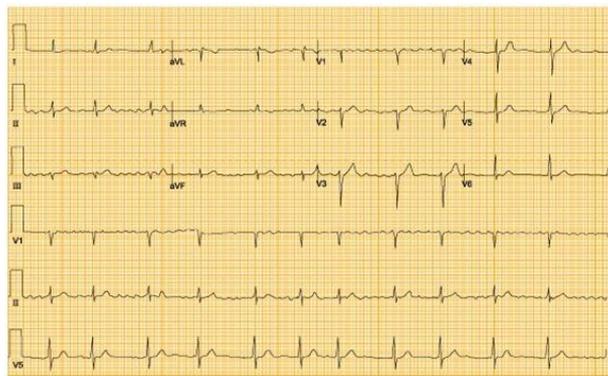


FIGURE 1.5 – ECG représentatif de fibrillation auriculaire

### 1.3.1.4 Bradycardie

C'est la diminution du rythme cardiaque à moins de 50 battements par minutes, ça ne met pas systématiquement la vie en danger même au contraire cela fait référence parfois à une excellente santé chez certains sportifs. Cependant elle peut être due à un problème avec le stimulateur cardiaque ou le nœud sino-auriculaire. La bradycardie persistante justifie la mise en place d'un stimulateur cardiaque (pacemaker).[16]

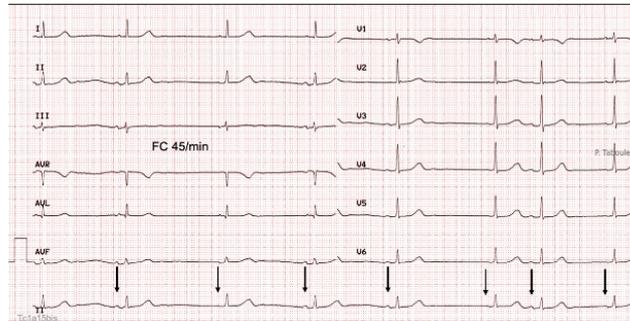


FIGURE 1.6 – ECG représentatif de bradycardie

### 1.3.1.5 Extrasystole

est une forme fréquente d'arythmie cardiaque caractérisée par des contractions cardiaques supplémentaires qui se produisent de manière prématurée, perturbant ainsi le rythme régulier du cœur. Elle est souvent perçue comme une sensation de "saut" ou de "battement sauté" dans la poitrine.[16]

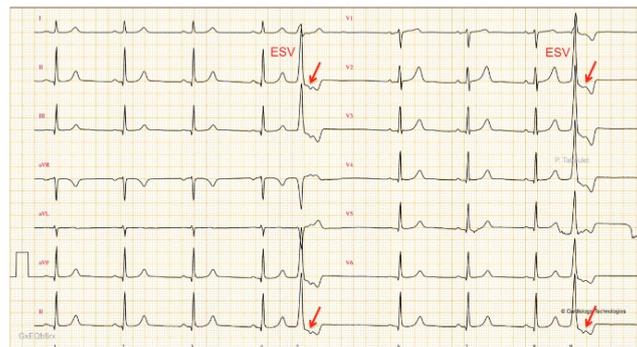


FIGURE 1.7 – ECG représentatif de extrasystole

## 1.3.2 Causes et facteurs de risque des arythmies

Plusieurs causes peuvent être déclencheurs d'arythmies cardiaques comme :

### Maladies cardiaques :

Certaines maladies cardiaques (valvulopathies, les crises cardiaques...) ont la capacité de provoquer des dommages aux tissus du cœur et perturber le rythme cardiaque. [49]

### Hypertension artérielle :

Est l'un des facteurs les plus courants de l'apparition des arythmies cardiaques, car elle peut

générer une surpression dans l'oreillette gauche, ce qui peut enclencher des problèmes de rythmes cardiaques. [49]

**Hyper et hypo-thyroïdie :**

Ces maladies qui concernent la glande thyroïde causent des déséquilibres hormonaux qui peuvent causer des arythmies cardiaques. [49]

**Facteurs génétique (antécédents familiaux) :**

Des altérations génétiques transmises héréditairement sont responsables de certaines anomalies du rythme cardiaque.

Exemple le syndrome du QT long ou le syndrome de Wolff-Parkinson-White sont des arythmies qui peuvent être transmises de génération en génération. [49]

**Stimulants et substances :**

L'utilisation abusive de substances excitantes ou stimulantes telles que la caféine, la nicotine, l'alcool, les drogues récréatives (comme la cocaïne) ou certains médicaments peut provoquer ou intensifier les arythmies chez certaines personnes. [49]

**Autres facteurs :**

D'autres facteurs peuvent également provoquer des arythmies, tels que : Tabagisme, manque d'exercice, problèmes de thyroïde, consommation de drogues, fièvre, électrocution, chirurgie, etc. [49]

### 1.3.3 Mécanisme d'arythmies cardiaques

ce sont les manières dont une arythmie se produit ou est générée au niveau électrique dans le cœur.[41]

**Impulsions électriques anormales :**

Les arythmies cardiaques sont souvent causées par des impulsions électriques anormales qui se produisent en dehors du nœud sinusal. [41]

**Distribution anormale d'impulsion (réentrée) :**

Ceci est le mécanisme le plus courant des troubles du rythme cardiaque. La réentrée se produit lorsque des impulsions électriques circulent en boucle dans le cœur plutôt que de se propager de manière régulière.[41]

**Augmentation de l'automaticité auriculaire :**

C'est la capacité qu'ont des cellules cardiaques à générer des impulsions électriques. [41]

**Déclencheurs (triggers) :**

Certains événements ou facteurs déclencheurs peuvent provoquer des irrégularités du rythme cardiaque chez certaines personnes. Cela peut impliquer des substances stimulantes comme la caféine, le stress émotionnel, la prise d'alcool ou certains médicaments.[41]

### 1.3.4 Symptômes et conséquences des arythmies

[49]

#### Symptômes des arythmies : [49]

Vertiges.

Fatigue.

Palpitations.

Essoufflement.

Étourdissements.

Douleur thoracique.

Perte de conscience.

Tachycardie et bradycardie.

Dans certains les symptômes ne sont pas apparents et l'arythmie est découverte que lors d'un examen médical.

#### Conséquences : [49]

Mort subite.

Insuffisance cardiaque.

Accident vasculaire cérébral (AVC).

Altération de la qualité de vie.

Les conséquences dépendent du type de la gravité de l'arythmie allant d'arythmie bénignes jusqu'à grave.

### 1.3.5 Diagnostic

[49]

Après avoir effectué un questionnaire complet, et pratiquer un test d'effort, si le médecin soupçonne une arythmie il fera appel à ce genre de test :

- électrocardiogramme au repos (ECG).
- électrocardiogramme à l'effort (Épreuve d'effort).
- enregistrement ECG par la méthode Holter ou CardioMémo.
- moniteur diagnostique implantable.
- étude électro physiologique.

## 1.4 Électrocardiographe

Est un dispositif qui mesure et enregistre l'activité cardiaque du cœur

⊗ le graphe de cet enregistrement s'appelle un électrocardiogramme (ECG).

⊗ l'électrocardiogramme est actuellement l'examen le plus courant pour étudier le fonctionnement du cœur ainsi que les anomalies cardiaques.

## 1.5 Electrocardiogramme

Un électrocardiogramme (ECG) est un examen qui analyse le fonctionnement du cœur en évaluant son activité électrique. Lors de chaque pulsation cardiaque (battement), une impulsion électrique (ou « onde ») parcourt le cœur, entraînant la contraction du muscle cardiaque pour permettre l'éjection du sang hors du cœur.

Un ECG mesure et enregistre l'activité électrique qui passe à travers le cœur, puis un médecin fait l'interprétation des résultats.

Un ECG ne permet pas que de détecter les troubles cardiaques, mais aussi les troubles non cardiaques (maladies pulmonaires), ainsi que faire de l'exploration et de la surveillance médicale.[45]

### 1.5.1 Types d'électrocardiogrammes

Il existe plusieurs type d'ECG notamment ECG d'effort et de repos et l'ECG Holter.

L'ECG de repos est le plus récemment effectué et est réalisé au repos, au cours d'une consultation médicale généralement.

l'ECG d'effort est effectué et enregistré pendant l'exercice physique encadré chez le cardiologue. Il est généralement prescrit lorsque le patient décrit des palpitations ou des douleurs à la poitrine, mais que son électrocardiogramme de repos est normal..

L'ECG Holter est un enregistrement continu de l'activité électrique cardiaque sur une période prolongée (généralement 24 heures), permettant de détecter des anomalies qui pourraient ne pas être détectées lors d'un ECG de courte durée, et à quel moment de la journée les troubles du patient sont les plus importants.[45]

## 1.6 Interprétation d'électrocardiogramme

[24]

### 1.6.1 Ondes d'électrocardiogramme (ECG)

les ondes d'ECG représentent les différentes vibrations du graphe vers le haut et vers le bas. qui sont les réactions graphiques des stimulations cardiaques qui se répètent à chaque battement de cœur. le signal ECG est représenté par des lettres d'alphabet « P, Q, R, S et T » [24]

#### 1.6.1.1 Onde P

L'onde P représente la dépolarisation auriculaire, conduisant une impulsion électrique à travers les oreillettes. Elle est généralement positive, apparaissant avant le complexe QRS et facilite le remplissage des ventricules.[24]

#### 1.6.1.2 Complexe QRS

Il désigne la dépolarisation des ventricules, c'est-à-dire leur contraction, . Pour un cas normal, il a une durée inférieure à 0.12 seconde et son amplitude variable est comprise entre 5 et 20 mV , selon les diverses dérivations de l'ECG.[24]

## Onde Q

Habituellement, l'onde Q est une onde négative qui survient avant l'onde R. Elle peut être constatée dans diverses dérivations de l'ECG, La présence et la structure de l'onde Q peuvent donner des informations sur la présence d'une dégénérescence du muscle cardiaque ou d'une blessure au cœur.[24]

## Onde R :

L'onde R est habituellement une onde positive et symbolise la dépolarisation ventriculaire. Sa grandeur et sa forme peuvent différer d'un individu à l'autre, toutefois une onde R typique est communément de configuration positive et clairement définie. Des irrégularités dans l'onde R peuvent signaler des soucis de conduction ou des anomalies électriques.[24]

## Onde S

est habituellement une onde négative qui suit l'onde R. Elle peut aussi donner des renseignements sur la dépolarisation ventriculaire en fonction de sa présence et de sa forme. Dans certaines situations pathologiques, l'onde S peut être amplifiée ou absente.[24]

### 1.6.1.3 Onde T

L'onde T représente la repolarisation des ventricules (c'est-à-dire leur retour à un état électrique de repos). Elle se produit durant la phase de relaxation cardiaque(diastole), et ne reflète d'aucun événement mécanique.. C'est un processus entièrement électrique, souvent positive et de forme dissymétrique.Sa durée oscille entre 0.16s et 0.24S. [24]

### 1.6.1.4 Onde U

Parfois, une minuscule onde U peut être perçue postérieurement à l'onde T. Bien que son origine et sa signification ne soient pas complètement élucidées, elle peut être liée à des situations pathologiques telles que l'hypokaliémie.[24]

## 1.6.2 Segments et intervalles qui caractérisent un ECG normal.

En plus des ondes un ECG comporte un certain nombre d'intervalles et de segments qui sont porteurs d'information très utiles sur la vitesse de conduction de l'impulsion électrique dans les différentes parties du cœur. [24]

### 1.6.2.1 Intervalle RR

représente le temps écoulé entre deux dépolarisations ventriculaires consécutives. Il est utilisé pour déterminer la fréquence cardiaque.[24]

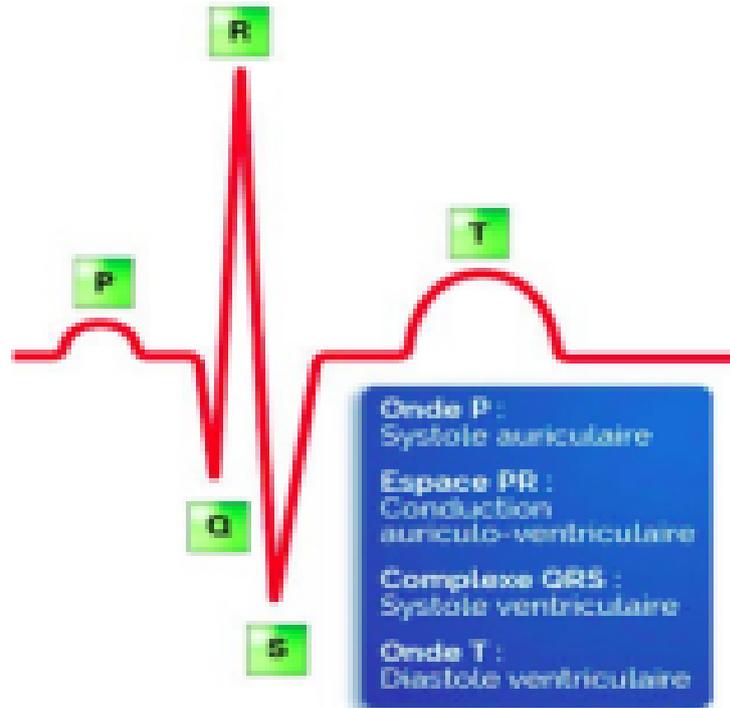


FIGURE 1.8 – Ondes d'ECG

### 1.6.2.2 Intervalle PR

Il est calculé depuis le début de l'onde p jusqu'au début du complexe QRS. Il représente le temps nécessaire pour la conduction à travers le nœud auriculo-ventriculaire (AV) et les faisceaux de conduction. La durée habituelle de l'intervalle PR est couramment comprise entre 120 et 200 ms.[24]

### 1.6.2.3 Intervalle QT

c'est le temps qui s'écoule entre le début du complexe QRS et la fin de l'onde T. Sa durée normale est comprise entre 0,36 et 0,44 seconde.[24]

### 1.6.2.4 Segment ST

La ligne isoélectrique sert à lier la fin de l'onde S et le début de l'onde T. Le segment ST représente une phase de repos électrique pour le myocarde. Tout écart vers le haut ou le bas peut être un signe d'ischémie ou de lésion du myocarde.[24]

### 1.6.2.5 Segment PR

La connexion entre la fin de l'onde P et le début du complexe QRS est appelée la ligne de jonction. La période PR correspond à la durée qui s'écoule entre le début de l'onde P et le début du complexe QRS. Cette période est constante chez les adultes et la durée oscille entre 0,12 et 0,20 seconde.[24]

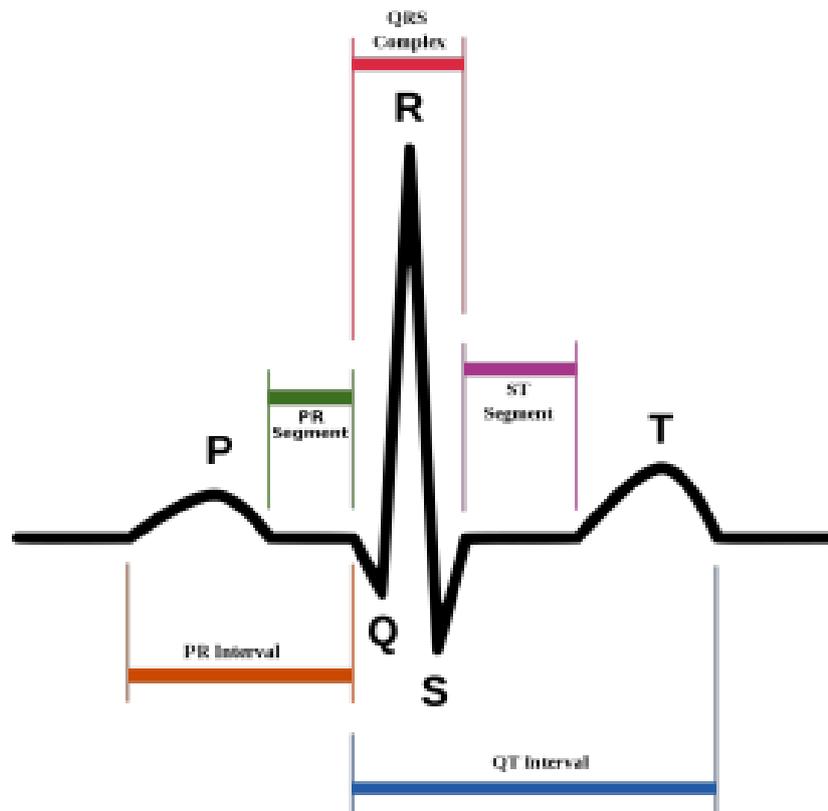


FIGURE 1.9 – Intervalles d'ECG

## 1.7 Conclusion

À la clôture de ce chapitre, on affirme que le cœur est un organe vital et que l'ECG est la plus grande innovation pour repérer les diverses pathologies cardiaques. Dans le prochain chapitre, nous étudierons les idées du deep learning et ses diverses méthodes.

