

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE AKLI MOAND OULHADJE-BOUIRA



Faculté des Sciences et des Sciences Appliquées  
Département d'informatique

**Mémoire de fin d'étude**

Présenté par:

**MERDOUD Kenza**  
**BOUSBAIN Karim**

En vue de l'obtention du diplôme de **Master 02** en :

Filière: INFORMATIQUE

Option: Génie Système Informatique

**Thème:**

**Détection de maladies par traitement d'image**

**Devant le jury composé de:**

BOUDJELABA HAKIM  
ABBAS Akli  
HAMID Rabeh

UAMOB  
UAMOB  
UAMOB  
UAMOB

Président  
Encadreur  
Examineur  
Examineur

Année Universitaire 2017/2018

## *Remerciements*

*Nous remercions Dieu le tout puissant, qui nous a donné la force et la patience pour l'accomplissement de ce travail.*

*Nous tenons à exprimer notre gratitude à monsieur ABBAS pour son orientation, supervision et sa patience. Le travail sous sa supervision a été une expérience de confiance.*

*Nous tenons tout d'abord à exprimer notre reconnaissance envers les membres du jury pour l'intérêt qu'ils ont porté à ce mémoire en acceptant d'en être examinateurs. Nous remercions les enseignants de la département informatique qui m'enseignée .*

*Enfin, Nous tenons à remercier tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.*

*Merci à tous et à toutes.*

## *Dédicaces*

*Je dédie ce travail aux êtres qui me sont les plus chers :*

*A l'âme de mes grands-parents, mes symboles, qui m'est très chère .*

*A mon très cher père, source de sagesse, mon guide.*

*A mes très chère : mère, grand-mère source d'affection.*

*A mes très chères sœur Laila et Imene.*

*A mes très chère frère Aziz et smail.*

*A tous les membres de ma grande famille petits et grands.*

*À l'âme de tante Nacira et mes tantes Noura et Karima avec ses enfants sur tout Farah.*

*A mon oncle Amar.*

*A mon amie et binôme Kenza, pour sa compréhension, et sa tolérance lors de la réalisation de notre projet.*

*Et enfin a tous mes amis la spécialité GSI*

*BOUSBAINÉ Karim*

# *Dédicaces*

*Je dédie ce mémoire à :*

*Mes parents :*

*Ma mère, qui a œuvrée pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.*

*Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.*

*Mon frère MOH et mes sœurs Amina et Maïssa qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.*

*A ma copine et ma sœur ROUAM RANIA qui ma soutenue et ma aidée dans chaque étape de mon travail.*

*Et enfin à tous mes amis la spécialité GSI*

*Merdoud Kenza*

# Table des matières

<b>Table des matières</b> .....	i
<b>Liste des figures</b> .....	iv
<b>Liste des tableaux</b> .....	vi
<b>Liste des abréviations</b> .....	vii
<b>Introduction générale</b> .....	1
<b>1. Capteurs et traitement d'image</b> .....	<b>2</b>
1.1 Introduction .....	2
1.2 capteurs.....	2
1.2.1 Définition .....	2
1.2.2 Types de capteurs .....	2
1.2.3 Caractéristiques des capteurs .....	3
1.2.4 Classifications des capteurs .....	3
1.2.5 Exemples de capteurs .....	4
1.2.6 Architecture d' un capteur.....	4
1.3 Capteur d' image .....	5
1.3.1 Composition d'un capteur d' image .....	5
1.3.2 Les composants d'un capteur d' image .....	5
1.3.3 Mode de fonctionnement des capteurs d'images CMOS .....	5
1.3.4 Le CCD (charge coupled device) .....	6
1.3.5 Différence entre CMOS et CCD .....	6
1.4 Un réseau de capteurs .....	6
1.4.1 Utilisation traditionnelle des capteurs .....	8
1.4.2 Un exemple d'application.....	8
1.5 Environnements mobiles et réseaux de capteurs .....	9
1.5.1 Caractéristiques des environnements mobiles .....	9
1.5.2 Les réseaux cellulaires.....	10
1.5.3 Les réseaux Ad Hoc .....	10
1.6 Les réseaux de capteurs sans fil .....	11
1.6.1 Composants d'un réseau de capteurs sans fils .....	11
1.6.2 Composition d'un capteur sans fil .....	11
1.6.3 Facteurs de Conception des réseaux de capteurs sans fils.....	13

1.7 Acquisition d'image .....	13
1.7.1 Pourquoi traitons-nous les images .....	14
1.7.2 Historique du traitement d'image .....	14
1.7.3 Définition .....	15
1.7.4 Le besoin du traitement d'image .....	16
1.7.5 Les recherches en cours .....	18
1.7.6 Différents aspects du traitement des images .....	19
1.7.7 But du traitement d'image .....	20
1.8 Détection d'anomalie avec traitement d'image .....	21
1.8.1 ANOMALIE .....	21
1.8.2 La détection d'anomalies .....	21
1.8.3 Différents types de données .....	21
1.8.4 Techniques de détection basées sur l'apprentissage automatique .....	22
1.8.5 Techniques de détection des anomalies appliquées à la sécurité maritime .....	22
1.8.6 Les techniques de modélisation des comportements existantes .....	23
1.9 Conclusion .....	24
<b>2 . Réseaux de neurones et apprentissage profond .....</b>	<b>25</b>
2.1 réseaux de neurones .....	25
2.1.1 Définition.....	25
2.2 Le neurone .....	26
2.2.1 Le modèle le biologique .....	26
2.2.2 Vers une simulation du neurone biologique .....	22
2.2.3 Le modèle formel.....	22
2.3 Architecture des réseaux de neurones .....	29
2.3.1 Réseaux statiques "feed-forward " .....	29
2.3.2 Réseaux multicouches (ou Perceptron Multi Couche PMC) .....	30
2.3.3 Réseau récurrents "Feed-back".....	30
2.4 Apprentissage d'un réseau de neurone .....	31
2.4.1 Apprentissage supervisé .....	32
2.4.2 Apprentissage non supervisé .....	32
2.5 Apprentissage Profond .....	33
2.5.1 Introduction .....	33
2.5.2 Deep Learning DNNs (Réseaux de neurones profonds).....	33
2.5.3 Fonctionnement du deep learning.....	43
2.5.4 Applications du Deep Learning .....	35
2.5.5 Domaine du système deep Learning.....	35

2.5.6 Thèmes récurrents de l'apprentissage en profondeur "	36
2.6 Conclusion	37
<b>3. Implémentation</b>	<b>38</b>
3.1 Introduction	38
3.2 conception .	38
3.3 Présentation des outils	39
3.3.1 Le software	39
3.3.2 Le hardware	42
3.3.3 L'implémentation	42
3.3.4 Validation	44
3.4 Conclusion .	46
<b>Conclusion générale</b>	<b>47</b>
<b>Bibliographie</b>	<b>48</b>
<b>Resumé</b>	<b>51</b>

# Liste des figures

<b>Figure I.1 : un capteur</b> .....	3
<b>Figure I.2 : exemples de capteurs</b> .....	4
<b>Figure I.3 : Architecture d'un capteur</b> .....	4
<b>Figure I.4: Différence entre CMOS et CCD</b> .....	6
<b>Figure I.5 : schéma d'un réseau de capteur</b> .....	7
<b>Figure I.6 : exemples d'utilisation traditionnel des capteurs</b> .....	8
<b>Figure I.7 : Modèle des réseaux cellulaires</b> .....	10
<b>Figure I.8 : composants d'un capteur sans fils</b> .....	12
<b>Figure I.9 : Différents appareils de détection des images</b> .....	13
<b>Figure I.10: histoire du traitement d'image</b> .....	15
<b>Figure I.11 : Traitement d'image et domaines connexes</b> .....	16
<b>Figure I.12 : Diagramme montrant différentes phases du traitement D'images numériques</b> .....	17
<b>Figure I.13 : Image représentationnelle des rayons X, IRM et tomographie assistée par ordinateur (CT)</b> .....	18
<b>Figure I.14 : Représentation schématique des différentes étapes impliquées dans un processus complet de traitement d'image</b> .....	20
<b>Figure II.1 : Schéma général d'un réseau de neurones</b> .....	25
<b>Figure II.2 : Un neurone biologique et ses principaux composants</b> .....	26
<b>Figure II.3. Schéma d'un neurone formel</b> .....	27
<b>Figure II.4 :Fonction de transfert : du neurone «linéaire» ; du neurone «seuil symétrique » ; du neurone «linéaire saturée symétrique» et du neurone «tangente hyperbolique»</b> .....	29
<b>Figure II.5 : Exemple d'un réseau de neurones non bouclé</b> .....	29
<b>Figure II.6 : Perceptron Multi Couche PMC</b> .....	30
<b>Figure II.7 : Forme canonique d'un réseau de neurones bouclé</b> .....	31
<b>Figure II.8 : schéma d'un réseau de neurone profond</b> .....	33
<b>Figure II.9 : Gradient Descent</b> .....	35
<b>Figure III.1: Conception du classificateur</b> .....	38
<b>Figure III.2 : des poumons normaux</b> .....	41
<b>Figure III.3 : des poumons malades</b> .....	41



<b>Figure III.4 : graphe de précision pour le modèle .....</b>	<b>44</b>
<b>Figure III.5 : graphe d'erreurs pour le modèle.....</b>	<b>45</b>
<b>Figure III.6 : résultat de la classification .....</b>	<b>45</b>

# Liste des tableaux

<b>Tableau 2.1</b> : Quelques fonctions d'activation .....	28
<b>Tableau 2.2</b> : Exemple de différents types d'apprentissages en mode supervisé.....	32
<b>Tableau 2.3</b> : Exemple de différents types d'apprentissages en mode non supervisé .....	32

# Liste des abréviations

<b>GPS</b>	Global Positioning System
<b>RAM</b>	Random Access Memory
<b>CMOS</b>	Complementary Metal Oxyde Semiconductor
<b>CCD</b>	Charge Coupled Device
<b>MOS</b>	Métal Oxyde Semi-conducteur
<b>GSM</b>	Global System for Mobile
<b>WSN</b>	wireless Sensor Networks
<b>IRMI</b>	magerie par Résonance Magnétique
<b>HMM</b>	Hidden Markov Model
<b>PMC</b>	Perceptron Multi Couche
<b>DNN</b>	Deep Neural Network
<b>DL</b>	Deep Learning
<b>IA</b>	Intelligence artificiel
<b>DP</b>	Programmation Dynamique
<b>SL</b>	Supervised Learning
<b>UL</b>	Unsupervised Learning
<b>RL</b>	Reinforcement learning
<b>MDL</b>	Minimum Description Length
<b>GPU</b>	Graphique Processing Unit
<b>RNN</b>	Recurent Neural Network

# Introduction générale

L'image est depuis toujours, considérée comme un moyen de communication efficace attirant et plus approprié, pour transmettre les messages entre différentes personnes ; elle est dite le langage universel ; qui traduit les informations sans se restreindre à une communauté de personnes particulière.

Le contenu des images est analysé par le traitement d'image, par différentes manières tel que : la segmentation, l'extraction, l'indexation et la classification.

Cependant, les avancées technologiques en terme d'acquisitions d'images (camera, microscope, capteur) et de stockage génèrent des bases de données riches en informations et multiplient les champs d'applications.

Grace aux progrès réalisés dans les domaines des matériaux, de l'électronique et de l'informatique, une multitude de capteurs à haute performance sont très variés et leurs domaines d'applications très vastes. Et on distingue plusieurs types de capteurs ; d'image, de son.

Après la détection des images par les différents capteurs, on les classe et leur fait un apprentissage avec différentes méthodes, comme les réseaux de neurones et deep learning, afin de développer un système pouvant affecter automatiquement une classe à une image.

À la fin pour faire l'apprentissage il était prévu de réaliser une base de données pour les volailles (saines et malades) et à cause d'un manque de temps on n'a pas pu les avoir, pour cela on a choisi une autre base de données.

Ce mémoire est structurée en trois chapitres comme suit:

Premier chapitre : ce chapitre est dédié à une introduction aux différents capteurs et leurs fonctionnements, et le traitement d'images et la détection des anomalies dans ces images.

Deuxième Chapitre : dans ce chapitre on parle sur les différentes techniques d'apprentissage comme les réseaux de neurones et les réseaux de neurones convolutionnels et le deep learning.

Troisième Chapitre : au cours de ce chapitre on a présenté les différents outils de programmation utilisés et les résultats obtenus après l'apprentissage.

# Chapitre 1

## 1.1 Introduction

Dans ce chapitre nous parlerons des capteurs et leur fonctionnement. En suite, nous aborderons le volé traitement des images (l'acquisition, la détection des anomalies, et leur application dans des différents domaines) dont l'objectif de l'utilisé ou de l'exploré dans le domaine des capteurs.

## 1. 2 capteurs

### 1. 2.1 Définition

Un **capteur** (senser) est un dispositif sensible à un phénomène déterminé et transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une déviation d'une aiguille, une tension électrique, une hauteur de mercure, ou une image-vidéo.

Il s'agit d'une simple interface entre un processus physique et une information manipulable [1].

### 1.2. 2 Types de capteurs :

#### Les simples capteurs :

il ya plusieurs types de capteurs on peut citer [2]:

- ✓ capteurs de position
- ✓ Capteur de vitesse
- ✓ Capteur d'accélération
- ✓ Capteur de pression
- ✓ Capteur de mouvement
- ✓ Capteur de luminosité
- ✓ Capteur de température

#### Capteurs plus complexes :

- ✓ Capteurs de son,
- ✓ capteurs d'images



Figure I.1 : un capteur

### 1.2.3 Caractéristiques des capteurs

- Dispositifs physiques de petite taille.
- Capacité énergétique limitée dans le temps.
- Fonction principale : acquisition de données :
- température, humidité, etc.
- Géopositionnement (données GPS).
- Dotés de faibles capacités de calcul et de stockage.
- Des capacités de communication sans fil :
- ZigBee (802.15.4), WiFi (802.11), etc.
- Différents types de capteurs (Fixes, Agiles ou Mobiles).

### 1.2.4 Classifications des capteurs

on peut classer les capteurs comme suit [3]:

#### Des capteurs fixes

- Implantés définitivement dans un endroit.
- Effectuant des mesures pour un endroit unique.

#### Des capteurs agiles

- Variation discrète de la position.

#### Des capteurs mobiles

- Positions variant dans le temps et dans l'espace.
- Architecture d'un capteur sans fil
- Unité de traitement : processeur.
- Unité de stockage : mémoire "flash" ou mémoire vive (RAM).
- Unité d'énergie : batterie.
- Unité de détection : convertisseur analogique/numérique.
- Unité de transmission : antenne et récepteur.

### 1.2.5 Exemples de capteurs



#### Imote2

Multimédia/Scalaire

Caméra, audio & détection de mouvement, accéléromètre 3 axes, capteurs d'humidité et de luminosité



#### MicaZ

Scalaire

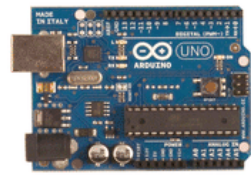
Accéléromètre 2 axes, capteurs d'humidité et de luminosité  
pression atmosphérique



#### TelosB

Scalaire

Capteurs de luminosité, visible et infrarouge, température, humidité, haut-parleur et led



#### Arduino

Multimédia/Scalaire

Extensions (shield) : Capteurs de luminosité, visible et infrarouge, température, humidité, haut-parleur et led  
Wifi, gps, rfid, etc

Figure I.2 : exemples de capteurs

### 1.2.6 Architecture d'un capteur

Le premier rôle d'un capteur est de détecter un événement (exp : des mouvements, un changement de température, ...). Il est donc capable de récolter des données relatives à son environnement, de les traiter, puis, si nécessaire, de les communiquer à des capteurs voisins via un médium sans fil [4].

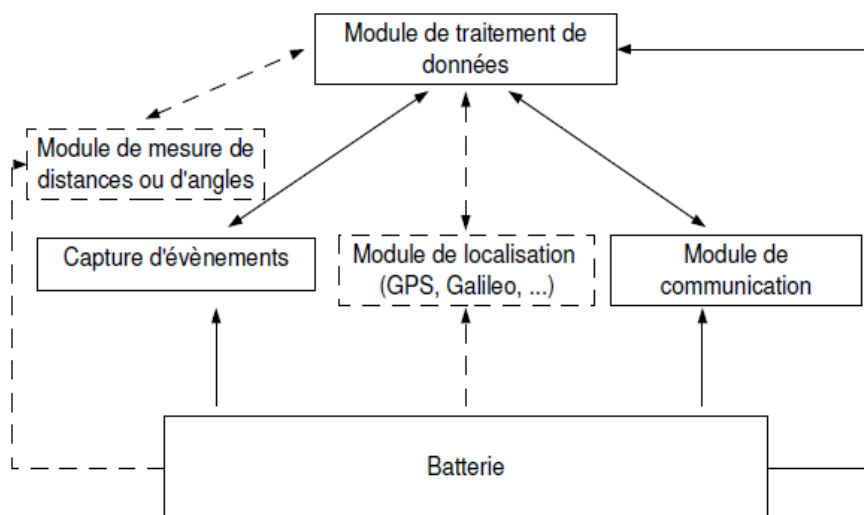


Figure I.3 : Architecture d'un capteur.

- Un module de capture : il détecte les événements ayant lieu dans son rayon de Perception.
- un module de traitement : doté d'un processeur et d'une mémoire, il traite les données détectées et éventuellement les communique.
- un module de communication : il se charge de la transmission et de la réception des données via un médium sans fil.

- un module de localisation (optionnel) : il peut s'agir d'un système GPS (Parkinson et al, 1996), Galileo (CNES et ESA) ou d'un autre système de localisation qui donne au capteur sa position exacte.
- un module de mesure (optionnel) : il mesure la distance ou l'angle avec un capteur voisin.
- une batterie : elle alimente tous les autres modules.

Les interactions entre ces modules sont illustrées sur la figure 3 ci-dessus. Chacun d'entre eux est alimenté par la batterie. La consommation d'énergie est essentiellement due aux modules de communication sans fil et de traitement des données. Le module optionnel de localisation est également une source de consommation d'énergie non négligeable.

### 1.3 Capteur d'image

Permet de transformer l'énergie lumineuse de chaque point de l'original en un signal électrique.

#### 1.3.1 Composition d'un capteur d'image

- Un capteur numérique est considéré comme un assemblage de couches :  
Des filtres limitent la lumière au spectre visible et écartent les petits détails impossibles à enregistrer.
- Des microlentilles permettent de correctement guider la lumière jusqu'aux récepteurs.
- Une matrice de Bayer décompose la lumière en composantes Rouge, Vert et Bleue (RVB).
- Des photosites transforment la lumière reçue en signal électrique.

#### 1.3.2 Les composants d'un capteur d'image

Pour transformer la lumière en fichier numérique le capteur réalise plusieurs étapes :

- Filtrer les informations inutiles, pour éviter d'enregistrer les infrarouges (c'est le rôle du filtre IR) et les très petits détails de l'image qu'il est impossible de correctement enregistrer (filtre AA, pour "Anti-Aliasing").
- Guider les rayons lumineux vers les récepteurs (les photosites) à l'aide de microlentilles.
- Séparer la lumière selon ses 3 composantes Rouge, Verte et Bleu, avec la matrice de Bayer.
- Convertir les photons (la lumière) en électrons (l'électricité), ce que font les photosites.

#### 1.3.3 Mode de fonctionnement des capteurs d'images CMOS

La détection du signal des capteurs CMOS repose sur le même principe que les CCD : on crée sous la surface du silicium une région appauvrie en porteurs libres, dans lequel se trouve un champ électrostatique interne. La lumière incidente apporte dans cette zone l'énergie suffisante pour permettre le passage d'un électron dans sa bande de conduction ( $E_G > 1.12\text{eV}$  pour le silicium). La paire électron-trou ainsi créée est alors séparée par le champ électrique interne, ce qui produit un courant que l'on appellera 'courant photonique.



### 1.3.4. Le CCD (charge coupled device)

Consiste en une juxtaposition matricielle de capacités MOS. Pour obtenir l'image de la scène, il faut compter le nombre d'électrons piégés dans chaque puits, qui est proportionnel au nombre de photons incidents sur chaque pixel. Pour ce faire, on commande séquentiellement la tension des grilles de chaque capacité MOS au rythme d'une horloge, ce qui va permettre de transférer les électrons d'une capacité vers sa voisine : on parle de registre à décalage. Après  $n$  transferts, les charges sont converties en tension dans un condensateur puis amplifiées. Elles sont ensuite codées numériquement à l'aide d'un convertisseur analogique/numérique à l'extérieur de la matrice CCD.