# Deep Learning pour la classification des arythmies cardiaques

## 2.1 Introduction

L'intelligence artificielle (IA) est un domaine de recherche en constante évolution qui utilise des techniques avancées pour permettre aux machines d'apprendre, de raisonner et de prendre des décisions autonomes. Depuis ses débuts, l'IA a connu des avancées spectaculaires et a un impact significatif dans de nombreux domaines, tels que la santé, le transport, l'économie et bien d'autres. Ce chapitre a pour objectif de présenter les différentes méthodes de l'IA qui sont utilisées pour créer des systèmes intelligents capables de résoudre des problèmes complexes. Nous explorerons les concepts fondamentaux de l'IA, les approches traditionnelles ainsi que les méthodes plus récentes basées sur l'apprentissage automatique et les réseaux neuronaux. Nous aborderons également les principales catégories de méthodes de l'IA, notamment l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage par renforcement, et d'autres techniques émergentes qui seront utilisées dans les chapitres suivants.

## 2.2 Intelligence artificielle

L'intelligence artificielle (IA) est une branche de l'informatique qui s'intéresse à la création des machines capables de penser et de prendre des décisions de manière autonome. L'objectif de l'IA est de créer des systèmes informatiques qui peuvent réaliser des tâches qui, autrement nécessiteraient l'intelligence humaine pour être accomplies.[50] Les applications de l'IA sont nombreuses et variées, allant des assistants vocaux aux voitures autonomes en passant par la détection de fraudes dans les transactions financières.

## 2.3 Apprentissage automatique

### 2.3.1 Définition

L'apprentissage automatique, une application de l'intelligence artificielle (IA), permet aux systèmes de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmés. Il se concentre sur le développement de programmes informatiques capables d'accéder aux données et de les utiliser pour en tirer des informations pertinentes. Plutôt que d'être pro-

grammé avec des instructions spécifiques, l'apprentissage automatique utilise des algorithmes pour apprendre de manière itérative à partir des données. Selon SAS, un éditeur de logiciels d'analyse de données, "l'apprentissage automatique est une méthode d'analyse de données qui automatise la création de modèles analytiques". Le processus d'apprentissage commence par l'observation de données, comme des exemples, des expériences directes ou des instructions, afin de rechercher des modèles dans les données et de prendre des décisions éclairées à l'avenir.[22]

### 2.3.2 Approches de l'apprentissage automatique

L'apprentissage automatique se divise en trois sous-domaines (voir Figure 2.1) :

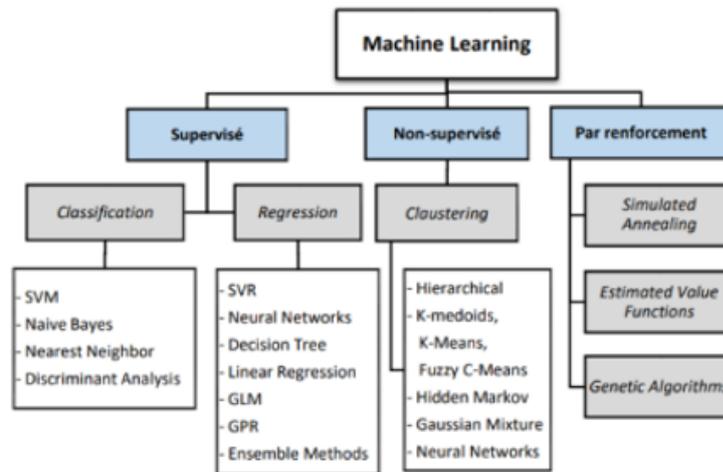


FIGURE 2.1 – Différentes approches de l'apprentissage automatique [1]

- **L'apprentissage supervisé** Le Supervised Learning, ou apprentissage supervisé en français, est la méthode d'apprentissage la plus courante en Machine Learning. Elle consiste à guider la machine dans son apprentissage en lui montrant des exemples de la tâche à accomplir, comme son nom l'indique. Cette technique est utilisée dans de nombreuses applications telles que la reconnaissance vocale, la vision par ordinateur, les régressions et les classifications. Avec l'apprentissage supervisé, la machine peut acquérir la capacité d'accomplir une tâche en étudiant des exemples de cette même tâche. En général, la machine peut apprendre à établir une relation  $F : x \rightarrow y$  entre une entrée  $x$  et une sortie  $y$ , en étudiant un grand nombre d'exemples d'associations [38]  $x \rightarrow y$

- **L'apprentissage non supervisé** : L'apprentissage non supervisé en machine learning ne nécessite pas l'utilisation d'exemples annotés pour entraîner la machine. Au lieu de cela, la machine doit travailler avec des données non étiquetées pour trouver des schémas et des structures significatives dans les données. Les modèles d'apprentissage non supervisé sont utilisés pour le classement des données, le calcul approximatif de la densité de distribution et la réduction des dimensions. Contrairement à l'apprentissage supervisé, il n'y a pas de variable  $Y$  à prédire car la machine ne travaille qu'avec des données non étiquetées. [47]

- **L'apprentissage par renforcement** : L'apprentissage par renforcement, également connu sous le nom de "reinforcement Learning", est une technique d'apprentissage automatique qui

permet à une machine d'accomplir des tâches complexes de manière autonome. Le reinforcement Learning est une technique de machine Learning qui vise à permettre à une entité virtuelle (agent) de prendre des décisions dans un environnement interactif en maximisant des récompenses quantitatives. Pour cela, l'agent réalise des essais et ajuste sa stratégie d'action en fonction des récompenses fournies par l'environnement. [48] L'apprentissage par renforcement est utilisé dans divers domaines tels que : la robotique, la fouille de texte, la finance et la santé. En résumé, l'apprentissage supervisé utilise des données étiquetées pour apprendre à prédire les sorties, l'apprentissage non supervisé utilise des données non étiquetées pour découvrir des structures dans les données, et l'apprentissage par renforcement apprend à partir d'interactions avec un environnement pour prendre des décisions optimales.

## 2.4 Apprentissage profond

Le Deep Learning, ou apprentissage profond en français, est une sous-catégorie du Machine Learning qui utilise des réseaux de neurones artificiels inspirés du cerveau humain pour apprendre à partir de données. [14] Le deep learning utilise des réseaux neuronaux inspirés du cerveau humain. En s'entraînant avec de grandes quantités de données, ces réseaux peuvent analyser et traiter de nouvelles données en temps réel, sans intervention humaine. Les processeurs graphiques (GPU) sont optimisés pour cette tâche grâce à leur capacité de traitement parallèle. [33] Le Deep Learning entraîne des réseaux neuronaux à différents niveaux de complexité pour traiter de grandes quantités de données et accomplir des tâches complexes comme la médecine, la reconnaissance d'image, la traduction, etc. Deep learning : définition, fonctionnement et applications (hubspot.fr) Il est également utilisé pour la compréhension du langage naturel et la résolution de problèmes complexes en intelligence artificielle.

## 2.5 Différence entre le deep learning et le machine learning

Il est important de comprendre les distinctions entre le machine learning et le deep learning, qui est un sous-domaine du premier. Le machine learning est une technologie qui utilise des données organisées et des rétroactions humaines pour améliorer un système, tandis que le deep learning est une section du machine learning qui peut traiter des données non structurées. Le deep learning utilise un réseau neuronal avec plusieurs couches pour permettre au système d'apprendre de manière autonome, contrairement au machine learning qui nécessite une intervention humaine.[53]

## 2.6 Réseau de neurones convolutif (CNN)

Un CNN, ou réseau neuronal convolutionnel, est un type de réseau de neurones artificiels conçu spécifiquement pour le traitement et la reconnaissance d'images en utilisant les données de pixels. Ces systèmes sont particulièrement puissants pour effectuer des tâches génératives et descriptives grâce à l'apprentissage en profondeur (deep learning) de l'IA. Ils peuvent être utilisés pour la reconnaissance d'images et de vidéos, ainsi que pour les systèmes de recommandation et le traitement du langage naturel (NLP) à l'aide de la vision par ordinateur. Le premier CNN a été inventé par Yann LeCun en 1989. [4] La première couche de convolution dans un CNN repère les caractéristiques fondamentales de l'image telles que les bords, les formes et les textures. Les couches suivantes cherchent des caractéristiques plus complexes en se basant sur

les caractéristiques détectées par la couche précédente. Après la dernière couche de convolution, une couche entièrement connectée combine toutes les caractéristiques repérées par les couches de convolution et utilise ces informations pour classifier l'image. [4]

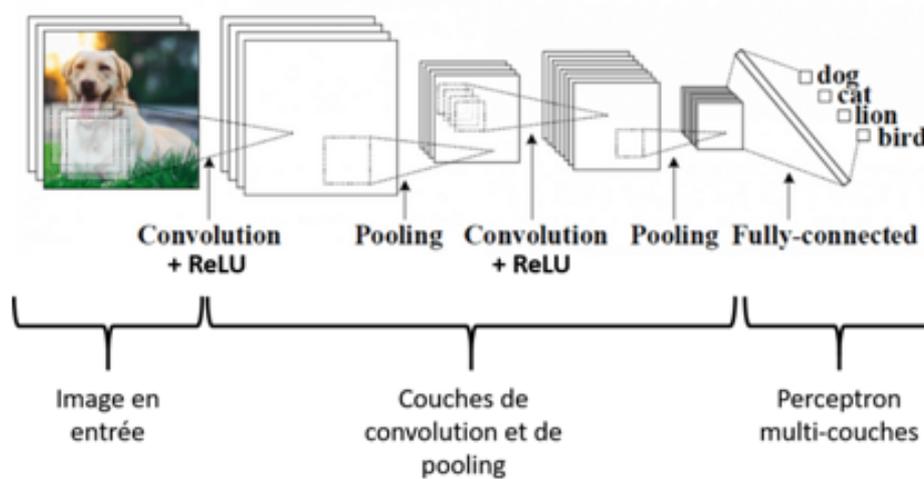


FIGURE 2.2 – Architecture d'un CNN

**1 La couche de convolution :** Dans les réseaux de neurones convolutifs, la couche de convolution est une pièce maîtresse et est généralement la première couche de ces réseaux. Les couches de convolution sont composées de filtres, qui sont des tableaux de valeurs connus sous le nom de cartes de caractéristiques (ou "feature maps" en anglais).[SITE25]

En termes simples, une convolution consiste à appliquer un filtre mathématique à une image en faisant glisser une matrice sur l'image. Pour chaque pixel, la valeur du pixel est multipliée par la valeur correspondante de la matrice, et le résultat de ces multiplications est ensuite sommé. Cette technique permet de détecter des caractéristiques spécifiques dans l'image qui peuvent être utiles pour l'analyse.[18]

Noté que lors de l'opération de convolution, la taille de la carte de caractéristique résultante est dépendante de deux paramètres :le **padding** et le **stride**.

●**Stride :** En ce qui concerne une opération de convolution ou de pooling, la valeur de décalage S représente le nombre de pixels que la fenêtre se déplace après chaque opération.

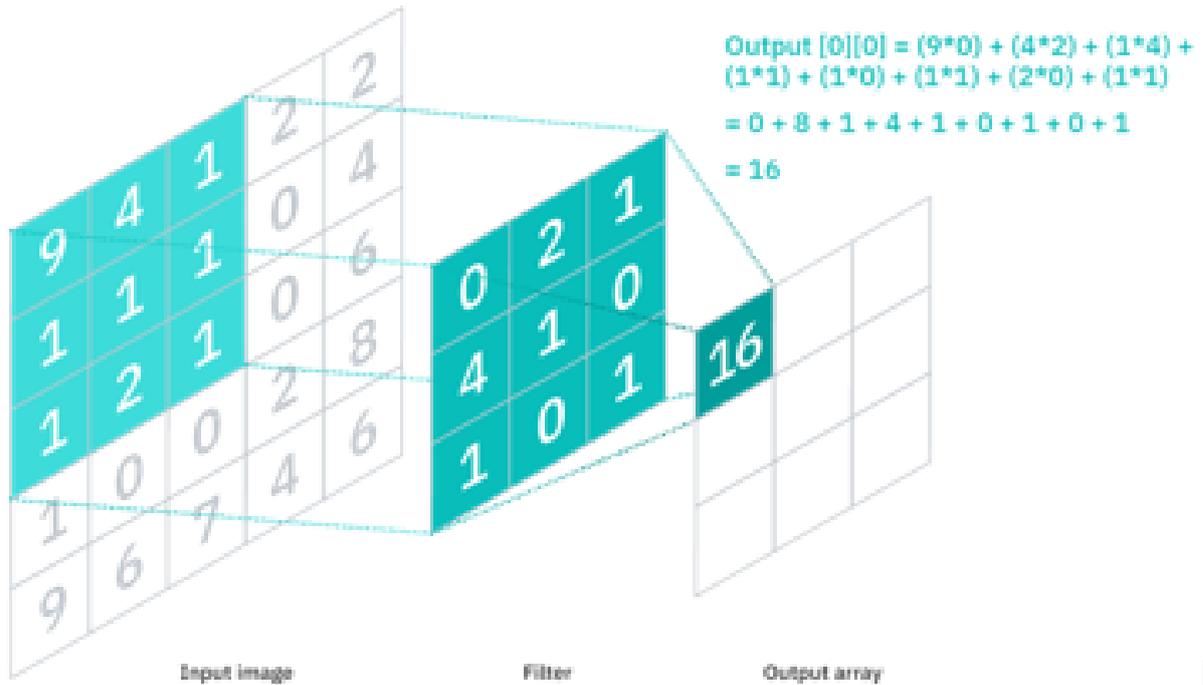


FIGURE 2.3 – Illustration d’une couche de convolution

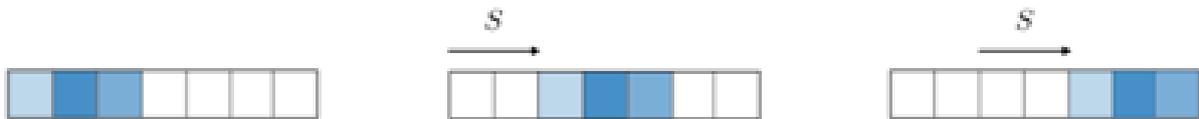


FIGURE 2.4 – Illustration du stride

- **Padding** : Parfois, le filtre ne s’ajuste pas exactement à l’image d’origine. Nous disposons de deux alternatives :
  - Ajouter des zéros (zero-padding) pour rendre l’image conforme.
  - Éliminer la section de l’image qui ne s’adapte pas au filtre. Cette méthode est connue sous le nom de rembourrage valide qui préserve uniquement une partie valide de l’image. [36]