### 1.3.5. Différence entre CMOS et CCD

	CCD	CMOS
Facteur de remplissage	50 à 98 %	30 à 50 %
Sensibilité à la lumière	Élevée (0,1 lux)	Plus faible (10 lux)
Sensibilité aux UV et IR	Étendue	Étroite
Réponse	Moyenne	Très rapide (10 à 100x plus)
Consommation d'énergie	Élevée (2 à 4x plus que CMOS)	Faible
Dynamique	Élevée	Moyenne
Uniformité du capteur	Élevée (1 à 2 % de FPN*)	Faible à modérée (5 à 20 % de FPN*)
Bruit électronique	Faible	Moyen à élevé (10x plus que CCD)
Courant d'obscurité	Faible	Moyen
Sélection d'une région d'intérêt	Très limité	Élevé
Anti blooming	Absent (sauf solution custom)	Présent par défaut
Shuttering	Assez rapide	Faible
Biasing et clocking	Multiple, tension élevée	Unique, basse tension

Figure I.4: Différence entre CMOS et CCD.

## 1.4 Un réseau de capteurs

Un ensemble de capteurs autonomes à faible coût, interconnectés par un réseau de communications, Pour rendre un service de mesures dans une certaine zone géographique qui coopèrent pour acquérir et transmettre des mesures. De nombreuses avancées techniques et technologiques dans les domaines de la micro-électronique, de la micromécanique, et des technologies de communication sans fil, permettent de créer de petits objets communicants équipés de capteurs a un coût raisonnable [5].

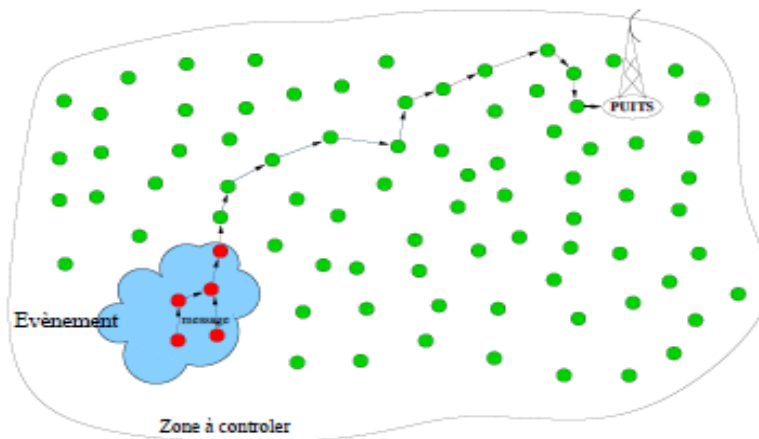
Ces nouveaux objets appelés nœud sou capteurs sont équipés d'une unité de mesure (les capteurs), d'une unité de calcul, de mémoires et d'une radio pour communiquer. Enfin, pour l'alimentation, ces nœuds possèdent une pile ou un système de récupération d'énergie dans l'environnement.

Cette technologie rend possible le déploiement de réseaux de capteurs sans fil. Les réseaux de capteurs ont de nombreuses perspectives d'application dans des domaines très variés : applications militaires, domotique, surveillance industrielle ou de phénomènes naturels, relevé de compteurs.

Chaque application a ses propres contraintes. Dans tous les domaines, le rôle d'un réseau de capteurs est cependant à peu près toujours le même, voir figure 4.

Les nœuds doivent surveiller certains phénomènes grâce à leurs capteurs puis envoient les informations à un puits. Le puits est un nœud particulier doté d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie.

Ce puits peut être connectée à Internet ou possède un lien radio de type GSM ou GPRS qui lui permet d'envoyer les informations (données ou alertes) à un centre de contrôle pour l'utilisateur final. Il peut y avoir plusieurs puits mobiles ou fixes dans un réseau mais pour des raisons de coût, il y a moins de puits que de nœuds.



**Figure I.5 :schéma d'un réseau de capteur.**

Ce type d'application implique des contraintes communes à de nombreux réseaux de capteurs. Les capteurs sont employés sur la zone à contrôler. Comme celle-ci est étendue et potentiellement difficile d'accès, ils doivent s'auto-organiser pour envoyer de proche en proche les messages jusqu'au puits.

On ne remplace pas les nœuds qui ne fonctionnent pas ou qui n'ont plus d'énergie, il faut donc qu'ils vivent le plus longtemps possible avec une quantité d'énergie très faible étant donnée le coût élevé des piles. Pour certaines applications de surveillance où le trafic est très sporadique, on souhaite plusieurs dizaines d'années d'autonomie pour le réseau. Ces réseaux sont composés d'un très grand nombre de nœuds.

### 1.4.1 Utilisation traditionnelle des capteurs

les capteurs sont utilisés souvent dans différents domaines comme [6] :

- l'aéronautique.
- l'automobile.

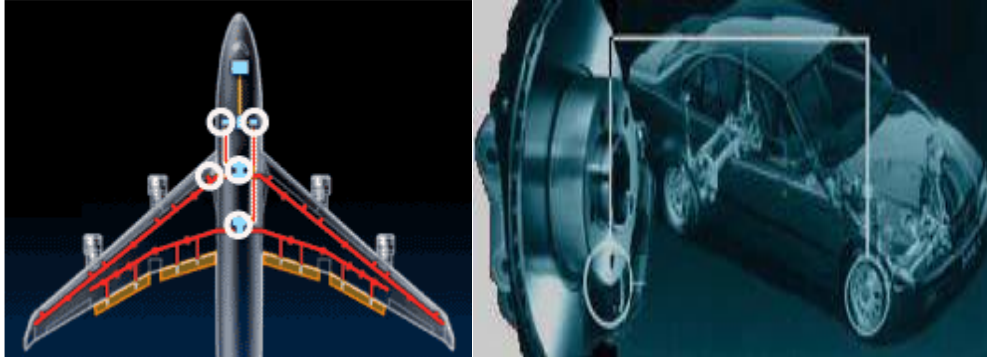


Figure I.6 : exemples d'utilisation traditionnelle des capteurs

### 1.4.2 Un exemple d'application

il existe plusieurs applications, parmi elles [7] :

**Surveillance de l'environnement** : air, eau, ou terre

Domaines d'application :

- Surveillance
  - Détection d'intrusions
  - Feu de forêt
  - Mesures météorologiques
  - Départ de coups d'artillerie
  - Détection des mouvements ennemis
  - Contrôle de la qualité de l'air
- Couverture
  - Zone/point ponctuel
- Environnement
  - Militaire /Civil
  - Hostile/non-hostile
- Applications
  - Agricole et Environnement
  - Médical
  - Structure ou édifice
  - Domotique

## 1.5 Environnements mobiles et réseaux de capteurs

L'évolution rapide de la technologie dans le domaine de communication sans fil, a permis à des usagers munis d'unités de calcul portables d'accéder à l'information indépendamment des facteurs : temps et lieu. Ces unités, qui se communiquent à travers leurs interfaces sans fil, peuvent être de diverses configurations : avec ou sans disque, des capacités de sauvegarde et de traitement plus ou moins modestes et alimentés par des sources d'énergie autonomes (batteries). L'environnement de calcul résultant est appelé *environnement mobile* (ou nomade). Cet environnement n'astreint plus l'utilisateur à une localisation fixe, mais lui permet une libre mobilité tout en assurant sa connexion au réseau[8].

Les environnements mobiles permettent une grande flexibilité d'emploi. En particulier, ils permettent la mise en réseau des sites dont le câblage serait trop onéreux à réaliser dans sa totalité, voire même impossible (par exemple en présence d'une composante mobile).

L'environnement mobile offre beaucoup d'avantages par rapport à l'environnement habituel. Cependant de nouveaux problèmes peuvent apparaître (le problème de routage), causés par les nouvelles caractéristiques du système. Les solutions conçues pour les systèmes distribués avec des sites statiques, ne peuvent pas donc être utilisées directement dans un environnement mobile.

Les réseaux mobiles ou sans fil peuvent être classés en deux catégories, les réseaux cellulaires avec infrastructure et les réseaux ad hoc sans infrastructure fixe.

### 1.5.1 Caractéristiques des environnements mobiles

#### La communication sans fil

La communication sans fil dans les réseaux mobiles est beaucoup moins fiable que celle basée sur les réseaux filaires. La propagation des signaux utilisés subit des perturbations (microcoupures ou erreurs de transfert) dues à l'environnement, qui altèrent l'information transmise, il s'ensuit alors, un accroissement du délai de transit des messages à cause de l'augmentation du nombre de retransmissions.

En effet, les liaisons de communication sans fil présentent un taux d'erreurs plus important comparées aux liaisons filaires dans les réseaux statiques. Tandis que ces derniers offrent un taux d'erreur de l'ordre de  $10^{-5}$  sur une ligne téléphonique,  $10^{-7}$  sur un câble coaxiale et  $10^{-12}$  sur une fibre optique.

Les liaisons sans fil connaissent un taux d'erreur allant de  $10^{-2}$  jusqu'à  $10^{-6}$ . De plus, ces liaisons sont très sensibles au phénomène d'interférence causé par la présence de plusieurs unités mobiles dans une même région de communication. L'une des limitations principales connues par les environnements mobiles vient de la faiblesse de la bande passante des médias de communication sans fil utilisés.

On distingue les réseaux utilisant l'infrarouge avec un débit de 1Mbps, la communication radio dans les réseaux locaux sans fil avec 2 à 11 Mbps et celle dans les réseaux des régions étendues (Wireless WANs) qui possèdent une bande passante égale à 19,2 kbps. La bande passante limitée dans les réseaux mobile affecte les performances des protocoles distribués[9].

### 1.5.2 Les réseaux cellulaires

Les réseaux cellulaires sont généralement décomposés en deux ensembles d'entités distinctes, les sites fixes appartenant à un réseau filaire classique, et les sites mobiles. Certains sites fixes appelés stations support mobile (Mobile support station) ou station de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les unités mobiles (UM) localisées dans une zone géographique limitée appelée cellule (figure 6).

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages via des liaisons sans fil ayant une bande passante limitée qui réduit sévèrement le volume d'informations échangées. Toutefois, l'ensemble des stations de base sont connectées entre eux par un réseau de communication filaire, généralement fiable et d'un débit élevé.

A un instant donné, une unité mobile n'est directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station de base à laquelle elle est directement rattachée.

Dès lors, et pour envoyer un message d'une unité mobile UM1 à une autre unité mobile UM2, UM1 doit envoyer le message à sa station de base SB1 à travers le réseau sans fil qui le transmet à la station de base de l'unité UM2 appelée SB2, qui à son tour le transmet à UM2.

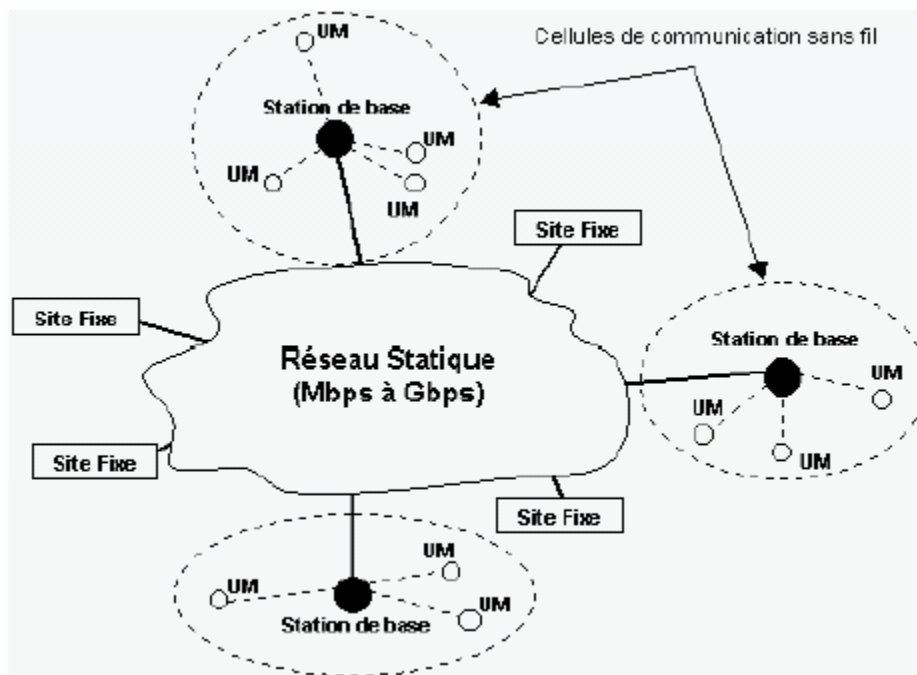


Figure I.7 : Modèle des réseaux cellulaires

### 1.5.3 Les réseaux Ad Hoc

Les réseaux ad hoc sont auto organisés et ne se reposent sur aucune infrastructure fixe. Les éléments les composants sont généralement reliés par radio, potentiellement mobiles et peuvent être amenés à entrer ou sortir du réseau à tout moment [10].

Ces réseaux sont caractérisés par la capacité de chaque participant d'agir à la fois comme client et comme routeur du réseau. Si un émetteur n'est pas à portée directe de la machine destination, les informations devront être transmises de proche en proche, le long d'un chemin établie et maintenu par le réseau en cas de modification de la topologie (ces réseaux sont d'ailleurs qualifiés de multi-saut).

Le domaine des réseaux ad hoc est vaste et récent. Un travail est conséquent et actuellement effectué dans le groupe MANET et l'IETF dans le but de normaliser un ou des protocoles de routage pour ces réseaux.

Ces travaux, ainsi que de nombreux autres portant sur des protocoles de plus haut niveau (qualité de service, découverte de service,...) s'appuient sur des simulations utilisant en particulier la norme 802.11 de IEEE.

## 1.6 Les réseaux de capteurs sans fil

Un réseau de capteurs sans fil (WSN) est un système distribué de grande échelle mettant en communication un grand nombre d'entités autonomes communément appelées capteurs sans fils.

Ces capteurs forment donc les nœuds du réseau.

Dans le cas d'un réseau à grande échelle, les capteurs ne sont pas tous dans le voisinage du puits et les messages seront acheminés du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts.

Un réseau de capteurs sans fil est un type particulier de réseaux ad-hoc qui sont utilisés pour l'interconnexion spontanée des systèmes informatiques.

Les principales différences entre les réseaux de capteurs sans fil et les réseaux ad-hoc traditionnels sont le problème de l'énergie et le facteur d'échelle, exemple (ad hoc sa taille considérée sur une échelle entre 10 et 100 nœuds) [11].

### 1.6.1 Composants d'un réseau de capteurs sans fils

Un réseau de capteurs sans fil générique est composé d'un grand nombre de nœuds capteurs dispersés dans le terrain d'intérêt appelé *champ de captage*.

Les nœuds ont la possibilité de collecter périodiquement les données sur le phénomène surveillé et envoyer les rapports de captage à un nœud spécial appelé *puits* (sink).

Ce nœud est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises aux capteurs via des messages de requêtes.

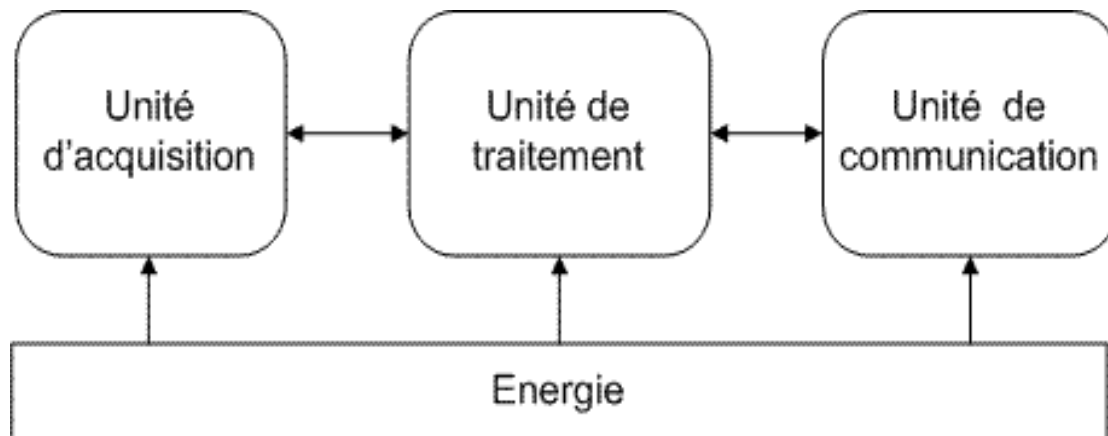
Un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts différents.

Par exemple, un nœud puits peut demander à tous les capteurs se trouvant dans la région nord du champ de captage d'envoyer un rapport de température chaque 1 minute, pendant qu'un autre peut être intéressé seulement par la haute température ( $> 40^{\circ} \text{C}$ ) dans la région sud. Par conséquent, un capteur doit pouvoir stocker toutes les requêtes reçues, et les traiter séparément [12].

### 1.6.2 Composition d'un capteur sans fil

Les capteurs sans fil considérés ici sont conçus comme de véritables systèmes embarqués, dotés de moyens de traitement et de communication de l'information, en plus de leur fonction initiale de relever des mesures. Ils représentent une révolution technologique des instruments

de mesure, issue de la convergence des systèmes électroniques miniaturisés et des systèmes de communication sans fils[13].



**Figure I.8 : composants d'un capteur sans fils**

Un capteur sans fil est composé fondamentalement de quatre unités élémentaires :

A. **Unité de Captage** : Ce composant est l'unité qui contient le ou les capteurs embarqués sur le nœud.

Habituellement, un convertisseur analogique-numérique (CAN) convertit les signaux provenant des capteurs (signaux analogiques) en signaux interprétables par l'Unité de Traitement (signaux numériques).

B. **Unité de Traitement** : Elle est généralement constituée d'un microcontrôleur dédié et de la mémoire.

Cette unité fournit aux capteurs la capacité d'exécuter des calculs sur les données et les conserver selon un scénario programmé. Bien que ce ne soit pas obligatoire, il est souhaitable qu'il existe des moyens de reprogrammer facilement les capteurs dans le cas d'un changement dans les exigences de l'application.

C. **Unité de Communication** : Elle est le plus souvent constituée d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein d'un réseau. Elle met en œuvre des protocoles de communication dépendant de la technologie utilisée (par exemple 802.11, 802.15.1, 802.15.4, etc. pour les technologies sans fil), tandis que les protocoles de plus haut niveau (routage, Localisation, etc.) sont mis en œuvre dans l'Unité de Traitement. Certaines technologies radio permettent de changer la fréquence et la puissance de transmission.

D. **Unité de Puissance** : Comme il est souhaitable de s'affranchir de toute connexion par câble, le capteur doit disposer de sa propre source d'énergie qui alimente le reste des unités. Cette unité se trouve généralement sous la forme de batterie standard de basse tension.

En fonction des applications pour lesquelles ils sont conçus, les capteurs sans fil pourraient également avoir d'autres modules, comme une Unité de Localisation, afin d'identifier leur position géographique, par exemple en utilisant un récepteur GPS ou une technique de triangulation. Certaines applications pourraient aussi avoir besoin de capteurs équipés d'un Mobilisateur pour qu'ils puissent se déplacer.

Enfin, s'il est nécessaire qu'un nœud soit maintenu en activité pendant une très longue période de temps, un Générateur de Puissance, tel que des cellules solaires, serait utile afin de tenir le nœud alimenté électriquement sans avoir à changer ses batteries.

### 1.6.3. Facteurs de Conception des réseaux de capteurs sans fils [14] :

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres, parmi lesquels on cite :

- la tolérance aux pannes,
- la scalabilité,
- le coût de production,
- l'environnement d'exploitation,
- la topologie du réseau,
- les contraintes matérielles,
- le support de transmission et la consommation d'énergie.

Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine.

Après avoir vu les capteurs on va parler sur les concepts fondamentaux du traitement d'image et la détection des anomalies.

## 1.7 Acquisition d'image

L'étude de l'image étudie les mécanismes physiques par lesquels un type particulier de modèles d'imagerie génère une observation d'image, et étudie également le modèle mathématique associé et l'algorithme employé par l'ordinateur intégré dans de tels dispositifs d'imagerie.[16]



Figure I.9 : Différents appareils de détection des images



### 1.7.1. Pourquoi traitons-nous les images

on traite les images pour différentes raisons [17] :

#### Acquérir l'image

- ✓ Bonne ouverture et balance des couleurs.
- ✓ Reconstruire l'image à partir de la projection.

#### Préparer pour l'affichage et l'impression :

- ✓ Ajuster la taille de l'image.
- ✓ Cartographie des couleurs, correction gamma, demi-teinte.

#### Faciliter le stockage et la transmission de l'image :

- ✓ Stocker efficacement une image dans un appareil photo numérique.
- ✓ Envoyer une image depuis l'espace.