**2. ReLu – Unité linéaire rectifiée** La fonction ReLu est essentielle dans les réseaux de neurones à convolution, car elle doit être appliquée à chaque pixel d’une image après la convolution. Elle remplace toutes les valeurs négatives par un 0, et si elle n’est pas appliquée, la fonction créée sera linéaire, ce qui pose un problème pour le XOR car aucune fonction d’activation n’est appliquée dans la couche de convolution. ReLu est préférée dans les réseaux de neurones à convolution car elle est rapide à calculer grâce à sa forme mathématique simple  $F(y) = \max(0, y)$ , ce qui améliore les performances du réseau par rapport à d’autres fonctions qui nécessitent des opérations plus coûteuses. D’autres fonctions non linéaires, telles que la tangente hyperbolique ou la sigmoïde, peuvent également être employées à la place de la fonction ReLU. Cependant, la plupart des scientifiques des données préfèrent utiliser ReLU, car en termes de performances, ReLU est meilleur que les deux autres. [36]

**3. Couche d’entrée entièrement connectée (Pooling)**

A pour objectif de diminuer la taille de l’image d’origine en effectuant un sous-échantillonnage, ce qui permettra de réduire la charge de traitement, d’alléger l’utilisation de la mémoire ainsi que le nombre de paramètres. [18]

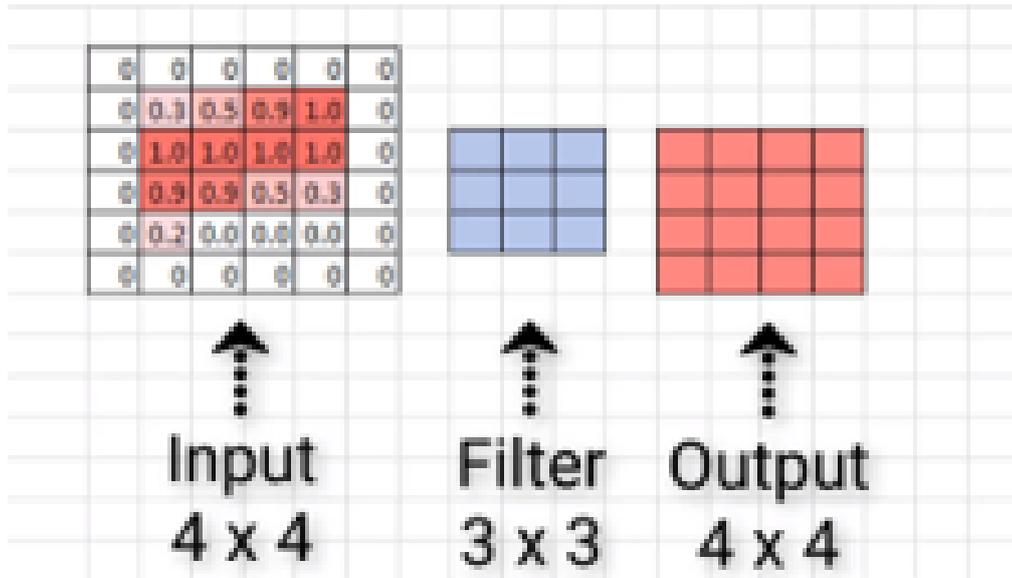


FIGURE 2.5 – Exemple d’application de zero-padding

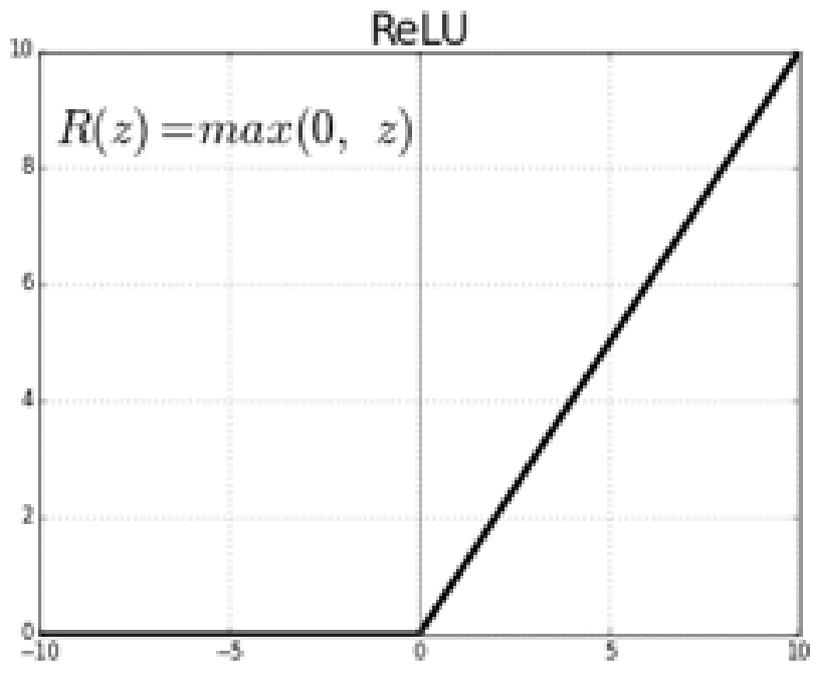


FIGURE 2.6 – Fonction d’activation Relu

Ce genre de couche est souvent situé entre deux couches de convolution : elle reçoit en entrée plusieurs cartes de caractéristiques, et applique à chacune d’entre elles l’opération de pooling. Une couche de pooling, agit comme une couche de réduction. Elle divise l’image en blocs et ne conserve que le maximum de chaque bloc. Cela permet de diminuer la dimension de l’image tout en préservant les caractéristiques les plus essentielles. On obtient en sortie le même nombre de cartes de caractéristiques qu’en entrée, mais celles-ci sont beaucoup plus petites[4].



FIGURE 2.7 – Action de la couche de pooling

**4. Couche entièrement connectée (fully-connected) :** Les CNN sont formés de plusieurs couches de convolution et pooling, suivies d'une couche entièrement connectée qui combine les caractéristiques des couches précédentes pour classer l'image et renvoie un vecteur de taille  $N$ , où  $N$  est le nombre de classes dans notre problème de classification d'images. L'image d'entrée devant appartenir à une classe est indiquée par chaque élément du vecteur. Les couches entièrement connectées ne sont pas une caractéristique des CNN car elles représentent toujours la dernière couche d'un réseau de neurones. Ce type de couche prend un vecteur en entrée et crée un nouveau vecteur en sortie. Pour ce faire, il applique une combinaison linéaire et éventuellement une fonction d'activation aux valeurs reçues en entrées. [4]

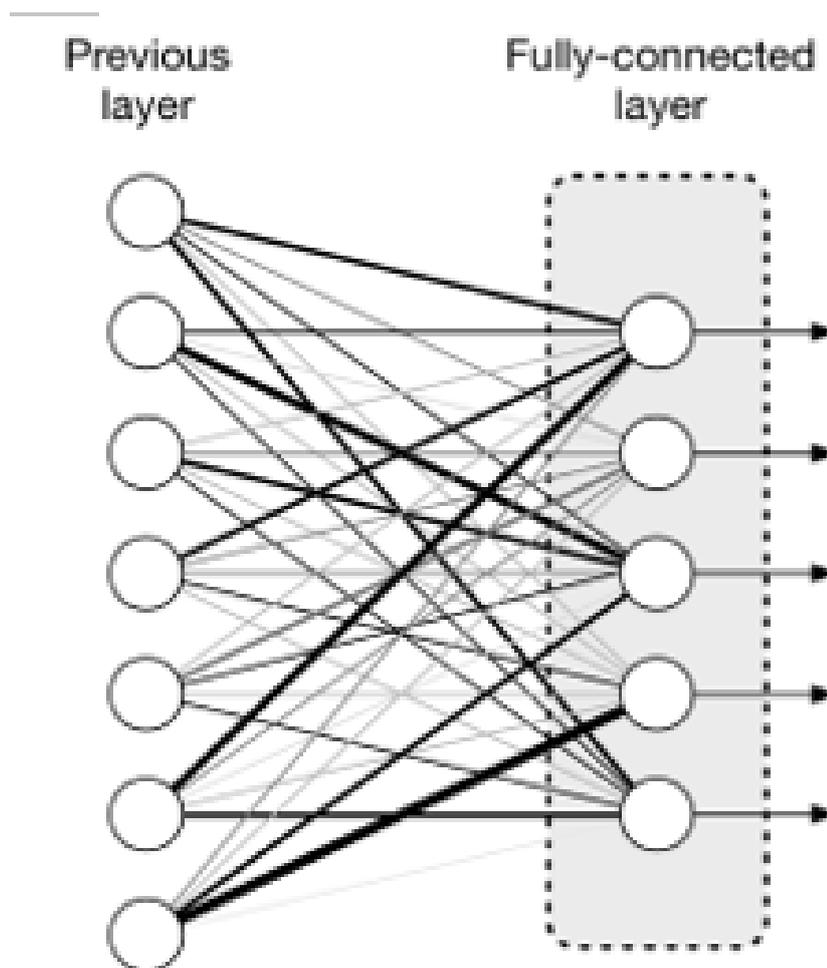


FIGURE 2.8 – Couche fully-connected

### 2.6.1 Les avantages des CNN

- Précision supérieure
- Efficacité du calcul
- Extraction des caractéristiques

### 2.6.2 Les domaines d'applications des CNN

Les CNN sont utilisés pour de nombreux cas d'utilisation dans diverses industries. Les applications réelles des CNN incluent :

- Classification d'image
- Détection d'objet
- Correspondance audiovisuelle
- Reconnaissance vocale
- Reconstruction d'objets [34]

### 2.6.3 Quelques modèles de CNN

En particulier, on présente brièvement six réseaux qui ont prouvé leur efficacité lors d'un concours organisé depuis 2010 sur une base de données d'images appelée ImageNet :

- LeNet proposé par LeCun et al. dans les années 1990.
- AlexNet développé par Alex Krizhevsky en 2012.
- Zfnet développée par Matthew Zeiler et Rob Fergus en 2013.
- GoogLeNet développé par Szegedy et al. de Google en 2014.
- VGGNet construit par Visual Geometry Group (VGG) en 2014.
- ResNet développé par Kaiming He et al. en 2015.

#### 2.6.3.1 Resnet

L'architecture ResNet est largement reconnue comme l'une des plus populaires parmi les réseaux neuronaux convolutifs. Développée par Microsoft Research en 2015, ResNet a rapidement établi de nouveaux records en performance lors de son introduction dans l'article fondateur de He et al.[32]

#### 2.6.3.2 Resnet50

ResNet-50 est un exemple de CNN composé de 50 couches, comprenant 48 couches convolutives, une couche MaxPool et une couche de pooling moyenne. Les réseaux résiduels sont une approche particulière des réseaux neuronaux artificiels (ANN) qui empilent des blocs résiduels pour former le réseau.[32]

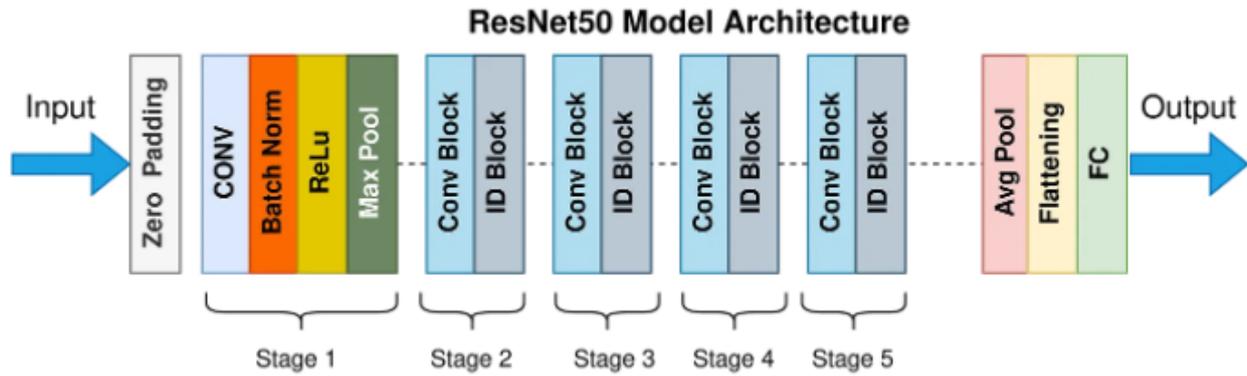


FIGURE 2.9 – Architecture de Resnet50

### 2.6.3.3 MobileNet

MobileNet est un modèle de vision par ordinateur. Il utilise des convolutions en profondeur pour réduire le nombre de paramètres, ce qui le rend plus léger et plus efficace en termes de mémoire et de puissance de calcul. Ainsi, MobileNet permet d'apporter des fonctionnalités avancées de vision par ordinateur sur les appareils mobiles tout en préservant leurs performances.

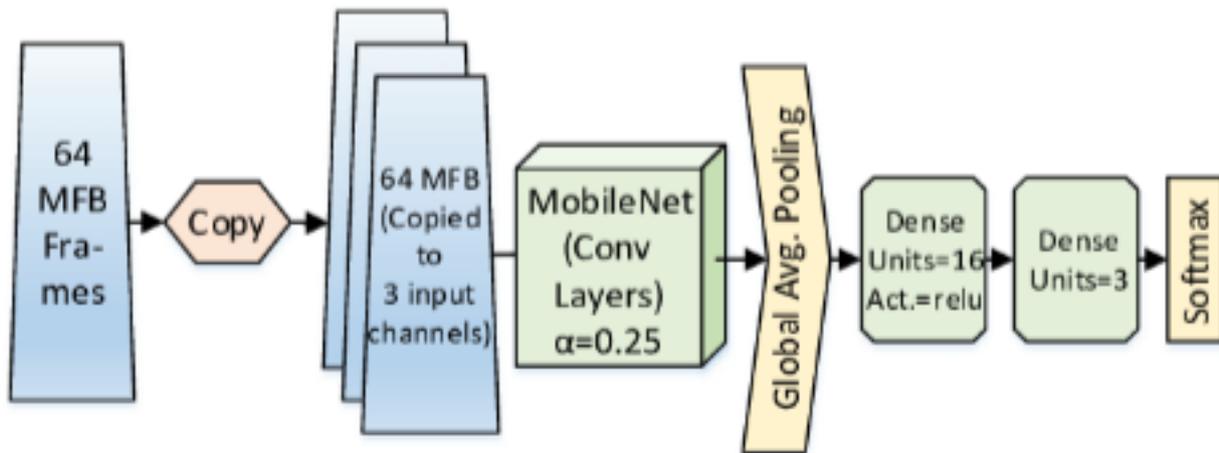


FIGURE 2.10 – Architecture de MobileNet

### 2.6.4 CNN pour la classification d'images médicales

Les réseaux neuronaux convolutifs (CNN) sont largement utilisés pour la classification des images médicales. Cette tâche est essentielle dans le domaine du traitement clinique et de l'enseignement, et les CNN se sont avérés efficaces dans ce domaine. Les CNN sont d'excellents extracteurs de caractéristiques, ce qui permet d'éviter une ingénierie de caractéristiques complexe et coûteuse lors de la classification des images médicales .

Plusieurs architectures CNN sont couramment utilisées pour la classification des images médicales, telles que ResNet, Xception, Inception-ResNet, Squeeze-and-ExcitationNet et Inception. Le choix de l'architecture dépend des besoins spécifiques de la tâche de classification. Par exemple, les architectures Squeeze-and-ExcitationNet et Inception sont adaptées lorsque l'information spatiale est hautement discriminante, tandis qu'un ensemble de modèles avec une architecture

robuste comme ResNet est utile pour classifier plusieurs structures anatomiques.

L'apprentissage par transfert avec les CNN est également une approche populaire pour la classification des images médicales. En utilisant des modèles CNN pré-entraînés, on peut améliorer les performances de la classification en exploitant les paramètres déjà appris .

En résumé, les CNN sont largement utilisés pour la classification des images médicales, et plusieurs architectures et approches peuvent être utilisées en fonction des besoins spécifiques de la tâche.

## 2.7 Transfert learning

L'apprentissage par transfert est une approche populaire en apprentissage automatique, où un modèle pré-entraîné sur une tâche similaire est utilisé comme point de départ pour résoudre un problème connexe . Dans le domaine des réseaux de neurones convolutifs (CNN), cette technique consiste à exploiter un modèle CNN pré-formé pour extraire des caractéristiques pertinentes à partir d'un nouvel ensemble de données . En gelant les premières couches du modèle pré-entraîné et en ne formant que les couches finales sur les nouvelles données, on peut adapter le modèle sans avoir à repartir de zéro. Cette approche a trouvé une large application dans la classification des images médicales, où les paramètres du réseau pré-formé sont exploités pour améliorer les performances de classification. En somme, l'apprentissage par transfert avec les CNN offre une solution efficace pour tirer parti des connaissances préalables et obtenir de bons résultats dans la classification d'images médicales.

## 2.8 Conclusion

En conclusion, dans ce chapitre nous avons parlé sur l'utilisation du Deep Learning, en particulier des réseaux neuronaux convolutifs (CNN), pour la classification d'images médicales, comme les électrocardiogrammes (ECG), a révolutionné le domaine de la médecine. Les CNN permettent une classification précise des arythmies cardiaques, facilitant ainsi le diagnostic précoce. L'apprentissage par transfert avec des modèles pré-entraînés améliore les performances et la capacité d'adaptation des CNN. Ces avancées ouvrent la voie à une médecine plus précise, efficiente et basée sur les données, offrant de nouvelles possibilités pour le diagnostic et la classification des maladies

# Etat de l'art de classification des arythmies cardiaques

## 3.1 Introduction

L'électrocardiographie (ECG) est une technique couramment utilisée pour évaluer l'activité électrique du cœur et détecter d'éventuelles anomalies cardiaques. Dans le domaine de la détection d'anomalie d'un ECG, de nombreuses méthodes ont été proposées, chacune visant à améliorer la précision et l'efficacité du diagnostic. Cette section examine l'état de l'art en matière de détection d'anomalie de l'ECG, en mettant l'accent sur les différentes approches et techniques qui ont été développées. Parmi ces méthodes figurent :

**Article 1 : "Diagnosis of arrhythmia based on ECG analysis using CNN"** Les auteurs Muayed S et al ,ont publié un article en 2020 intitulé "Diagnosis of arrhythmia based on ECG analysis using CNN". Ils ont utilisé des réseaux de neurones convolutifs (CNN) pour diagnostiquer les arythmies cardiaques à partir de l'analyse des signaux ECG. La méthode repose sur un réseau de neurones 2D-CNN entraîné avec des images d'ECG converties à partir de données de la base de données MIH-BIH. Les résultats des tests ont montré une précision de classification de 96,67% et une erreur de 0,004%. Cette méthode offre une réduction du temps de traitement et une flexibilité d'utilisation, ce qui en fait une approche prometteuse pour la détection précoce des arythmies cardiaques.

**Article 2 : "A Hybrid Deep CNN Model for Abnormal Arrhythmia Detection Based on Cardiac ECG Signal"** Les auteurs Amin Ullah et al, ont proposé un modèle hybride basé sur les réseaux de neurones convolutifs (CNN) pour détecter les arythmies cardiaques anormales à partir des signaux ECG. Le modèle utilise à la fois des CNN 1D et 2D pour améliorer la précision de classification. Lors des tests sur la base de données MIT-BIH, les modèles ont atteint des taux de précision de 97,38% et 99,02% respectivement. Cette méthode surpasse les algorithmes existants, offrant une classification précise des signaux d'électrocardiogramme et un potentiel de diagnostic des maladies cardiovasculaires.

**Article 3 : "Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation"** Les auteurs Amin Ullah et al, ont proposé une méthode de classification des arythmies cardiaques utilisant un modèle de réseau de neurones convolutionnels (CNN) bidimensionnel (2-D) avec une représentation d'image spectrale 2D de l'ECG. Le modèle a atteint une précision moyenne de classification de 99,11% sur le jeu de données

MIT-BIH, dépassant ainsi les résultats précédents pour la classification des arythmies. Cette approche prometteuse offre une précision élevée et démontre le succès de l'utilisation du CNN 2-D pour la classification précise des arythmies cardiaques.

**Article 4 : "A Self-Contained STFT CNN for ECG Classification and Arrhythmia Detection at the Edge"** Dans l'article intitulé "A Self-Contained STFT CNN for ECG Classification and Arrhythmia Detection at the Edge" par Mohammed M. Farag en 2022, l'auteur propose une méthode de classification des ECG et de détection des arythmies en utilisant un modèle léger de réseau de neurones convolutionnels (CNN) basé sur la transformation de Fourier à court terme (STFT). Le modèle atteint une précision de classification allant jusqu'à 99,1% et un score F1 de 95% en périphérie, avec une taille maximale de modèle de 90 Ko, un temps moyen d'inférence de 9 ms et une utilisation maximale de la mémoire de 12 Mo. Cette approche offre une solution efficace pour la surveillance des arythmies cardiaques en temps réel sur des appareils en périphérie.

**Article 5 : "Classification des arythmies cardiaques en utilisant les algorithmes SVM, KNN et Naive Bayes"** Dans l'article intitulé "Classification des arythmies cardiaques en utilisant les algorithmes SVM, KNN et Naive Bayes" de Raghavendra M Devadas, l'auteur explore l'utilisation des algorithmes SVM, KNN et Naive Bayes pour la classification des arythmies cardiaques. Les caractéristiques extraites des signaux cardiaques, telles que la fréquence cardiaque, la variabilité de la fréquence cardiaque et les formes d'ondes électriques, sont utilisées comme données d'entrée pour les algorithmes de classification. Les résultats ont montré une amélioration de 9,9% pour le SVM, de 3,3% pour le KNN et de 24,2% pour le Naive Bayes. Le modèle Naive Bayes a été particulièrement surprenant avec une augmentation de 24,2% en termes de précision et de valeur kappa. Cependant, le KNN n'a pas bien performé dans la classification des arythmies par rapport aux autres algorithmes.

### 3.2 synthèse bibliographique

Auteurs	Méthodes	Dataset	Avantages	Inconvénients	Accuracy
Muayed S et al	CNN 2D	MIH-BIH	-Utilisation de l'analyse du signal ECG pour diagnostiquer les arythmies cardiaques. -Utilisation de réseaux de neurones convolutifs (CNN) pour extraire des caractéristiques pertinentes des signaux ECG. -Performance encourageante avec des résultats montrant une précision élevée.	-Taille de l'échantillon non spécifiée, limitant la généralisation des résultats. -Validation externe insuffisante du modèle proposé. -Manque de comparaison avec d'autres méthodes de diagnostic des arythmies cardiaques.	96,67
Amin Ullah et al	CNN 1D , CNN 2D	MIH-BIH	-Approche hybride combinant des réseaux de neurones convolutifs (CNN) et des techniques de transformation de Fourier. -Utilisation de signaux (ECG) pour détecter les arythmies cardiaques. -Performance prometteuse avec de bons résultats en termes de sensibilité et de spécificité.	-Manque de comparaison avec d'autres méthodes de détection des arythmies cardiaques. -Validation externe limitée du modèle proposé. -Détails manquants tels que la taille de l'échantillon de données et les métriques spécifiques de performance.	97,38% , 99,02%
Amin Ullah et al	CNN 2D	CNN 2D MIH-BIH	-Utilisation de représentations d'images spectrales 2D des signaux (ECG) pour la classification des arythmies cardiaques. -Utilisation de techniques d'apprentissage profond, notamment des réseaux de neurones convolutifs (CNN). -Performance encourageante en termes de précision de classification.	-Validation externe limitée du modèle proposé. -Détails manquants tels que la taille de l'échantillon de données et les métriques spécifiques de performance. -Comparaison limitée avec d'autres méthodes de classification des arythmies cardiaques.	99,11%

Auteurs	Méthodes	Dataset	Avantages	Inconvénients	Accuracy
Mohammed M. Farag	CNN basé sur la transformée de Fourier à court terme STFT	MIH-BIH	-Modèle léger de réseau de neurones convolutifs (CNN) utilisant la transformation de Fourier à court terme (STFT) pour la détection des arythmies cardiaques. -Utilisation de techniques d'optimisation pour l'inférence sur des appareils edge.	-Manque de détails sur la taille de l'échantillon de données et les résultats de performance spécifiques. -Pas de comparaison avec d'autres approches de détection d'arythmies sur des appareils edge.	99,1%
Raghavendra M Devadas	KNN, SVM, NAIVE BAYES	UCI Arrhythmia dataset	SVM : - capacité à gérer des ensembles de données complexes, bonne performance dans la détection de certains types d'arythmies spécifiques KNN : - simplicité conceptuelle, performance solide dans la détection de certains types d'arythmies Naive Bayes : - simplicité et rapidité de l'algorithme, performance satisfaisante dans la détection de plusieurs types d'arythmies	SVM :- nécessite un réglage des hyperparamètres, peut être sensible au déséquilibre de classe. KNN : - sensibilité aux valeurs aberrantes, temps de calcul élevé pour de grandes bases de données. Naive Bayes : - supposition d'indépendance des caractéristiques, peut être limité par des distributions de données complexes.	SVM 71.4% KNN 62.6% NB 70.3%

### 3.3 Conclusion

Les résultats montrent qu'en générale le CNN 2D a des performances élevées et atteint un taux de précision de classification pouvant aller jusqu'au 99,11% avec la troisième méthode et 99,1% en utilisant le CNN basé sur la transformée de Fourier à court terme STFT.

## Nos approches

### 4.1 Introduction

Dans ce chapitre consacré à la mise en place d'un réseau neuronal convolutif (CNN), nous aborderons spécifiquement le cas de l'ECG, qui diffère des applications classiques des réseaux convolutifs tels que la reconnaissance d'images. Nous détaillerons la création de nos modèles utilisant l'apprentissage profond pour prédire les arythmies à partir de chaque battement de cœur. En outre, nous présenterons les résultats obtenus en utilisant la base de données publique "ecg\_image\_data, qui regroupe les ensembles de données "MIT-BIH Arrhythmia" et "PTB Diagnostic ECG Database".

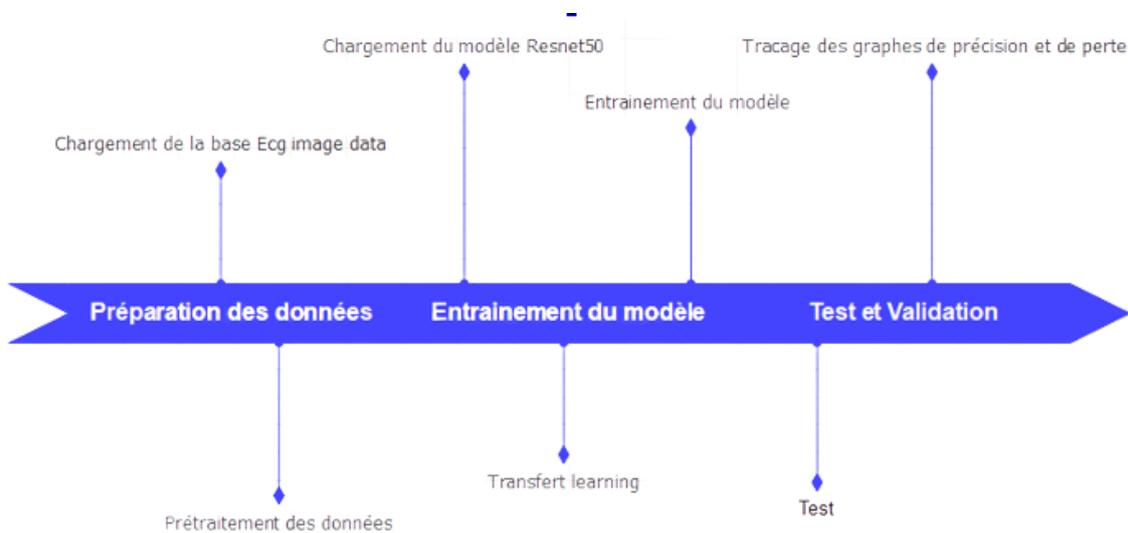


FIGURE 4.1 – Processus de construction de notre modèle

### 4.2 Base de données ECG image data

La base de données d'arythmie ECG est la version image de l'ensemble de données heartbeat [19] Cet ensemble de données comprend deux collections de signaux de battements cardiaques

provenant du MIT-BIH Arrhythmia Dataset et de la base de données PTB Diagnostic ECG. Les échantillons dans les deux collections sont nombreux, ce qui permet de former un réseau neuronal profond.

Cet ensemble de données a été utilisé pour étudier la classification des battements cardiaques en utilisant des architectures de réseaux neuronaux profonds et pour observer les capacités d'apprentissage par transfert. Les signaux correspondent aux formes d'électrocardiogramme (ECG) des battements cardiaques, incluant des cas normaux, des cas avec différentes arythmies et des cas d'infarctus du myocarde. Les signaux ont été prétraités et segmentés, chaque segment représentant un battement cardiaque.

Les classes correspondantes aux signaux sont les suivantes :

N : Battement normal

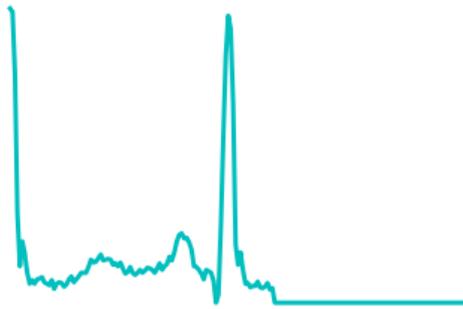


FIGURE 4.2 – Exemple d'une image ECG de battement normal

S : Extrasystole supraventriculaire

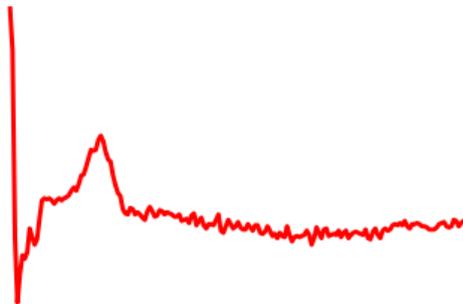


FIGURE 4.3 – Exemple d'une image ECG d'extrasystole supraventriculaire

V : Contraction ventriculaire prématurée

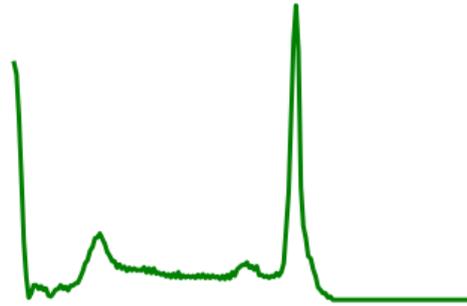


FIGURE 4.4 – Exemple d’une image ECG de contraction ventriculaire prématurée

F : Fusion entre un battement ventriculaire et normal

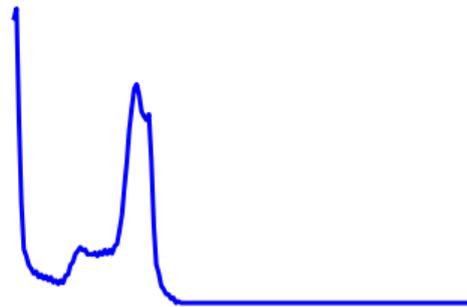


FIGURE 4.5 – Exemple d’une image ECG de fusion entre un battement ventriculaire et normal

Q : Battement non classifiable

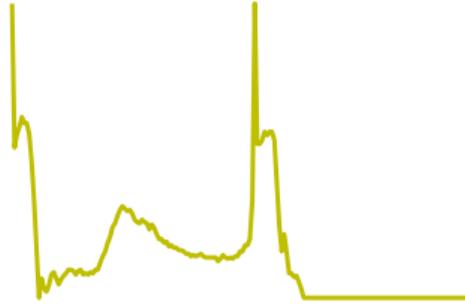


FIGURE 4.6 – Exemple d'une image ECG de battement non classifiable

M : Infarctus du myocarde

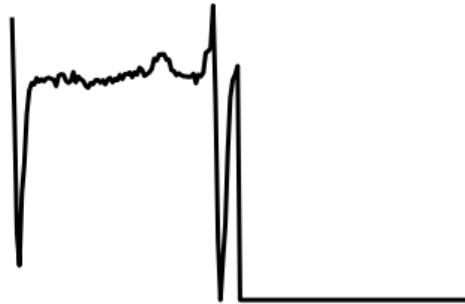


FIGURE 4.7 – Exemple d'une image ECG d'infarctus du myocarde

Le contenu de cette base est composé de ce qui suit :

### 4.2.1 La base de données MIT-BIH Arrhythmia

La base de données MIT-BIH Arrhythmia Database comprend 48 extraits de 30 minutes d'enregistrements ECG ambulatoires à deux canaux, provenant de 47 sujets étudiés entre 1975 et 1979. Les enregistrements ont été sélectionnés aléatoirement parmi un ensemble de 4000 enregistrements ECG ambulatoires d'une heure, comprenant des patients hospitalisés (environ 60%) et des patients externes (environ 40%). Les enregistrements ont été numérisés à 360 échantillons par seconde avec une résolution de 11 bits. [31]

```
Number of Samples: 189446
Number of Categories: 5
Sampling Frequency: 125Hz
Data Source: Physionet's MIT-BIH Arrhythmia Dataset
Classes: ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4]
```

FIGURE 4.8 – Contenu de la base MIT-BIH Arrhythmia

### 4.2.2 La base de données PTB Diagnostic ECG

Les ECG de cette collection ont été obtenus à l'aide d'un enregistreur prototype PTB non commercial, avec les spécifications suivantes : 16 canaux d'entrée (14 pour les ECG, 1 pour la respiration, 1 pour la tension de ligne), une résolution de 16 bits avec 0,5 V/LSB, une bande passante de 0 à 1 kHz et un enregistrement en ligne de la résistance cutanée.