#### Améliorer et restaurer des images :

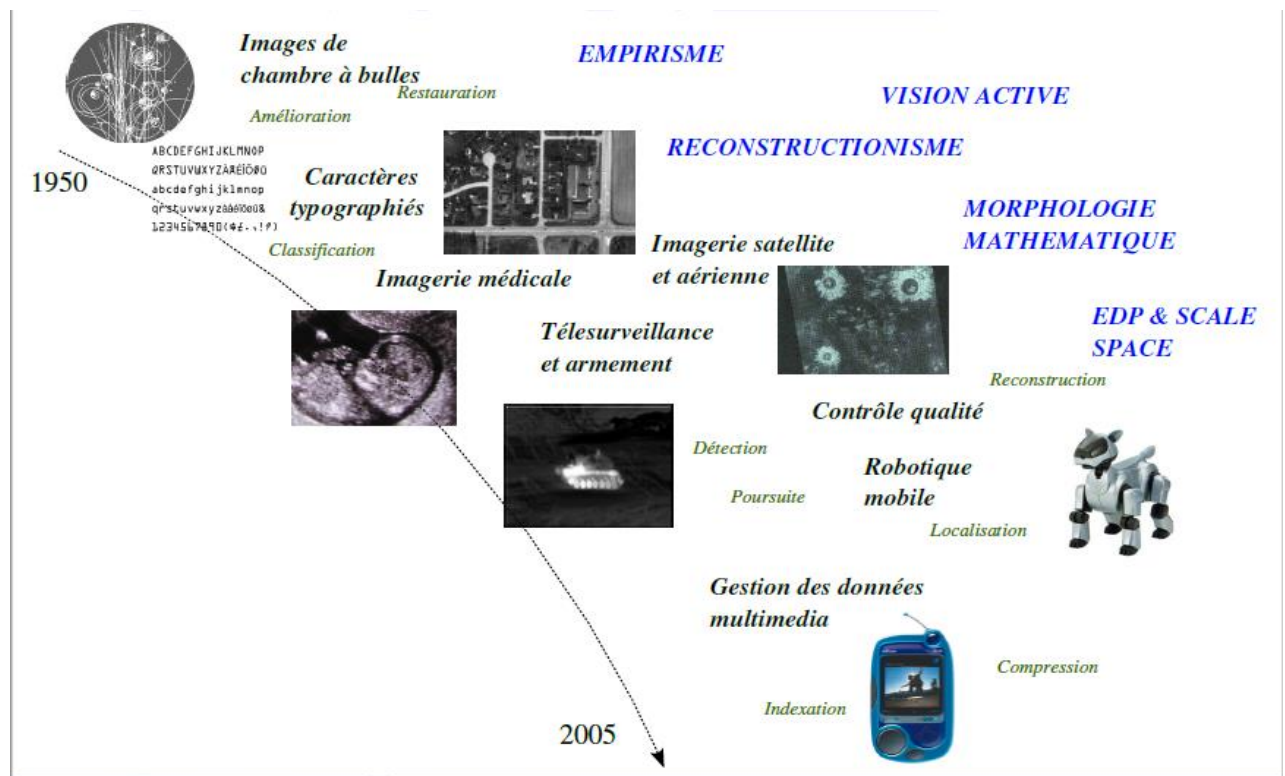
- ✓ Retoucher les photos personnelles.
- ✓ Améliorer des couleurs pour le filtrage de sécurité.

#### Extraire des informations à partir des images :

- ✓ Lire les codes à barres 2D.
- ✓ Reconnaissance de caractères.

### 1.7.2 Historique du traitement d'image

Le traitement d'image est né de l'idée et de la nécessité de remplacer l'observateur humain par la machine. L'image ou les signaux provenant des capteurs ont alors été codés et mémorisés sur différents supports à revoir ce morceau de phrase car il n'a aucun sens. Le champ des applications du traitement d'images est très vaste[18].



**Figure I.10: histoire du traitement d'image.**

### 1.7.3. Définition

Le traitement d'image est une science et technique par essence multidisciplinaires. De par les connaissances qu'il est nécessaire de maîtriser sur la nature du signal représenté, le développement d'un outil de traitement d'image ne se conçoit pas en l'absence de solides notions sur la discipline au sein de laquelle l'image est exploitée. Ces connaissances vont de l'art pictural aux mathématiques, en passant par les sciences humaines, la médecine et les sciences expérimentales. L'opération de formation d'une image conditionne ses caractéristiques et une connaissance approfondie de la physique qui sous-tend la création du signal et sa détection.[19]

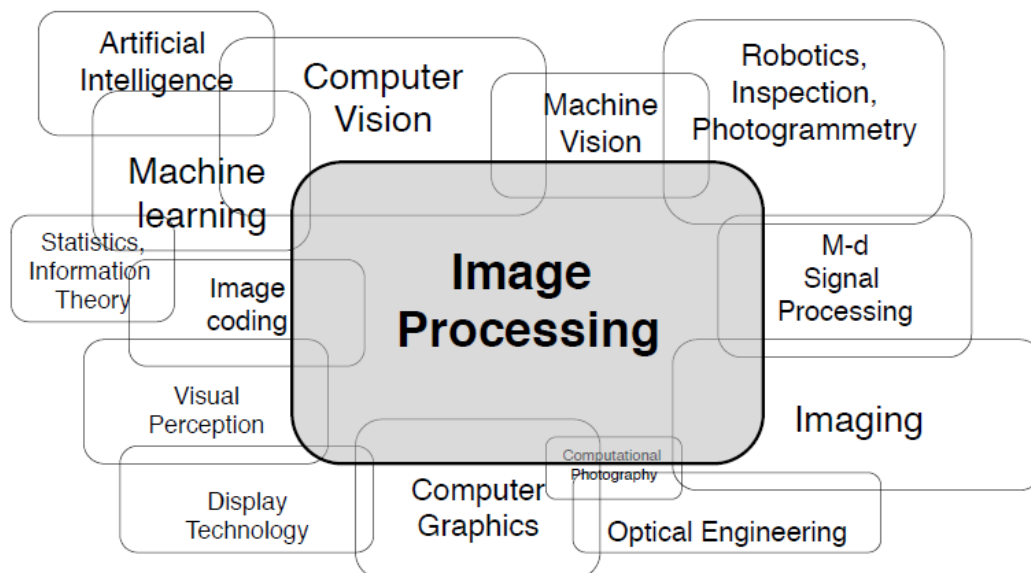
Le traitement d'images constitue également un domaine de recherche essentiel dans les disciplines de l'ingénierie et de l'informatique.

Il comprend essentiellement les trois étapes suivantes :

- Importation de l'image avec un scanner optique ou par photographie numérique.
- Analyser et manipuler l'image qui comprend la compression des données et l'amélioration de l'image.
- Un résultat final qui est basé sur l'analyse d'image qui, ce résultat est considéré comme une sortie et qui peut être modifiée.[20]

### 1.7.4 Le besoin du traitement d'image :

Le besoin en applications de traitement d'images se fait de plus en plus pressant à mesure que l'image numérique s'impose comme un support et une source d'information privilégiés. La multiplication des dispositifs d'acquisition conduit à la production d'un très grand nombre d'images qu'il devient difficile d'exploiter manuellement. Ces dispositifs permettent d'accéder à des informations jusqu'ici inconnues ou inaccessibles, qui prennent aujourd'hui une importance stratégique dans de nombreux domaines comme la biomédecine, la télésurveillance, le contrôle qualité, etc. Le traitement d'images s'inscrit dans un processus préliminaire destiné à préparer automatiquement les images à leur analyse, leur interprétation, leur restauration, leur archivage ou leur transmission. Il n'a aucun pouvoir décisionnel, mais son rôle est crucial puisqu'il consiste à extraire des images des informations qualitatives ou quantitatives par des procédures de réduction et d'abstraction de l'information initiale sans perte ni falsification de l'information pertinente.[21]

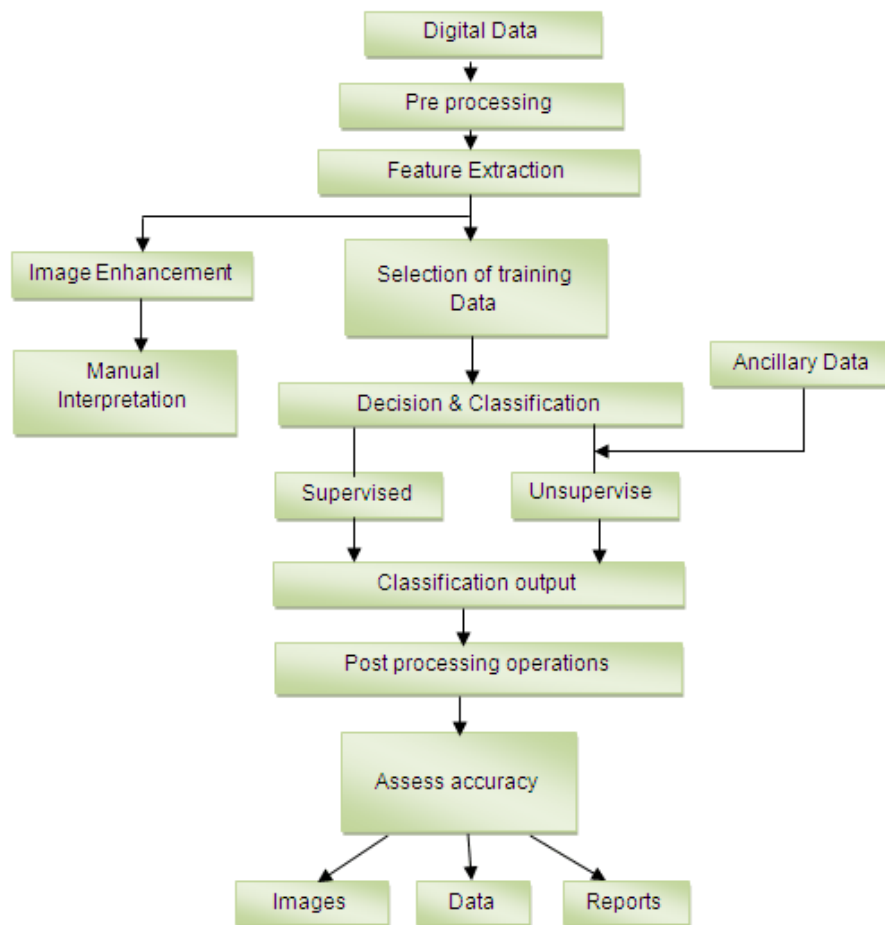


**Figure I.11 : Traitement d'image et domaines connexes : [22]**

#### Les types

Les deux types de méthodes utilisées pour le traitement d'image sont :

- Des techniques analogiques ou visuelles utilisées pour les copies papier comme les impressions et les photographies. Les analystes d'images utilisent différents principes fondamentaux d'interprétation tout en utilisant ces techniques visuelles. Le traitement de l'image ne se limite pas à la zone à étudier mais à la connaissance de l'analyste. L'association est un autre outil important dans le traitement de l'image grâce à des techniques visuelles. Les analystes appliquent donc une combinaison de connaissances personnelles et de données collatérales au traitement de l'image.
- Les techniques de traitement numérique aident à manipuler les images numériques en utilisant des ordinateurs. Pour cela il faut subir différentes phases de traitement, en utilisant la technique numérique qui sont le prétraitement, l'amélioration et l'affichage, l'extraction d'information.