La base de données contient 549 enregistrements de 290 sujets, comprenant l'âge, le sexe, le diagnostic et d'autres données médicales. Les enregistrements incluent 15 signaux mesurés simultanément, dont les 12 dérivations conventionnelles et les 3 dérivations de Frank. Les signaux sont numérisés à 1000 échantillons par seconde, avec une résolution de 16 bits sur une plage de  $\pm 16.384$  mV.

La base de données comprend des sujets atteints de diverses pathologies cardiaques, tels que des infarctus du myocarde, des cardiomyopathies, des blocs de branche, des troubles du rythme cardiaque, etc. Elle inclut également des sujets témoins en bonne santé. [6]

```
Number of Samples: 14552
Number of Categories: 2
Sampling Frequency: 125Hz
Data Source: Physionet's PTB Diagnostic Database
```

FIGURE 4.9 – Contenu de la base PTB Diagnostic ECG

## 4.3 Prétraitement des données

Dans la partie nos approches de notre mémoire, nous avons effectué la préparation des données et l'entraînement du modèle.

La figure suivante montre les étapes que nous avons effectué pour la préparation de nos données.

Tout d'abord, nous avons spécifié la taille cible souhaitée pour les images, dans ce cas (224,

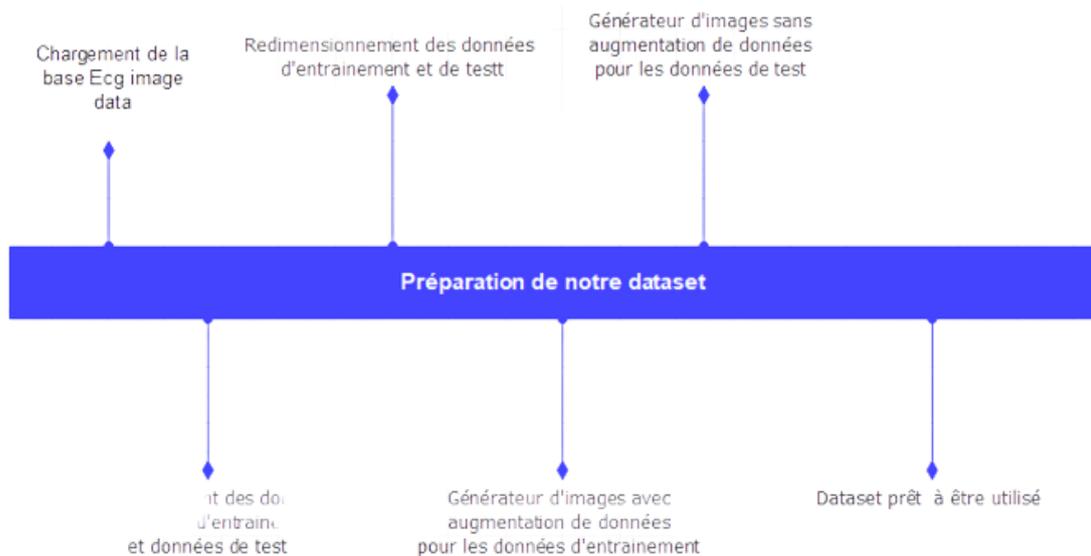


FIGURE 4.10 – Processus de traitement et de préparation de notre dataset

224). Ensuite, nous avons défini les chemins vers les dossiers source et de destination des images. Les images source se trouvent dans le dossier `~/kaggle/input/ecgimage-data/ECG_Image_data/test`, et les images redimensionnées seront sauvegardées dans le dossier `~/kaggle/working/test_resized`. Si le dossier de destination n'existe pas, nous le créerons.

Ensuite, nous parcourons les dossiers source pour chaque classe d'images. Pour chaque image dans le dossier source, nous utilisons OpenCV pour charger l'image. Ensuite, nous redimensionnons l'image à la taille cible spécifiée. Après cela, nous enregistrons l'image redimensionnée dans le dossier de destination correspondant à la classe de l'image.

```
# Préparation des données et entraînement du modèle
import os
import cv2
# spécifiez la taille de destination pour les images
target_size = (224, 224)

# spécifiez les chemins vers les dossiers d'images source et de destination
src = '/kaggle/input/ecg-image-data/ECG_Image_data/test'
src_folder = os.listdir(src)
test_resized = '/kaggle/working/test_resized'
if not os.path.exists(test_resized):
    os.makedirs(test_resized)

for folder in src_folder:
    dst_folder = os.path.join(test_resized, folder)

    # créez un dossier de destination s'il n'existe pas déjà
    if not os.path.exists(dst_folder):
        os.makedirs(dst_folder)

    f = os.path.join(src, folder)
    # bouclez sur toutes les images dans le dossier source
    for filename in os.listdir(f):
        # chargez l'image en utilisant OpenCV
        img = cv2.imread(os.path.join(f, filename))
        # redimensionnez l'image à la taille cible
        img = cv2.resize(img, target_size)
        # enregistrez l'image redimensionnée dans le dossier de destination
        cv2.imwrite(os.path.join(dst_folder, filename), img)
```

FIGURE 4.11 – Redimensionnement des données de test

Nous répétons le même processus pour les images d'entraînement, qui sont situées dans le dossier `~/kaggle/input/ecgimagedata/ECG_Image_data/train`. Les images redimensionnées seront sauvegardées dans le dossier

'/kaggle/working/train\_resized'.

```
# spécifiez les chemins vers les dossiers d'images source et de destination
src = '/kaggle/input/ecg-image-data/ECG_Image_data/train'
src_folder = os.listdir(src)
train_resized = '/kaggle/working/train_resized'
if not os.path.exists(train_resized):
    os.makedirs(train_resized)

for folder in src_folder:

    dst_folder = os.path.join(train_resized, folder)

    # créez un dossier de destination s'il n'existe pas déjà
    if not os.path.exists(dst_folder):
        os.makedirs(dst_folder)

    f = os.path.join(src, folder)
    # bouclez sur toutes les images dans le dossier source
    for filename in os.listdir(f):
        # chargez l'image en utilisant OpenCV
        img = cv2.imread(os.path.join(f, filename))
        # redimensionnez l'image à la taille cible
        img = cv2.resize(img, target_size)
        # enregistrez l'image redimensionnée dans le dossier de destination
        cv2.imwrite(os.path.join(dst_folder, filename), img)
```

FIGURE 4.12 – Redimensionnement des données d'entraînement

Cette préparation des données est importante pour assurer que toutes les images utilisées pour l'entraînement et le test du modèle ont la même taille, ce qui est souvent une exigence pour de nombreux modèles d'apprentissage automatique. En redimensionnant les images, nous nous assurons également de réduire la taille des données, ce qui peut faciliter le processus d'entraînement et améliorer les performances du modèle.

Après l'étape de redimensionnement nous avons obtenu notre dataset qui est réparti selon la figure 4.13 ci-dessous :

```
import os

# Définir les chemins vers les dossiers d'images de train et de test
train_dir = '/kaggle/working/train_resized'
test_dir = '/kaggle/working/test_resized'
def count_data(directory):
    count = 0
    for root, dirs, files in os.walk(directory):
        count += len(files)
    return count

train_data_count = count_data(train_dir)
test_data_count = count_data(test_dir)

print("Nombre de données dans train_dir:", train_data_count)
print("Nombre de données dans test_dir:", test_data_count)
```

Nombre de données dans train\_dir: 99199  
Nombre de données dans test\_dir: 24799

FIGURE 4.13 – Partitionnement des données

### 4.3.1 Entraînement des modèles

Tout d'abord, nous avons défini les chemins vers les dossiers contenant les images redimensionnées pour l'entraînement et le test. Les images d'entraînement sont situées dans le dossier '/kaggle/working/train\_resized', tandis que les images de test sont dans le dossier '/kaggle/working/test\_resized'.

```
# Définir les chemins vers les dossiers d'images de train et de test
train_dir = '/kaggle/working/train_resized'
test_dir = '/kaggle/working/test_resized'
```

FIGURE 4.14 – Chemins vers les dossiers contenant les images redimensionnées pour l'entraînement et le test

Ensuite, nous avons créé un générateur d'images pour les données d'entraînement avec augmentation des données.

```
# Créer un générateur d'images avec augmentation des données pour les données d'entraînement
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True
)
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)
```

FIGURE 4.15 – Générateur d'images pour les données d'entraînement avec augmentation des données.

**L'augmentation des données :** est une technique qui permet de générer de nouvelles images à partir des images existantes en appliquant des transformations telles que des rotations, des translations, des zooms et des retournements horizontaux. Cela aide à augmenter la diversité des données d'entraînement et à améliorer la capacité du modèle à généraliser. Nous avons utilisé la classe `ImageDataGenerator` de TensorFlow pour cela.

**`ImageDataGenerator` :** d'augmenter la taille de l'ensemble de données d'entraînement en créant des versions modifiées des images existantes. Cela améliore les performances du modèle en lui fournissant une plus grande variété d'exemples d'apprentissage.

Nous avons également créé un générateur d'images pour les données de test sans augmentation des données. Les images sont simplement redimensionnées à la taille cible.

Ensuite, nous avons chargé le modèle ResNet50 pré-entraîné, qui est un réseau neuronal convolutif largement utilisé pour la vision par ordinateur. Nous avons gelé les poids du modèle pré-entraîné afin de conserver les connaissances apprises sur les grandes quantités de données

```
# Créer un générateur d'images sans augmentation des données pour les données de test
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)
```

FIGURE 4.16 – Générateur d’images pour les données de test sans augmentation des données d’ImageNet.

Après avoir chargé le modèle Resnet50 nous avons effectué un transfert learning en ajou-

```
# Charger le modèle ResNet50 pré-entraîné sans la couche de classification finale
base_model = ResNet50(include_top=False, weights='imagenet', input_shape=(224, 224, 3))

# Congeler les poids du modèle pré-entraîné
base_model.trainable = False
```

FIGURE 4.17 – Chargement de Resnet50.

tant des couches supplémentaires au modèle pour la classification spécifique de notre tâche. Cela comprend une couche de Global Average Pooling pour réduire les dimensions des features, des couches Dropout pour la régularisation et deux couches Dense pour la classification finale. La dernière couche Dense a un nombre de nœuds égal au nombre de classes dans notre ensemble de données.

Nous avons compilé le modèle en spécifiant l’optimiseur, la fonction de perte et les métriques à utiliser pendant l’entraînement.

Ensuite, nous avons défini l’arrêt précoce (EarlyStopping) comme une technique pour éviter le surajustement du modèle. Cela permet de surveiller la métrique d’exactitude (accuracy) sur les données d’entraînement et d’arrêter l’entraînement si la performance ne s’améliore plus après un certain nombre d’époques.

Enfin, nous avons entraîné le modèle en utilisant la méthode `fit()` de TensorFlow. Nous avons spécifié le générateur d’images d’entraînement, le nombre d’étapes par époque, le nombre d’époques, le générateur d’images de test, le nombre d’étapes de validation, les rappels (callbacks) à utiliser, et le niveau de verbosité. Le modèle a été entraîné en utilisant les techniques pour éviter le surajustement, et l’historique de l’entraînement a été enregistré pour une analyse ultérieure.

**GlobalAveragePooling2D** : la couche GlobalAveragePooling2D de Keras réduit les dimensions spatiales d’un tenseur d’entrée en effectuant une opération de mise en commun moyenne globale. Elle calcule la moyenne des valeurs sur toutes les positions spatiales du tenseur pour chaque canal, ce qui permet de résumer l’information spatiale de manière globale. Cela peut être utilisé pour réduire la complexité des données et extraire les caractéristiques globales essentielles pour la tâche d’apprentissage.

**Dense** : est une couche dans Keras qui crée une couche de réseau neuronal entièrement connectée et applique l’opération sur les données d’entrée. Il est utilisé pour ajouter plus de couches à un réseau de neurones et modifier la dimensionnalité de la sortie de la couche précédente.  $Dense_{output} = activation(dot(input, kernel) + bias)$ [27]

**Dropout** : Le Dropout est une technique de régularisation utilisée dans les réseaux de neurones

```

# Ajouter des couches supplémentaires au modèle
model = tf.keras.Sequential([
    base_model,
    GlobalAveragePooling2D(),
    #Dropout(0.1),
    Dense(256, activation='relu'),
    Dropout(0.1),
    Dense(train_generator.num_classes, activation='softmax')
])

# Compiler le modèle
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Définir l'arrêt précoce (EarlyStopping)
early_stopping = EarlyStopping(monitor='accuracy', min_delta=0, patience=3, restore_best_weights=True, mode='max', baseline=0.99)

# Entraîner le modèle avec les techniques pour éviter le surajustement
epochs = 10
history = model.fit(
    train_generator,
    steps_per_epoch=len(train_generator),
    epochs=epochs,
    validation_data=test_generator,
    validation_steps=len(test_generator),
    callbacks=[early_stopping],
    verbose=1
)

```

FIGURE 4.18 – Entraînement du modèle

pour éviter le surapprentissage. Elle consiste à désactiver aléatoirement un pourcentage des unités d'entrée pendant l'entraînement. Cette méthode est implémentée au niveau des couches du réseau et peut être combinée avec d'autres formes de régularisation. [39]

**Relu** : La fonction ReLU est une activation couramment utilisée dans l'apprentissage en profondeur pour capturer des non-linéarités. Elle transforme les valeurs négatives en zéro et conserve les valeurs positives. Cela permet de modéliser les interactions spécifiques de manière efficace. [7]

**Softmax** : La fonction Softmax est une activation utilisée dans les réseaux de neurones pour transformer un vecteur de valeurs réelles en une distribution de probabilité sur les classes de sortie prédites. Elle garantit que les valeurs résultantes sont normalisées, c'est-à-dire que leur somme est égale à 1. [54]

**Adam** : Adam (Adaptive Moment Estimation) est un algorithme d'optimisation largement utilisé en apprentissage profond et en apprentissage automatique. Il ajuste les taux d'apprentissage de manière adaptative pour chaque paramètre du modèle, ce qui lui permet de converger plus rapidement et de mieux s'adapter aux données d'entraînement. [2]

**EarlyStopping** : EarlyStopping, appelé aussi arrêt anticipé, est une technique d'optimisation qui empêche le surapprentissage sans compromettre la précision. Il surveille la performance du modèle et arrête l'entraînement si aucune amélioration significative n'est observée. [20]

**Sur-ajustement** : ou Overfitting est un problème fréquent dans l'apprentissage automatique et en profondeur où un modèle présente une excellente performance sur les données d'entraînement mais ne généralise pas bien sur de nouvelles données ou les données de test. [15]

Les paramètres et les hyperparamètres sont des concepts importants dans l'apprentissage automatique et l'apprentissage en profondeur.