**Figure I.12 : Diagramme montrant différentes phases du traitement D'images numériques**

#### **Applications :**

**Systèmes de transport intelligents :** Cette technique peut être utilisée pour la reconnaissance automatique des plaques d'immatriculation et la reconnaissance des panneaux de signalisation.

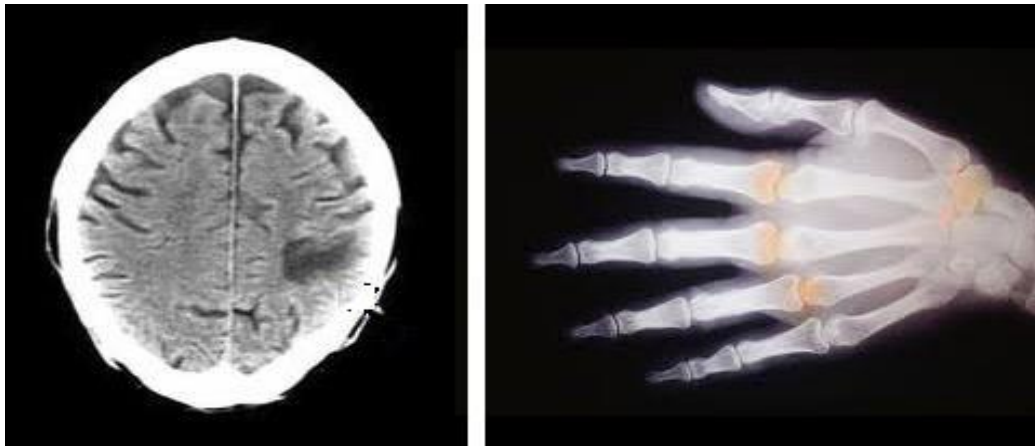
**Téledétection :** Pour cette application, les capteurs capturent les images de la surface de la Terre dans des satellites de téledétection ou dans un scanner multi-spectral monté sur un avion. Ces images sont traitées en les transmettant à la station terrienne. Les techniques utilisées pour interpréter les objets et les régions sont utilisées dans le contrôle des inondations, la planification urbaine, la mobilisation des ressources, le suivi de la production agricole, etc.

**Suivi des objets en mouvement :** Cette application permet de mesurer les paramètres de mouvement et d'acquérir un enregistrement visuel de l'objet en mouvement. Les différents types d'approche pour suivre un objet sont:

- Suivi basé sur le mouvement
- Suivi basé sur la reconnaissance

**Surveillance de la défense :** Les méthodes de surveillance aérienne sont utilisées pour surveiller en permanence la terre et les océans. Cette application est également utilisée pour localiser les types et la formation des navires de la surface de l'océan. Le devoir important est de diviser les différents objets présents dans la partie du corps de l'eau. Les différents paramètres tels que la longueur, la largeur, la surface, le périmètre, la compacité sont définis pour classer chacun des objets divisés. Il est important de reconnaître la distribution de ces objets dans différentes directions: est, ouest, nord, sud, nord-est, nord-ouest, sud-est et sud-ouest pour expliquer toutes les formations possibles des navires. Nous pouvons interpréter l'ensemble du scénario océanique à partir de la distribution spatiale de ces objets.

**Techniques d'imagerie biomédicale :** Pour le diagnostic médical, différents types d'outils d'imagerie tels que les rayons X, l'échographie, la tomographie assistée par ordinateur (CT), etc. sont utilisés. Les diagrammes de rayons X, IRM et tomographie assistée par ordinateur (CT) sont donnés ci-dessous.



**Figure I.13 : Image représentationnelle des rayons X, IRM et tomographie assistée Par ordinateur (CT)**

### 1.7.5 Les recherches en cours

Une vaste recherche est en cours dans la technique de traitement des images.

- 1) Imagerie du cancer - Différents outils tels que la TEP, l'IRM et la détection assistée par ordinateur aident à diagnostiquer et à prendre conscience de la tumeur.
- 2) Imagerie cérébrale - Se concentre sur le développement normal et anormal du cerveau, du vieillissement cérébral et des maladies courantes.
- 3) Traitement d'images - Cette recherche incorpore l'IRM structurelle et fonctionnelle en neurologie, l'analyse de la forme et de la structure osseuse, le développement d'outils d'imagerie fonctionnelle en oncologie et le développement de logiciels de traitement d'images TEP.
- 4) Technologie d'imagerie - Le développement de la technologie de l'image a obligé à déterminer si les nouvelles technologies sont efficaces et rentables. Cette technologie fonctionne dans les domaines suivants:

- Imagerie par résonance magnétique du genou
- Détection assistée par ordinateur en mammographie
- Échographie endoscopique dans la mise en scène du cancer de l'œsophage
- Imagerie par résonance magnétique dans les lombalgies
- Imagerie ophtalmique - Cela fonctionne en deux catégories:
  - 5) Développement de logiciels automatisés - Analyse les images rétinienne pour montrer les signes précoces de la rétinopathie diabétique
  - 6) Développement de l'instrumentation - Se concentre sur le développement de l'ophtalmoscope laser à balayage

### 1.7.6 Différents aspects du traitement des images

Les trois principales étapes de traitement d'une image sont [23]:

1. Le filtrage qui permet de nettoyer des images bruitées et/ou d'en isoler certains détails. Les outils principaux du filtrage sont : la convolution, la transformation de Fourier (filtres fréquentiels) et les ondelettes.
2. La segmentation permet de déterminer les contours et/ou les régions dans une image.
3. La restauration permet de retrouver une image à partir de données bruitées ou floues.

Le traitement d'image [19] recouvre tous les concepts scientifiques, les techniques et les moyens mis en œuvre pour permettre de tirer des conclusions sur un objet étudié à partir de la détection d'un signal bidimensionnel.

Dans le cas de l'imagerie médicale, il s'agit de formuler une ou plusieurs hypothèses diagnostiques à partir de la mesure de l'interaction d'un rayonnement avec un organisme vivant.

Très schématiquement, cela nécessite de procéder successivement à un échantillonnage du signal, puis à extraire de ce signal enregistré les composantes pertinentes pour l'objectif diagnostique final que l'on se donne. Il s'agit d'opérer un filtrage d'image.

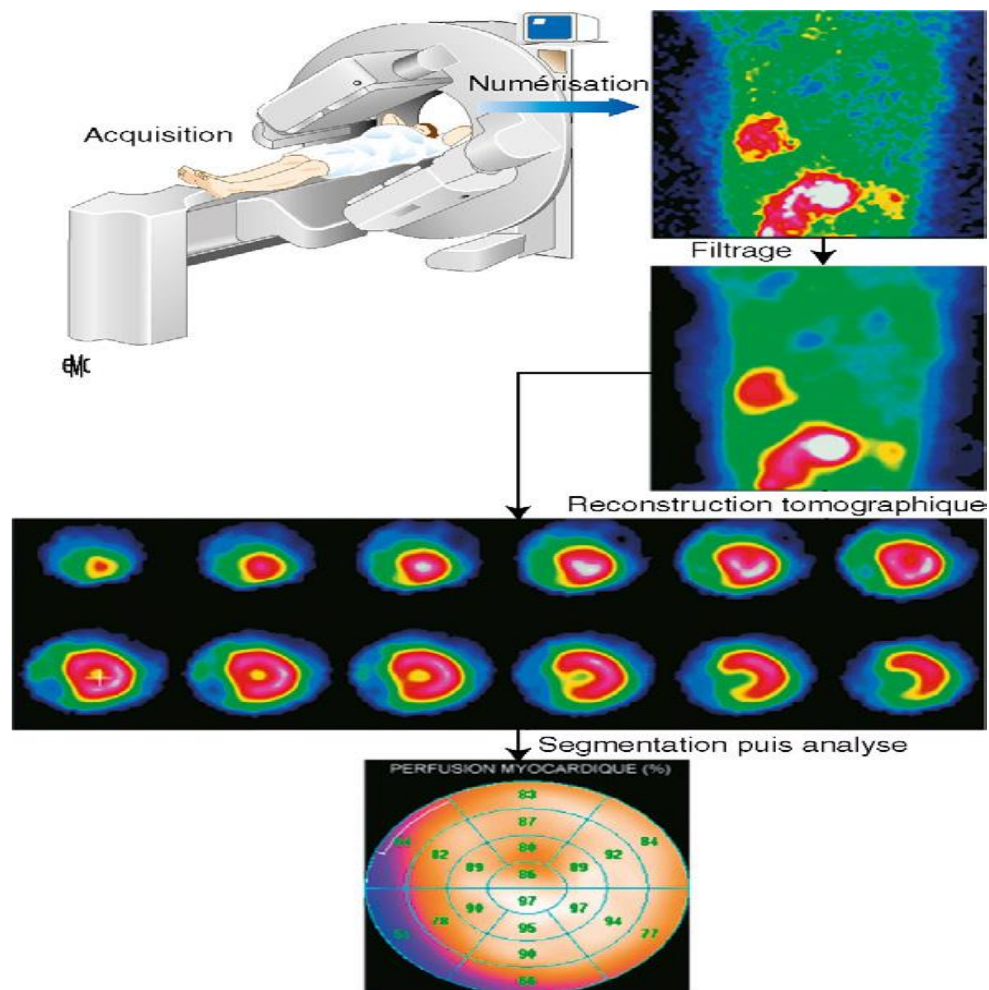
Le signal pertinent issu de cette opération de filtrage peut ensuite nécessiter une opération de reconstruction de manière à réorganiser les données enregistrées selon un mode plus approprié à une interprétation médicale. Enfin, lorsqu'une mesure quantitative doit être faite sur l'image produite, deux dernières étapes sont nécessaires. Il faut, d'une part, délimiter au sein de l'image reconstruite la région (surface ou volume) où la mesure doit être réalisée.

Cette étape porte le nom de segmentation.

D'autre part, il reste à procéder à une analyse d'image, c'est-à-dire à développer une méthode permettant de manipuler les signaux présents au sein de la zone sélectionnée pour évaluer une mesure significative d'un point de vue diagnostique.

Dans chaque domaine professionnel et dans le domaine médical en particulier, chacune de ces étapes doit être conçue en tenant compte de l'ensemble des connaissances spécifiques disponibles *a priori*. À l'issue de ce processus complet, un diagnostic médical peut être proposé sur une image optimisée à cette fin ou sur des mesures quantitatives réalisées sur l'image.

Ces étapes (numérisation, filtrage, reconstruction, segmentation et analyse) constituent les éléments de base du traitement de l'image.