**Les paramètres** : les paramètres sont internes au modèle et sont appris au cours du processus de formation, ils sont utilisés pour faire des prédictions sur de nouvelles données.

**Les hyperparamètres** : les hyperparamètres sont externes au modèle et sont définis avant le processus de formation, ils sont utilisés pour contrôler le processus d'apprentissage et déterminer les valeurs des paramètres du modèle qu'un algorithme d'apprentissage finit par apprendre (ex :

batch size, epoch, learning rate. . . ).

La Figure 4.19 ci-dessus représente des résultats montrant les performances du modèle sur dix itérations (epochs). La perte diminue et la précision augmente progressivement, indiquant une amélioration du modèle. Les mesures de validation sont utilisées pour évaluer sa performance sur de nouvelles données. L'objectif est d'obtenir un modèle qui généralise bien sur les données d'entraînement et de test. [42]

```

Epoch 1/10
3100/3100 [=====] - 1479s 474ms/step - loss: 0.1794 - accuracy: 0.9420 - val_loss: 0.0871 - val_accuracy: 0.9722
Epoch 2/10
3100/3100 [=====] - 1442s 465ms/step - loss: 0.0780 - accuracy: 0.9741 - val_loss: 0.1354 - val_accuracy: 0.9492
Epoch 3/10
3100/3100 [=====] - 1468s 473ms/step - loss: 0.0643 - accuracy: 0.9785 - val_loss: 0.1080 - val_accuracy: 0.9612
Epoch 4/10
3100/3100 [=====] - 1448s 467ms/step - loss: 0.0544 - accuracy: 0.9818 - val_loss: 0.2921 - val_accuracy: 0.8889
Epoch 5/10
3100/3100 [=====] - 1470s 474ms/step - loss: 0.0502 - accuracy: 0.9826 - val_loss: 0.0823 - val_accuracy: 0.9695
Epoch 6/10
3100/3100 [=====] - 1448s 467ms/step - loss: 0.0485 - accuracy: 0.9835 - val_loss: 0.0936 - val_accuracy: 0.9667
Epoch 7/10
3100/3100 [=====] - 1482s 478ms/step - loss: 0.0433 - accuracy: 0.9849 - val_loss: 0.3007 - val_accuracy: 0.8863
Epoch 8/10
3100/3100 [=====] - 1489s 480ms/step - loss: 0.0418 - accuracy: 0.9852 - val_loss: 0.1343 - val_accuracy: 0.9499
Epoch 9/10
3100/3100 [=====] - 1461s 471ms/step - loss: 0.0414 - accuracy: 0.9858 - val_loss: 0.1160 - val_accuracy: 0.9602
Epoch 10/10
3100/3100 [=====] - 1467s 473ms/step - loss: 0.0371 - accuracy: 0.9871 - val_loss: 0.1181 - val_accuracy: 0.9584

```

FIGURE 4.19 – Précision et perte tout au long de l'exécution du modèle.

Notre deuxième approche est basée sur le même prétraitement des données des mêmes bases de données et un modèle différent mobilnet qui est une architecture de réseau de neurones convolutifs pré-entraînés sur de grands ensembles de données.

Nous avons chargé ce modèle en spécifiant les poids pré-entraînés à utiliser et en excluant la couche de classification supérieure ”(include\_top=False)”

afin de pouvoir utiliser les caractéristiques apprises par le modèle sur un grand dataset, et de les utiliser comme extraction de caractéristiques pour nos tâches spécifiques.

Nous avons gelé les poids des couches du modèle MobileNet pré-entraîné pour les empêcher d'être mis à jour pendant l'entraînement. Cela permet de conserver les caractéristiques apprises par MobileNet et de se concentrer sur l'entraînement de la nouvelle couche de classification ajoutée.

suite à ça nous avons ajouté une nouvelle couche de classification GlobalAveragePooling2D au modèle pour réduire les dimensions spatiales des caractéristiques, suivi de deux couches entièrement connectées (Dense) avec des activations ReLU et des couches Dropout pour régulariser le modèle et éviter le surapprentissage. Enfin, nous avons ajouté une couche Dense finale avec une activation softmax pour obtenir les probabilités de chaque classe. et enfin créer notre modèle final tout en spécifiant les inputs comme l'entrée du modèle MobileNet pré-entraîné et les outputs comme les prédictions de votre nouvelle couche de classification.

enfin après compilation du modèle avec l'optimiseur ADAM la fonction de perte et les metrics d'évaluation ,nous avons entraîné le modèle en utilisant la méthode fit de tensor flow,puis nous avons spécifié le générateur d'images pour les données d'entraînement, et quelques hyperparamètres et évalué le modèle sur l'ensemble de test en utilisant la méthode.

### 4.3.2 Métriques d'évaluation

Une évaluation est nécessaire pour en déterminer les performances et la qualité du modèle sur des tâches.

```

Epoch 1/10
1550/1550 [=====] - 1225s 784ms/step - loss: 0.1015 - accuracy: 0.9705 - precision: 0.9807 - val_loss: 0.0261 - val_accuracy: 0.9932 - val_pre
cision: 0.9939
Epoch 2/10
1550/1550 [=====] - 1207s 779ms/step - loss: 0.0401 - accuracy: 0.9878 - precision: 0.9895 - val_loss: 0.0325 - val_accuracy: 0.9946 - val_pre
cision: 0.9959
Epoch 3/10
1550/1550 [=====] - 1214s 783ms/step - loss: 0.0313 - accuracy: 0.9984 - precision: 0.9917 - val_loss: 0.0167 - val_accuracy: 0.9971 - val_pre
cision: 0.9973
Epoch 4/10
1550/1550 [=====] - 1224s 790ms/step - loss: 0.0281 - accuracy: 0.9912 - precision: 0.9924 - val_loss: 0.0137 - val_accuracy: 0.9971 - val_pre
cision: 0.9975
Epoch 5/10
1550/1550 [=====] - 1224s 790ms/step - loss: 0.0255 - accuracy: 0.9921 - precision: 0.9933 - val_loss: 0.0160 - val_accuracy: 0.9983 - val_pre
cision: 0.9987
Epoch 6/10
1550/1550 [=====] - 1210s 780ms/step - loss: 0.0231 - accuracy: 0.9930 - precision: 0.9940 - val_loss: 0.0059 - val_accuracy: 0.9987 - val_pre
cision: 0.9987
Epoch 7/10
1550/1550 [=====] - 1201s 775ms/step - loss: 0.0221 - accuracy: 0.9934 - precision: 0.9943 - val_loss: 0.0230 - val_accuracy: 0.9939 - val_pre
cision: 0.9940
Epoch 8/10
1550/1550 [=====] - 1204s 777ms/step - loss: 0.0197 - accuracy: 0.9944 - precision: 0.9951 - val_loss: 0.0225 - val_accuracy: 0.9957 - val_pre
cision: 0.9959
Epoch 9/10
1550/1550 [=====] - 1200s 774ms/step - loss: 0.0193 - accuracy: 0.9944 - precision: 0.9949 - val_loss: 0.0151 - val_accuracy: 0.9967 - val_pre
cision: 0.9970
Epoch 10/10
1550/1550 [=====] - 1201s 775ms/step - loss: 0.0186 - accuracy: 0.9948 - precision: 0.9953 - val_loss: 0.0136 - val_accuracy: 0.9982 - val_pre
cision: 0.9984
Epoch: 1
Training - Loss: 0.10145191848278046 Accuracy: 0.9704734683036804
Validation - Loss: 0.02611345238983631 Accuracy: 0.9932255148887634

```

FIGURE 4.20 – Précision et perte tout au long de l’exécution du modèle mobilenet.

### 4.3.2.1 L’accuracy

Est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus, qui calcule la fréquence à laquelle les prédictions correspondent aux libellés. En d’autres termes, la métriques ‘accuracy’ correspond à la fraction des prédictions correctement identifiées par notre modèle.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Comme les autres métriques, l’accuracy est basée sur la matrice de confusion.

Confusion matrix		Reality	
		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

Matrice de confusion

FIGURE 4.21 – matrice de confusion

L’accuracy permet de caractériser la capacité (performance) du modèle à prédire de manière équilibrée pour les individus positifs et négatifs. Elle évalue le taux de prédiction exactes sur la totalité des individus.

TP (Vrais positifs) : Le modèle a correctement prédit les exemples de la classe positive.

TN (vrais négatifs) :Le modèle a correctement prédit les exemples de la classe négatifs.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

FP (Faux positifs) : Le modèle a prédit à tort des exemples de la classe négative comme étant de la classe positive.

FN (Faux négatifs) : Le modèle a prédit à tort des exemples de la classe positive comme étant de la classe négative.

#### 4.3.2.2 Validation accuracy

Au moment de l'entraînement d'un modèle de classification, il est habituel de fractionner les données en deux parties distinctes : un lot pour l'entraînement et un autre pour la validation, et la "validation accuracy" est la métrique utilisée pour mesurer les performances du modèle sur de nouvelle donnée d'apprentissage communément appelé donnée de validation.

est un outil de mesure qui évalue la fréquence des prédictions incorrectes effectuées par un modèle, en comparant les prédictions du modèle aux véritables valeurs cible (étiquettes) d'entraînement. Elle est employée pendant l'entraînement du modèle pour minimiser la fonction de perte, en ajustant les poids et les biais du modèle pendant l'entraînement.

Les fonctions de perte les plus courantes pour les modèles de classification en apprentissage profond sont l'entropie croisée binaire (binary cross-entropy) et l'entropie croisée catégorielle clairsemée (sparse categorical cross-entropy). [35]

#### 4.3.2.3 categorical cross-entropy

représente une fonction de perte qui est employée pour les problèmes de classification à multiples classes. Celle-ci quantifie l'écart entre les probabilités prédites par le modèle pour chaque classe et les étiquettes de classe véridiques.

La perte d'entropie croisée catégorielle est calculée comme la somme négative de la véritable étiquette  $Y$  multipliée par le logarithme de l'étiquette prédite pour chaque classe  $P$ .

loss =  $-\sum_{i=1..n} y_i \log(p_i)$  pour  $i=1..n$  /  $n$  : nombre de classes.

Il convient de souligner que la perte d'entropie croisée catégorielle est adaptée uniquement aux problèmes de classification multiclasse et ne convient pas aux problèmes de classification binaire. Pour ces derniers, il est possible d'utiliser d'autres fonctions de perte telles que l'entropie croisée binaire. [28]

#### 4.3.2.4 Validation loss

est une métrique qui sert à évaluer les performances d'un modèle d'apprentissage sur l'ensemble de validation.

La "validation loss" est comparable à la perte d'apprentissage et se calcule en additionnant les

erreurs pour chaque exemple figurant dans l'ensemble de validation.[3]

### 4.3.3 Evaluation des modèles

#### 4.3.3.1 ResNet50

Pendant le test, le modèle a pu généraliser correctement à partir des connaissances apprises lors de l'entraînement. Chaque prédiction a été en parfaite correspondance avec la classe réelle de l'image, ce qui témoigne de la capacité du modèle à capturer les caractéristiques discriminantes pertinentes pour la classification.

L'historique montre une progression constante de l'exactitude d'entraînement, atteignant

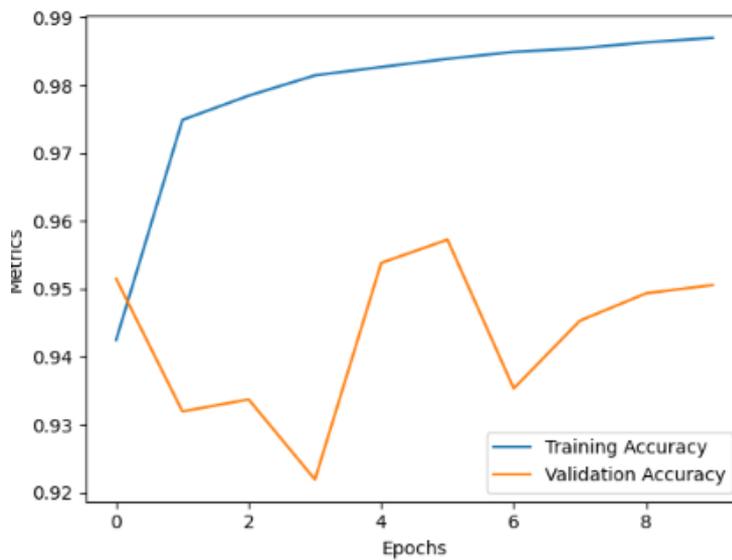


FIGURE 4.22 – Graphe d'accuracy du modèle Resnet50

98,7%, et une légère perturbation de la précision de validation autour de 95,15%. La diminution continue de la perte indique que le modèle s'améliore dans sa capacité à prédire avec précision les étiquettes des nouvelles données.

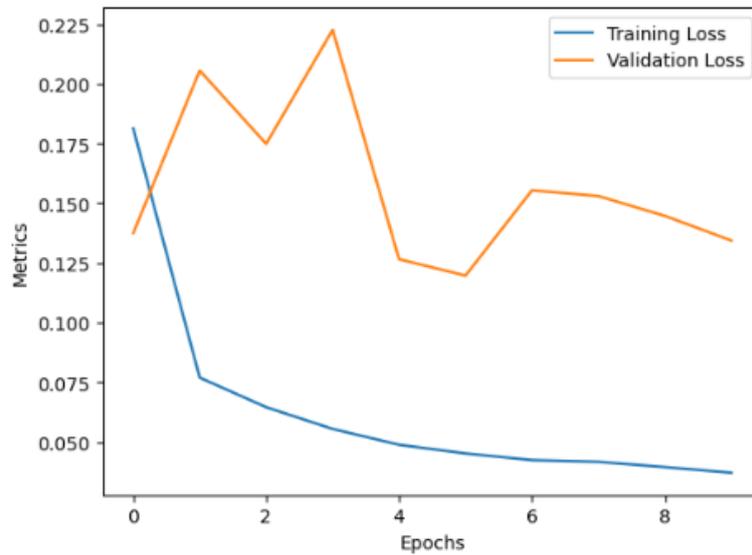


FIGURE 4.23 – Graphe de loss du modèle Resnet50

La perte diminue progressivement de 18,13% à 3,95% lors de la première epoch, tandis que la perte de validation diminue de 13,75% à 13,44%. Cela indique que le modèle s'améliore progressivement dans sa capacité à prédire avec précision les étiquettes de validation en ajustant ses paramètres pour minimiser l'erreur entre les prédictions et les étiquettes réelles.

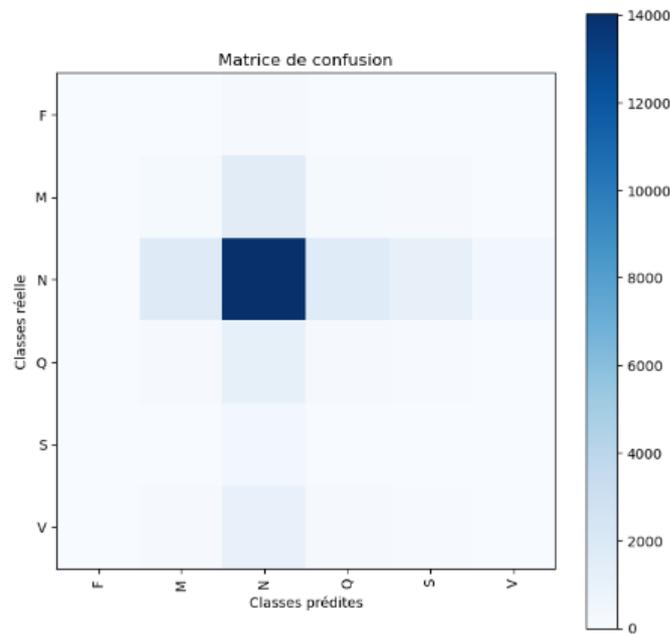


FIGURE 4.24 – Matrice de confusion du modèle Resnet50

Le modèle atteint une précision globale de 95,15%. Cependant, il présente un nombre significatif de faux négatifs (161), limitant sa sensibilité. Il a correctement identifié 24 638 vrais positifs.

### 4.3.4 Mobilenet

pour les résultats de notre deuxième modèle mobilnet nous remarquons que ces résultats montrent qu'il a été entraîné avec succès et qu'il a atteint une bonne performance en termes d'accuracy et de loss sur les échantillons d'entraînement et de validation

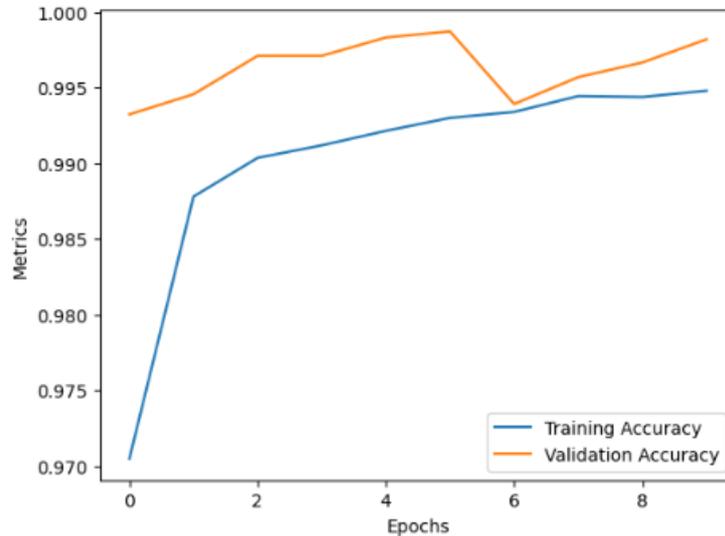


FIGURE 4.25 – Graphe d'accuracy loss du modèle mobilnet

Le modèle a montré une amélioration constante de l'accuracy au cours de l'entraînement, atteignant finalement 99,48%. L'accuracy de validation était déjà élevée dès le départ, à 99,32%, et a augmenté pour atteindre 99,82% à la dernière epoch. En conclusion, le modèle a atteint une très bonne performance en termes d'accuracy, démontrant sa capacité à classifier les données avec précision.

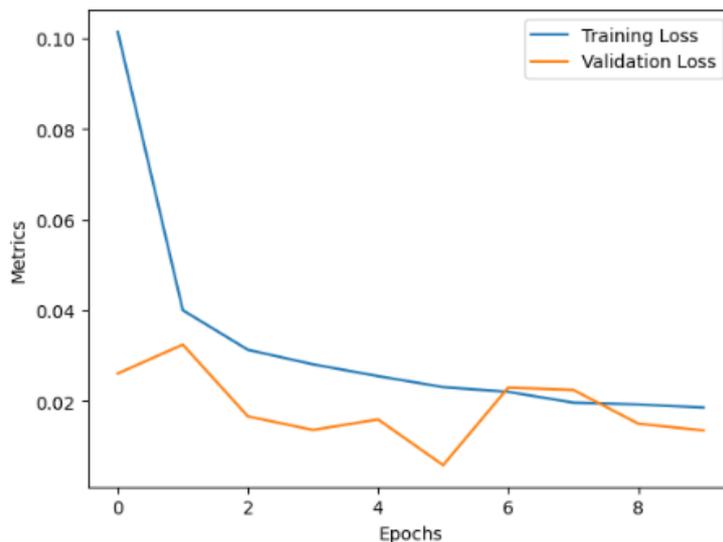


FIGURE 4.26 – Graphe de loss du modèle mobilnet

La loss du modèle a diminué de 0,1015 à 0,0186 au cours des 10 epochs d'entraînement. La loss de validation a également diminué de 0,0261 à 0,0136. Ces résultats indiquent une amélioration progressive de la capacité du modèle à minimiser l'erreur entre les prédictions et les vérités terrain, démontrant ainsi son efficacité d'apprentissage.

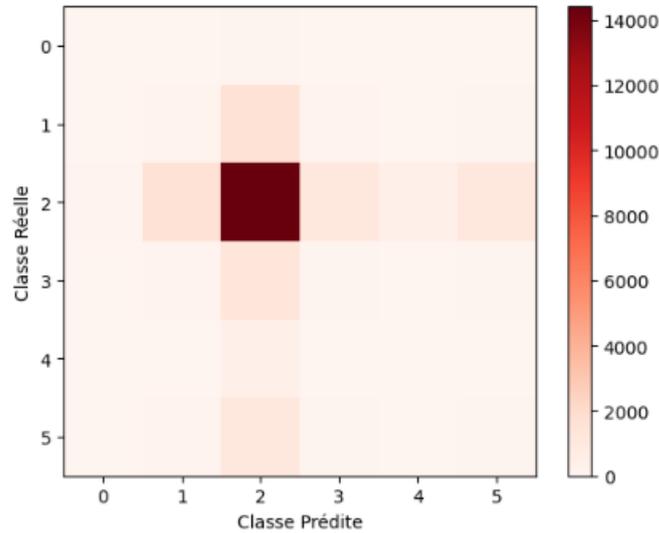


FIGURE 4.27 – Matrice de confusion du modèle mobilenet

Le modèle atteint une précision globale de 99,48%. et il approche un nombre de faux positifs et négatifs de 17 19 respectivement ; ils représentent des chiffres encourageants pour notre entraînement.

TABLE 4.1 – Tableau comparatif de nos modèles

Modèle	Accuracy	Loss
Resnet50	98,57%	0,0409
MobileNet	99,48%	0,0186

## 4.4 Présentation de notre application

Nous présentons dans ce qui suit, notre application desktop "Arrythmia" de classification des arythmies cardiaques par électrocardiogramme (ECG) en utilisant des techniques avancées d'apprentissage profond pour analyser et classifier les images de l'ECG. Cette application offre une assistance précieuse aux professionnels de la santé dans l'analyse et le diagnostic des arythmies cardiaques, en permettant une classification rapide et précise automatique des différents types d'arythmies cardiaques tels que : Battement normal, Battement prématuré supraventriculaire, Contraction prématurée ventriculaire, Fusion du battement ventriculaire et normal, Battement non classifiable, Infarctus du myocarde.

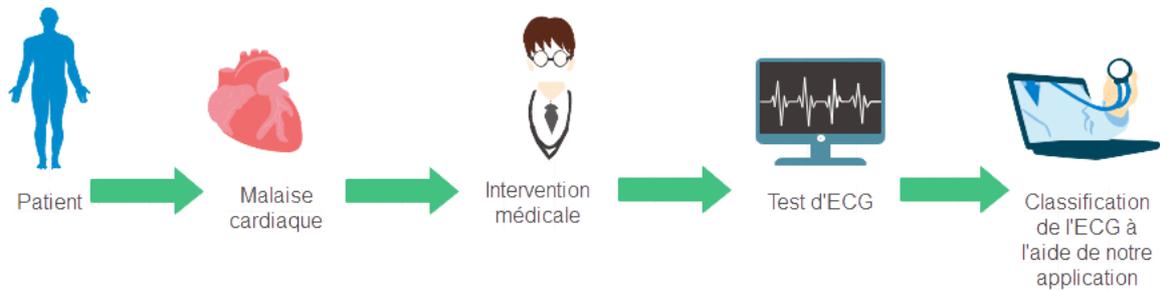


FIGURE 4.28 – Processus de classification d'ECG à l'aide de notre application

#### 4.4.1 Présentation des interfaces

dans ce qui suit, nous exposons les interfaces conviviales de cette application de classification des arythmies cardiaques basée sur des images d'ECG et utilisant le deep learning.

**Interface d'accueil :** cette interface offre un accès facile aux fonctionnalités essentielles.

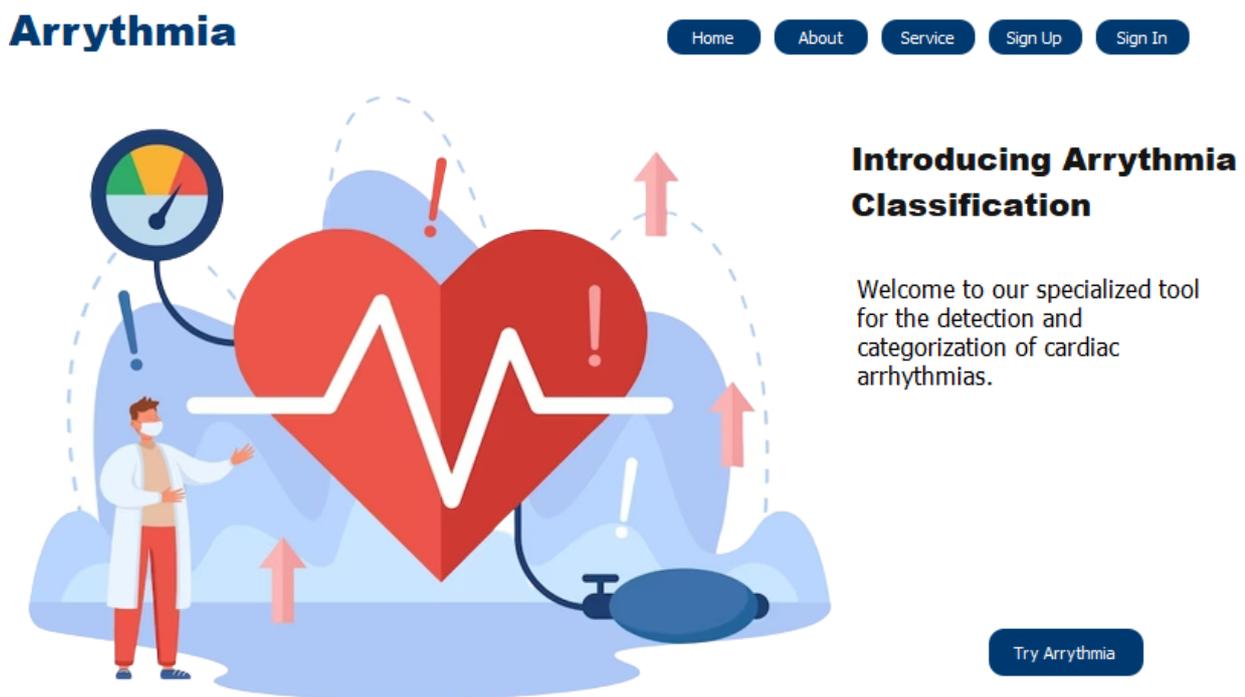


FIGURE 4.29 – Interface principale

**Interface d'inscription** : cette interface permet à un nouveau utilisateur de créer un compte afin de pouvoir accéder aux fonctionnalités de l'application . **Interface de connexion** : cette

## Arrythmia

### Sign Up

**First Name**  
\_\_\_\_\_

**Last Name**  
\_\_\_\_\_

**Email**  
\_\_\_\_\_

**Password**  
\_\_\_\_\_

**Phone Number**  
\_\_\_\_\_

Register

FIGURE 4.30 – Interface de d'inscription

interface permet à un utilisateur de connecter à son compte afin de pouvoir accéder aux fonctionnalités de l'application .

## Arrythmia

### Sign In

**Email**  
\_\_\_\_\_

**Password**  
\_\_\_\_\_

Log In

FIGURE 4.31 – Interface de connexion

**Interface de classification** : cette interface permet au medecin d'effectuer l'operation de classification sur les images d'ECG .

## Arrythmia



FIGURE 4.32 – Interface de classification

**Interface d’affichage** : cette interface permet d’afficher les résultats de la classification ainsi qu’une explication de la classe prédite.

## Arrhythmia

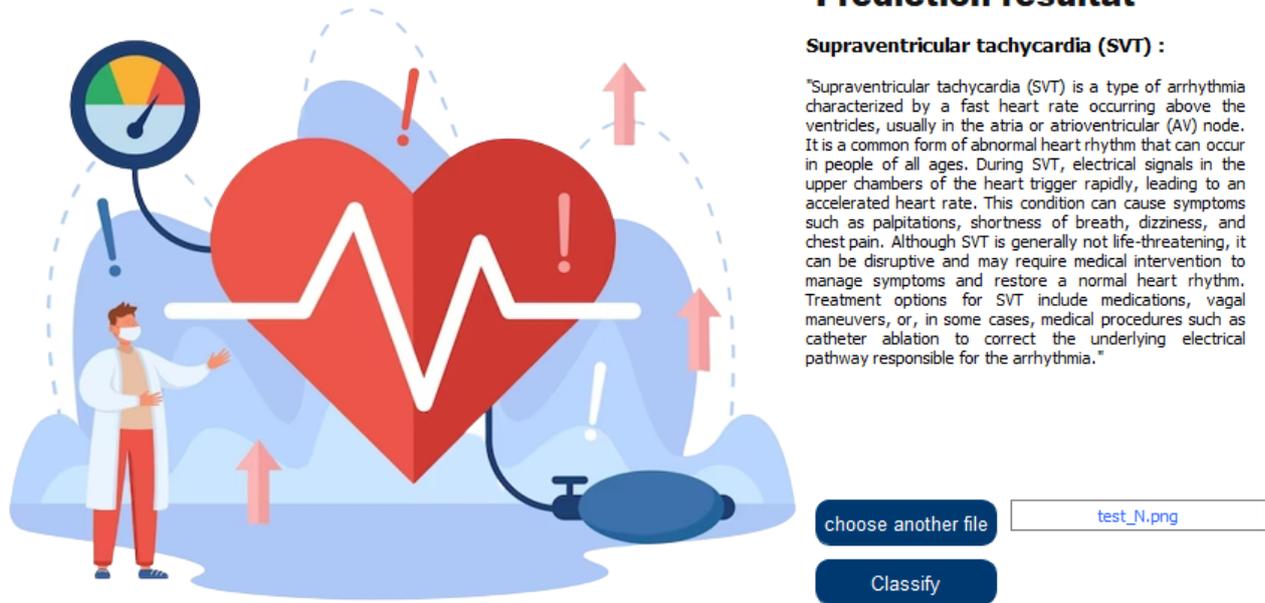


FIGURE 4.33 – Interface d’affichage

## 4.5 Conclusion

Dans ce chapitre, nous avons mis en œuvre et évalué notre modèle de classification d’images d’électrocardiogramme. Nous avons préparé les données en redimensionnant les images et en utilisant des générateurs pour charger les données d’entraînement et de test. Nous avons utilisé le modèle pré-entraîné ResNet50 comme base, en ajoutant des couches supplémentaires adaptées à notre tâche. Le modèle a été compilé avec des paramètres appropriés et entraîné en utilisant des techniques pour éviter le surajustement. Lors des tests et de la validation, notre modèle a prédit avec une précision parfaite les classes réelles des images, démontrant sa capacité à généraliser et à produire des résultats fiables. En conclusion, ce chapitre a abouti à un modèle performant et fiable pour la classification des arrhythmies cardiaques à base d’images d’électrocardiogramme,

## Conclusion générale

En conclusion, ce mémoire a exploré l'application du deep learning, en particulier les réseaux neuronaux convolutifs (CNN), pour la classification des arythmies cardiaques à partir d'images d'électrocardiogramme (ECG) en utilisant le transfert learning. L'utilisation du transfert learning a permis d'exploiter les connaissances acquises par un modèle pré-entraîné sur des données d'ECG afin de fournir une base solide pour la classification des arythmies cardiaques. Les résultats obtenus ont démontré que cette approche était prometteuse, avec des performances encourageantes dans la détection et la classification précise des différentes anomalies de rythme cardiaque. Cette approche présente un potentiel considérable pour aider les professionnels de la santé dans le diagnostic précoce et précis des arythmies cardiaques, ce qui peut contribuer à améliorer les soins aux patients et leur qualité de vie. Cependant, des travaux futurs sont nécessaires pour explorer davantage les possibilités d'amélioration de cette méthode, notamment en utilisant des ensembles de données plus vastes et en exploitant d'autres techniques avancées de deep learning. En somme, cette étude ouvre de nouvelles perspectives passionnantes pour l'utilisation du deep learning et du transfert learning dans le domaine de la classification des arythmies cardiaques basée sur les images d'ECG.