**Figure I.14 :Représentation schématique des différentes étapes impliquées dans un processus complet de traitement d'image.**

### 1.7.7 But du traitement d'image

Le but du traitement d'image est divisé en 5 groupes[20]:

1. Visualisation : Observez les objets qui ne sont pas visibles.
2. Affinement de l'image et restauration : Pour créer une meilleure image.
3. Récupération d'image : Recherchez l'image qui vous intéresse.
4. Mesure du motif : Mesure divers objets dans une image.
5. Reconnaissance d'image : Distinguez les objets dans une image.

## 1.8 Détection d'anomalie avec traitement d'image

### 1.8.1 ANOMALIE

Pour définir les anomalies et les comprendre doit définir ce qu'est un comportement normal. Donc les comportements normaux sont définis comme des événements qui sont récurrents, qui se répètent de manière prévisible, ou fréquente [24].

Les anomalies sont définies comme les comportements non-normaux, inhabituels et non prévisibles, ou les événements s'écarter de la routine.

### 1.8.2 La détection d'anomalies

Les différents cas d'application et techniques de la détection d'anomalies sont de plus répertoriés dans plusieurs études [25].

Les travaux traitant de la détection d'anomalies regroupent sous une même appellation des techniques de détection de comportements anormaux et d'états erronés d'un système. Selon les domaines de recherche et les cas d'étude, les *anomalies* sont dénommées de différentes manières telles que données aberrantes, exceptions, attaques, erreurs ou violation de service. La détection d'anomalies se définit usuellement par l'action de discriminer dans un jeu de données composé d'*observations* (i.e. les lignes) et d'*attributs* (i.e. les colonnes) non connu à l'avance caractérisant un système cible, des observations qui ne correspondent pas à la tendance globale représentée par la majorité des observations . Ces méthodes se différencient notamment entre elles de par les données traitées pour détecter des anomalies ainsi que les algorithmes de détection utilisés.

### 1.8.3 Différents types de données

Les données à traiter pour la détection d'anomalies dans un système informatique peuvent être de trois grands types.

Premièrement, nous trouvons des journaux d'historique d'évènements (ou *logs*), systèmes ou applicatifs comme utilisés pour la détection d'anomalies des travaux. Ces journaux regroupent des évènements survenus respectivement au niveau des ressources d'un système ou correspondant aux opérations des tâches d'une application. Les logs systèmes fournissent une vue globale d'une machine. Leur analyse dépend du système d'exploitation de la machine. Les logs applicatifs incluent des évènements générés par l'exécution d'une application et dépendent le plus souvent de l'implémentation de celle-ci lorsque qu'elle n'est pas standardisée.

Deuxièmement, la détection peut également se faire à partir des traces d'audit qui regroupent des évènements de haut niveau correspondant à des échanges sur un réseau ou à des actions exécutées par des utilisateurs. Selon le niveau d'observation « paquets réseau » le format des traces d'audit est plus ou moins dépendant de l'environnement de déploiement. Les logs ainsi que les traces d'audits sont des données qu'il est nécessaire de décomposer avant leur utilisation pour la détection.



Troisièmement, certaines méthodes de détection traitent des statistiques d'utilisation d'une application ou bien d'observations de performances système. Les statistiques d'utilisation d'une application sont par exemple les indicateurs de performances système utilisées par cette application ou des compteurs (compteurs de ressources, d'erreurs, etc.). Ces données sont dépendantes de l'implémentation de l'application.

La détection profite donc sur une représentation plus complète d'un système mais elle est amenée à être plus longue à exécuter.

### 1.8.4 Techniques de détection basées sur l'apprentissage automatique

La détection d'anomalies se doit de discriminer une observation correspondant à une anomalie ou d'une observation de comportement normal.

Lorsque la détection d'anomalies est mise en œuvre par des techniques d'apprentissage automatique, des algorithmes sont utilisés afin de réaliser cette discrimination, sur la base de précédentes observations. Elle est appelée *prédiction*.

Un algorithme d'apprentissage automatique fait une prédiction quant à la catégorie d'une nouvelle observation, sur la base de précédentes observations déjà traitées.

La catégorie d'une observation, par exemple la catégorie de comportement normal et la catégorie anomalie, est assignée à une observation grâce à une étiquette, aussi appelée *label*. La prédiction faite par l'algorithme peut être confirmée ou infirmée a posteriori par un opérateur.

Afin de lever une ambiguïté éventuelle, nous précisons qu'il est commun de dire qu'un "algorithme prédit des labels" mais également qu'un "algorithme prédit des défaillances", dans le cas où celui-ci à une défaillance.

Dans le cas où un *modèle* peut être construit à partir d'observations labélisées dans le but de discriminer les observations de labels différents, on parle de *classification supervisée* de données.

Les différents labels indiquent alors des *classes* d'appartenance. Les observations labélisées permettant de construire le modèle sont appelées *données d'entraînement*.

Le modèle créé fait des prédictions quant à la classe de tout nouveau cas non connu (appelé *données de test*) et ne faisant donc pas partie du jeu de données d'entraînement. Une classification supervisée s'opère donc en deux phases que nous appelons *phase d'entraînement* et *phase de détection*.

Lorsqu'un modèle est construit pour discriminer des observations d'anomalie par rapport à des observations de comportement normal, on parle d'une *classification binaire*. Un classifieur peut également permettre de discriminer par exemple des anomalies de plusieurs types par rapport à un comportement normal ou plusieurs types de comportements normaux par rapport à une unique anomalie.

Dans le cas où aucune donnée d'entraînement labélisée n'est utilisée pour la construction d'un modèle, on parle de *classification non supervisée* de données.

### 1.8.5 Techniques de détection des anomalies appliquées à la sécurité maritime

Le problème de la surveillance des conteneurs maritimes est que de nombreux facteurs peuvent influencer les itinéraires empruntés par un porte-conteneurs.

Une modification de l'itinéraire peut être justifiée par les conditions économiques mondiales, par des facteurs culturels, par des facteurs politiques ou même par des facteurs

environnementaux. Par exemple, un itinéraire de navire est conditionné par un facteur politique comme un embargo, un navire américain ne peut pas entrer dans un port cubain, sinon il constitue une violation de l'embargo imposé par les États-Unis à Cuba. Ou un navire peut s'écarter de son itinéraire initial en raison des conditions météorologiques, telles que des ouragans, des icebergs, des marées ou des phénomènes naturels.

### 1.8.6 Les techniques de modélisation des comportements existantes

Les approches de détection d'anomalies comportementales utilisent des techniques de modélisation variées pour assurer la modélisation des comportements normaux qu'ils souhaitent observer. Ces techniques sont [25] :

**1- les réseaux de neurones :** ont particulièrement été exploités pour la modélisation de règles différenciant les flux réseaux normaux des flux divergeant. Utilise les réseaux de neurones afin d'identifier les connections anormales aux réseaux du SI. Après une période d'apprentissage contenant des connections identifiées comme normales et anormales, entraîne le réseau de neurones afin de classer de nouvelles connexions dans la classe appropriée. Le pouvoir de généralisation des réseaux de neurones permet de créer des règles de regroupement efficaces (dépendant du type de réseau de neurones utilisé). Cet outil est particulièrement adapté lorsque l'on possède des données étiquetées (normales et anormales). Malheureusement les règles générées ne sont pas interprétables par un humain et elles ne permettent pas la modélisation de séquences.

**2- Les techniques de DataMining (ECD Extraction de Connaissances à partir de Données) :** offrent également des propriétés intéressantes en termes de généralisation de règles. Ces techniques ont pour objectif d'extraire un savoir ou une connaissance d'une grande masse d'informations.

Les méthodes de DataMining permettent de déceler des relations entre des données (corrélation d'information) et de les regrouper suivant ces critères. De nombreuses méthodes sont utilisées pour permettre l'extraction de ces connaissances (Machines à Support de Vecteur, k-means, Local Outlier Factor, différentes méthodes de régression, K-NN). S'appuie sur une modélisation des processus (clusterisation) et de leurs appels systèmes correspondant.

Ces techniques de modélisation permettent la génération de règles génériques à l'aide de données d'apprentissage, elles présentent l'avantage d'avoir une meilleure lisibilité des règles générées et de déterminer les regroupements les plus fréquents.

Les techniques de DataMining ne permettent pas de représenter directement des séquences d'évènements mais plus des groupes d'évènements.

**3- Les automates d'états. :** utilisés afin de déterminer les séquences d'appels systèmes utilisées habituellement dans les programmes. Cette technique de modélisation lui permet de définir des séquences d'appels systèmes comprenant un état de départ, un état d'arrivée et des boucles possibles au sein de la séquence. L'utilisation d'automates d'états est bien adaptée pour la modélisation de séquences. Chaque nœud ne représentant qu'un seul et même état, les graphes générés pour la modélisation complète d'une séquence peuvent vite atteindre des tailles non lisibles par un analyste. De plus, les automates d'états ne permettent pas de déterminer les chemins les plus probables au sein du graphe généré.

**4- Les Réseaux de Pétri (RdP) :** permettent de modéliser des “séquences” d’actions ou/et d’états.

Ces RdP permettent de modéliser aussi bien des comportements de systèmes dynamiques que des évènements discrets.

Cette approche réalise une description des relations existantes entre des conditions et des évènements. Utilise les arbres de fautes et les réseaux de Pétri colorés pour créer, générer (arbres de fautes) et représenter (RdP) des scénarios d’attaques. Les réseaux de Pétri colorés permettent d’identifier l’ensemble des états et des actions nécessaires à la réalisation d’un scénario d’attaques.

**5- Les chaînes de Markov et les chaînes de Markov cachées :**

(Hidden Markov Model HMM) sont particulièrement adaptées pour modéliser les séquences de commandes utilisées par les utilisateurs. Dans cette approche, un graphe de chaîne de Markov est utilisé pour représenter chaque utilisateur.

A utilisé les chaînes de Markov dans une directive similaire pour modéliser des séquences d’appels système réalisées par les utilisateurs. Si un utilisateur effectue une succession d’appels système peu fréquente alors son activité sera considérée comme anormale.

Modélise les processus de traitement des paquets de flux TCP par des HMMs et utilise cette modélisation afin de classer les nouveaux flux TCP et de détecter d’éventuelles anomalies.

**6- les réseaux Bayésiens :**

Constituent la dernière classe de méthodes de modélisation de comportements normaux. Les réseaux Bayésiens sont des modèles probabilistes graphiques permettant d’acquérir, de capitaliser et d’exploiter des connaissances. S’appuyant sur la loi de Bayes, les réseaux Bayésiens permettent de calculer l’ensemble des probabilités conditionnelles des variables des nœuds qui le composent. Les réseaux Bayésiens permettent de modéliser parfaitement les séquences d’évènements ou des groupes de propriétés. Les réseaux Bayésiens modélisent les probabilités des différentes séquences d’évènements en se différenciant des chaînes de Markov par la représentation de plusieurs états d’une variable sur un même nœud du graphe.