## Annexe

### Outils logiciels

#### Environnement et langage de programmation

- **Kaggle** : Kaggle est une plateforme web leader en Data Science, rassemblant une communauté de plus de 536 000 membres actifs dans 194 pays. Elle offre un environnement Jupyter Notebooks personnalisable avec accès gratuit à des GPU, ainsi qu'une vaste quantité de données et de codes publiés par la communauté. Avec plus de 50 000 jeux de données publics et 400 000 notebooks publics, Kaggle est appréciée par des entreprises telles que Walmart et Facebook, proposant des concours de machine learning, des projets variés et des opportunités d'apprentissage et d'échange avec des experts du domaine.[8]
- **Visual studio code** : Visual Studio Code est un éditeur de code open-source développé par Microsoft qui offre un large éventail de fonctionnalités grâce à des extensions. Il prend en charge de nombreux langages de programmation et offre des fonctionnalités telles que l'auto complétion, la coloration syntaxique, le débogage et les commandes Git.[17]

- **Python** : Python est un langage de programmation interprété, orienté objet et haut niveau, apprécié pour sa simplicité et sa lisibilité. Il offre des structures de données intégrées, favorise la modularité et la réutilisation du code grâce aux modules et packages. Python peut être utilisé pour le développement rapide d'applications et comme langage de script. Son interpréteur et sa bibliothèque standard sont gratuits et disponibles sur différentes plates-formes.[37]

#### Framework

- **Tensorflow** : L'équipe Google Brain a développé Tensorflow pour mener des recherches sur le Machine Learning (ML) et le Deep Learning. Ce framework permettait aux chercheurs et aux développeurs de collaborer sur un modèle d'intelligence artificielle. [9]/tensorflow
- **Keras** : Keras, une API de réseau de neurones écrite en Python, est une bibliothèque Open Source qui fonctionne avec des frameworks tels que Theano et TensorFlow. Créée Francis Chollet, un membre de l'équipe Google Brain, elle se distingue par sa modularité, sa rapidité et sa facilité d'utilisation, offrant une approche simple et intuitive pour développer des modèles de Deep Learning.[9]/keras
- **Qt Designer** : Qt est un framework orienté objet développé en C++ par la société Qt Development Frameworks, une filiale de Nokia. Il propose une variété de composants tels que des widgets d'interface graphique, des outils pour accéder aux données, des fonctionnalités de connexion réseau, de gestion des threads, d'analyse XML, et bien d'autres. Dans certains contextes, Qt peut être considéré comme un framework, notamment lorsqu'il est utilisé pour concevoir des interfaces graphiques ou pour structurer une application en exploitant les mécanismes des signaux et des slots.

#### Bibliothèques utilisées

- **Open Cv** : OpenCV est une bibliothèque open source largement reconnue comme l'outil standard pour la Computer Vision et le traitement d'images. C'est un élément essentiel du Deep Learning. Explorez les caractéristiques et les fonctionnalités de cette puissante bibliothèque.[9]/opencv
- **Os** : Le module os de Python est une bibliothèque polyvalente permettant d'effectuer des opérations système courantes de manière indépendante du système d'exploitation, accéder aux informations et manipuler les fichiers et les répertoires. C'est un outil essentiel pour travailler avec le système d'exploitation dans les applications Python.[44]

- **Matplotlib** : Matplotlib est une bibliothèque permettant de tracer des graphiques, inspirée de Matlab. Elle offre deux approches d'utilisation : l'approche objet, plus avancée mais complexe, et l'utilisation de pyplot, plus simple grâce à des raccourcis. Matplotlib permet de créer des graphiques personnalisés et de dessiner sur une même figure plusieurs graphiques. Cette bibliothèque facilite la création de graphiques au sein d'applications complexes en Python, sans quitter le langage lui-même.
- **Numpy** : NumPy, abréviation de "Numerical Python", est une bibliothèque open source en Python utilisée pour la programmation scientifique, la data science, l'ingénierie et les mathématiques. Elle offre des fonctionnalités puissantes pour les calculs mathématiques et statistiques, y compris la manipulation de matrices et de tableaux multidimensionnels. Son intégration avec d'autres langages comme C/C++ et Fortran est également facile. [9]/numpy
- **Tkinter** : Tkinter (interface Tk) est un module inclus dans la bibliothèque standard de Python, utilisé pour créer des interfaces graphiques comprenant des fenêtres, des widgets tels que des boutons, des zones de texte, des cases à cocher, etc., ainsi que pour gérer les événements liés au clavier et à la souris. Tkinter est disponible sur Windows et la plupart des systèmes Unix, ce qui permet de rendre les interfaces créées avec Tkinter portables. [5]
- **PyQt5** : PyQt est un module GUI permettant de connecter le framework Qt en C++ avec Python. Il offre de nombreuses fonctionnalités, incluant des abstractions de socket réseau, des threads, et la prise en charge de diverses technologies. PyQt permet de créer des composants logiciels réutilisables et est compatible avec différentes plateformes. [10]

# Bibliographie

- [1] Shervine Amidi AFSHINE AMIDI. *Shervine Amidi*. 2021. URL : <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>.
- [2] ATANUSaha. *geeksforgeeks*. 2021. URL : <https://www.geeksforgeeks.org/adam-adaptive-moment-estimation-optimization-ml/>.
- [3] BAELDUNG. *baeldung*. 2023. URL : <https://www.baeldung.com/cs/training-validation-loss-deep-learning>.
- [4] Nada BELAIDI. *blent*. 2022. URL : <https://blent.ai/blog/a/cnn-comment-ca-marche>.
- [5] *blaisepascal*. URL : <https://info.blaisepascal.fr/tkinter/>.
- [6] Ralf-Dieter BOUSSELJOT. *PTB Diagnostic ECG Database*. 2004. URL : <https://www.physionet.org/content/ptbdb/1.0.0/>.
- [7] DANSBECKER. *kaggle*. 2018. URL : <https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning>.
- [8] *datascientest*. 2023. URL : <https://datascientest.com/kaggle-tout-ce-quil-a-savoir-sur-cette-plateforme>.
- [9] *datascientest*. 2023. URL : <https://datascientest.com>.
- [10] *datascientest*. 2023. URL : <https://datascientest.com/pyqt-tout-savoir>.
- [11] *datascientest*. 2023. URL : <https://datascientest.com/les-autoencoders-modeles-dapprentissage-non-supervise>.
- [12] Université Paris Ouest Nanterre La DÉFENSE. “Chapitre 4 : Régression linéaire”. In : (2020).
- [13] Nolwenn DEGESELLE. *deuxiemeavis*. 2021. URL : <https://www.deuxiemeavis.fr/blog/article/15-la-cardiologie-interventionnelle>.
- [14] CÉLINE DELUZARCHE. *futura sciences*. 2021. URL : <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>.
- [15] DEWANGNAUTIYAL. *geeksforgeeks*. 2021. URL : <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>.
- [16] Dora Laty DR GREGORY SCHOUKROUN. *santemagazine*. 2022. URL : <https://www.santemagazine.fr/sante/fiche-maladie/arythmie-cardiaque-177275%20url>.
- [17] ELX54. *framalibre*. 2021. URL : <https://framalibre.org/content/visual-studio-code>.
- [18] Bouchebah FATEH. “CNN”. In : (2023).

- [19] SHAYAN FAZELI. *ECG Heartbeat Categorization Dataset*. 2018. URL : <https://www.kaggle.com/shayanfazeli/heartbeat>.
- [20] Dr. Saptarsi GOSWAMI. *towardsdatascience*. 2020. URL : <https://towardsdatascience.com/early-stopping-a-cool-strategy-to-regularize-neural-networks-bfdeca6d722e>.
- [21] *IBM Cloud*. URL : <https://www.ibm.com/fr-fr/topics/neural-networks>.
- [22] Zakariyaa ISMAILI. *brightcape*. 2019. URL : <https://brightcape.co/quest-ce-que-machine-learning-pourquoi-machine-learning-important/>.
- [23] la production du JDN. *journal du net*. 2022. URL : <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501879-machine-a-vecteurs-de-support-svm-definition-et-cas-d-usage/>.
- [24] ANP-BC JESSICA SHANK COVIELLO DNP APRN. "ECG interpretation". In : 7 (2020).
- [25] JOHAN-CLLAU. "Perceptron. Teori perceptron dan simple". In : *medium* (2020).
- [26] Kassidy KELLEY. *LeMagIT*. 2020. URL : <https://www.lemagit.fr/conseil/Machine-Learning-les-9-types-dalgorithmes-les-plus-pertinents-en-entreprise>.
- [27] *keras*. URL : [https://keras.io/api/layers/core\\_layers/dense/](https://keras.io/api/layers/core_layers/dense/).
- [28] Kiprono Elijah KOECH. *towardsdatascience*. 2020. URL : <https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>.
- [29] LAMBERT.R. *pensée artificielle*. 2019. URL : <https://penseeartificielle.fr/comprendre-lstm-gru-fonctionnement-schema/>.
- [30] LUCAS. "COMMENT L'INTELLIGENCE ARTIFICIELLE VA-T-ELLE RÉVOLUTIONNER LA SANTÉ?" In : *coheris* (2019).
- [31] Roger MARK. *MIT-BIH Arrhythmia Database*. 2005. URL : <https://physionet.org/content/mitdb/1.0.0/>.
- [32] Suvaditya MUKHERJEE. *towardsdatascience*. 2022. URL : <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>.
- [33] *oracle*. URL : <https://www.oracle.com/dz/artificial-intelligence/machine-learning/what-is-deep-learning/>.
- [34] Amrita PATHAK. *geekflare*. 2022. URL : <https://geekflare.com/fr/convolutional-neural-networks/>.
- [35] Sadrach PIERRE. *bultin*. 2022. URL : <https://bultin.com/data-science/loss-functions-deep-learning-python>.
- [36] PRABHU. *medium*. 2018. URL : <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>.
- [37] *python*. 2023. URL : <https://www.python.org/doc/essays/blurb/>.
- [38] Guillaume SAINT-CIRGUE. *machine learnia*. 2029. URL : <https://machinelearnia.com/apprentissage-supervise-4-etapes/>.
- [39] PAVAN SANAGAPATI. *kaggle*. 2019. URL : <https://www.kaggle.com/code/pavansanagapati/what-is-dropout-regularization-find-out>.
- [40] *saturncloud*. URL : <https://saturncloud.io/blog/what-is-higher-validation-accuracy-than-training-accuracy-in-tensorflow-and-keras/#:~:text=in%5C%20machine%5C%20learning,-,Validation%5C%20accuracy%5C%20is%5C%20th%5Ce%5C%20accuracy%5C%20of%5C%20a%5C%20model%5C%20on%5C%20new,seen%5C%20the%5C%20validatio%5Cn%5C%20data%5C%20before..>

- [41] Dr Elodie SURGET. *Fondation recherche medicale*. 2021. URL : <https://www.frm.org/recherches-maladies-cardiovasculaires/troubles-du-rythme-cardiaque/mecanismes-fibrillations>.
- [42] Kelsey TAYLOR. *hitechnectar*. 2021. URL : <https://www.hitechnectar.com/blogs/hyperparameter-vs-parameter/>.
- [43] Data Science TEAM. *Data Science*. 2020. URL : <https://datascience.eu/fr/apprentissage-automatique/comprendre-les-reseaux-lstm/>.
- [44] pythonforge TEAM. *pythonforge*. 2021. URL : <https://pythonforge.com/module-os-systeme-dexploitation/>.
- [45] UNKNOWN. *Cleveland Clinic*. 2022. URL : <https://my.clevelandclinic.org/health/diagnostics/16953-electrocardiogram-ekg>.
- [46] UNKNOWN. *clinalliance*. 2023. URL : <https://www.clinalliance.fr/infos-conseils/conseils-cliniques-ssr/principales-maladies-cardiovasculaires/>.
- [47] UNKNOWN. *datascientest*. 2023. URL : <https://datascientest.com/apprentissage-non-supervise>.
- [48] UNKNOWN. *datascientest*. 2023. URL : <https://datascientest.com/q-learning-le-machine-learning-avec-apprentissage-par-renforcement>.
- [49] UNKNOWN. *institut de cardiologie de l'université de ottawa*. 2021. URL : <https://www.ottawaheart.ca/fr/maladie-du-c593ur/les-arythmies-anomalies-du-rythme-cardiaque>.
- [50] UNKNOWN. *NetApp*. 2022. URL : <https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/>.
- [51] UNKNOWN. *coeuretavc*. 2022. URL : <https://www.coeuretavc.ca/maladies-du-coeur/qu-est-ce-que-les-maladies-du-coeur/comment-fonctionne-le-coeur>.
- [52] UNKNOWN. *fedecardio*. 2023. URL : <https://www.fedecardio.org/je-m-informe/1-activite-electrique-du-coeur/>.
- [53] Hugo WELTER. *hubspot*. 2019. URL : <https://blog.hubspot.fr/marketing/deep-learning#:~:text=L'utilit%C3%A9%20essentielle%20du%20deep,reconnaissance%20faciale%2C%20robotique...>
- [54] Thomas WOOD. *DeepIA*. 202. URL : <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>.

# Résumé

L'électrocardiogramme (ECG) est un test crucial pour évaluer l'activité électrique du cœur en enregistrant les variations des ondes électriques résultant de la polarisation et de la dépolarisation des ventricules et des oreillettes. Il joue un rôle essentiel dans le domaine médical en permettant la détection de divers problèmes cardiaques. Dans ce contexte, notre étude aborde la problématique de la détection des anomalies de rythme cardiaque en utilisant des techniques de deep learning, plus précisément les réseaux neuronaux convolutifs (CNN). Notre approche se base sur l'utilisation du transfert learning, où nous exploitons un modèle pré-entraîné sur des images d'ECG pour classifier les arythmies cardiaque. Les performances de notre algorithme ont été évaluées sur des signaux ECG provenant de patients différents de la base de données " MIT-BIH Arrhythmia" et " PTB Diagnostic ECG ". Cette approche novatrice ouvre de nouvelles perspectives dans le domaine de la classification des arythmies cardiaques en exploitant les avancées du deep learning.

**Mots clés :** CNN, deep learning (DL), Electrocardiogramme (ECG), MIT-BIH, PTB Diagnostic ECG.

## Abstract

The electrocardiogram (ECG) is a crucial test for evaluating the electrical activity of the heart by recording the variations in electrical waves resulting from the polarization and depolarization of the ventricles and atrials. It plays a vital role in the medical field by enabling the detection of various heart problems. In this context, our study addresses the problem of detecting cardiac rhythm anomalies using deep learning techniques, specifically convolutional neural networks (CNN). Our approach is based on transfer learning, where we leverage a pre-trained model on ECG images to classify cardiac arrhythmias. The performance of our algorithm has been evaluated on ECG signals from different patients in the " MIT-BIH Arrhythmia " and " PTB Diagnostic ECG " databases. This innovative approach opens up new perspectives in the field of cardiac arrhythmia classification by harnessing the advancements of deep learning.

**Keywords :** CNN, deep learning (DL), Electrocardiogram (ECG), MIT-BIH, PTB Diagnostic ECG.