## 1.9 Conclusion

Dans ce chapitre nous avons abordé les différents types de capteurs, leurs fonctionnements. Nous avons aussi discuté de l’aspect traitement d’image que planifions d’intégrer dans les capteurs précédemment abordé.

Dans le chapitre qui suivra, nous aborderons Réseaux de neurones et apprentissage profond qui sera utilisé dans le traitement d’image dans les capteurs.

## 2.1 réseaux de neurones

Les neurones sont considérés comme le support physique de l'intelligence. Les réseaux de neurones « artificiels », pour les différencier des réseaux de neurones biologiques, sont composés de neurones artificiels simples, petites fonctions mathématiques, qui permettent, de former des fonctions complexes très utiles.

### 2.1.1 Définition

On appelle réseaux de neurones un ensemble d'entités (les neurones) interconnectées. Dans la grande majorité des cas, les neurones sont en fait des fonctions calculées par un programme informatique, mais ils sont parfois réalisés sur des circuits électroniques.

Les neurones sont caractérisés par un état d'excitation qui dépend de celui des neurones situés en amont, ainsi que de la force des liens qui les relient.

De manière générale, un réseau de neurones comporte (figure 1) [26]:

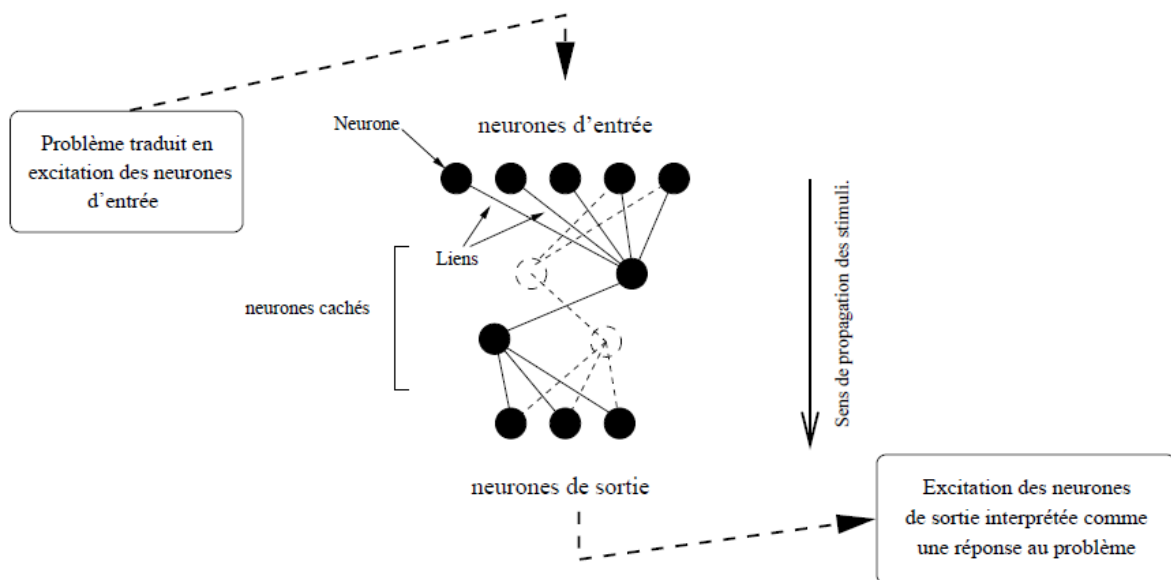


Figure II.1 : Schéma général d'un réseau de neurones.

1. des neurones d'entrée, auxquels on attribue une excitation en fonction des données que le réseau doit traiter.
2. d'autres neurones au travers desquels l'excitation des neurones d'entrée se propage et est modifiée.
3. des neurones de sortie dont l'état d'excitation fournit une réponse au problème posé en entrée.

## 2.2 Le neurone

Comme les réseaux de neurones mis au point par les informaticiens sont largement inspirés de ce que la biologie nous apprend sur ceux que l'on trouve chez les êtres vivants, il convient d'abord de décrire brièvement le modèle biologique.

### 2.2.1 Le modèle biologique

Chez les êtres vivants, les neurones sont les cellules nerveuses. Un neurone est doté de ramifications que l'on nomme les dendrites par lesquelles transite l'information venue de l'extérieur vers le corps cellulaire. Le neurone traite cette information et renvoie le résultat au travers de son axone. Ce signal émis par le neurone peut ensuite être transmis, au travers d'une synapse, à un autre neurone, ou encore à un muscle ou à une glande [27].

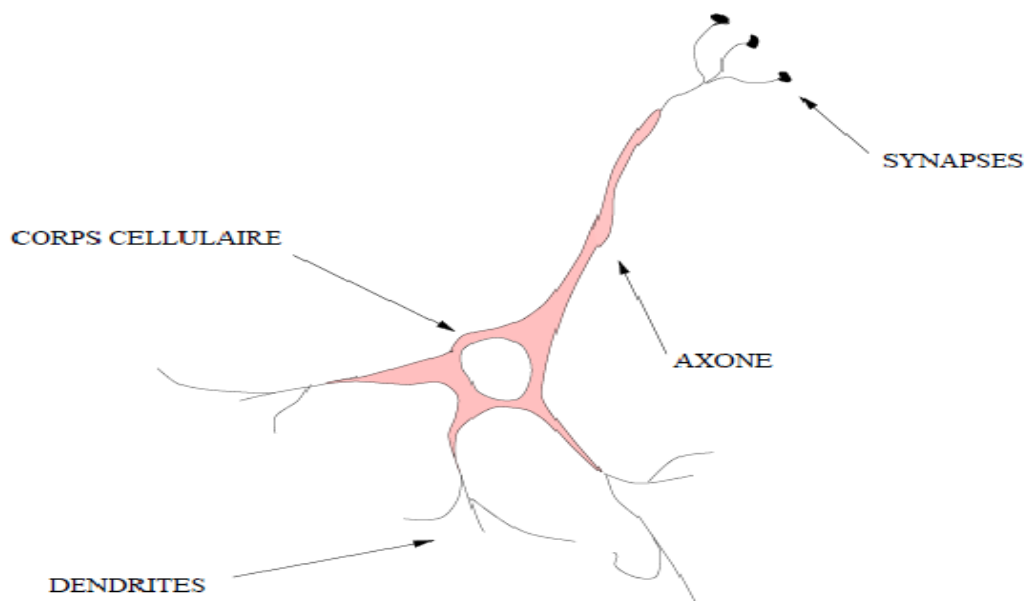


Figure II.2 : Un neurone biologique et ses principaux composants.

### 2.2.2 Vers une simulation du neurone biologique

Voici quelque résultante d'expériences sue les cellules nerveuses. Nous nous intéressons ici au comportement des neurones biologique, comportement dont sont inspirées les caractéristiques des neurones simules ou formels. Les quelques observation qui suivent vont nous aider à faire le parallèle entre les neurones biologiques et les neurones simules.

Lorsqu'on stimule un neurone (par un courant électrique par exemple), sa membrane cellulaire se dépolariselorsque la stimulation est suffisamment intense, la dépoliarisation est telle qu'une inversion de polarité brutale apparait et se propage le long de l'acône de manière unidirectionnelle, c'est le potentiel d'action.

Les synapses peuvent être inhibitrices : elles induisent alors une hyperpolarisation qui a tendance à empêcher la formation d'un potentiel d'action sur le neurone en aval. On comprend aisément leur rôle dans le contrôle d'antagonistes où, par exemple, les muscles extenseurs d'un membre ne doivent pas travailler en même temps que les muscles fléchisseurs.

Les synapses peuvent être excitatrices : elles induisent une dépoliarisation qui tend à générer un potentiel d'action sur les neurones en aval.

Voyons maintenant comment ces caractéristiques comportementales sont implémentées dans les neurones formels.

### 2.2.3 Le modèle formel

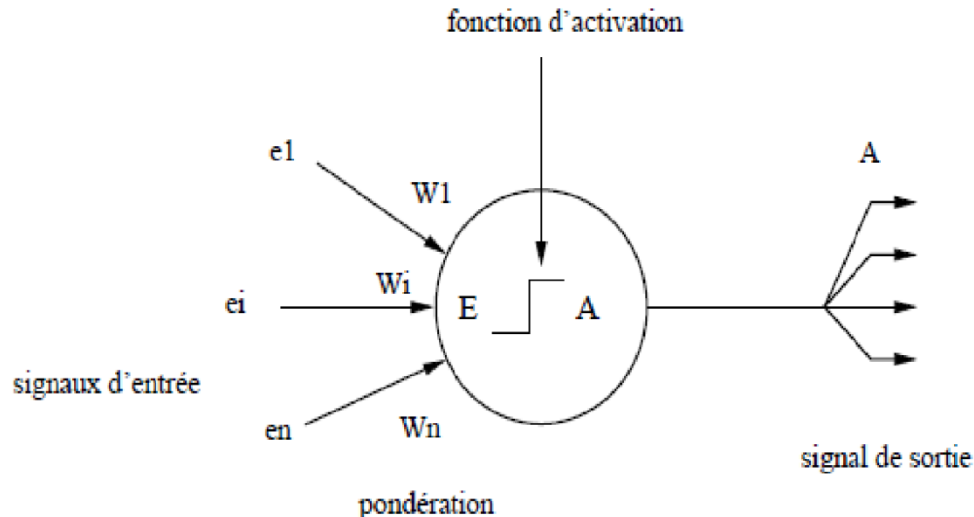


Figure II.3. Schéma d'un neurone formel.

Les neurones formels sont dotés de caractéristiques inspirées de celles des neurones biologique que nous avons passées en revue dans la section précédente :

**Le potentiel d'action des cellules nerveuses :** il s'agit ici (pour le neurone formel) d'une valeur numérique, qui peut être transmise à des neurones en aval. Un neurone formel, ne peut transmettre qu'une valeur **unique** qui correspond à son état d'activation.

**Les dendrites des neurones biologiques** : permettent de recevoir différents signaux de l'extérieur. De la même manière, un neurone formel recevoir des signaux  $e_i$  de plusieurs neurones. Ces signaux sont combinés en un signal d'entrée unique  $E$  :

$$E = \sum w_i \cdot e_i$$

Où les  $w_i$  sont les poids affectés aux signaux extérieurs.

**Les synapses** : les nombres  $w_i$  pondèrent les signaux émis par différents neurones situés en amont. On retrouve ici l'analogie des synapses qui, rappelons-le, peuvent être inhibitrices ( $w_i < 0$ ), ou excitatrices ( $w_i > 0$ ).

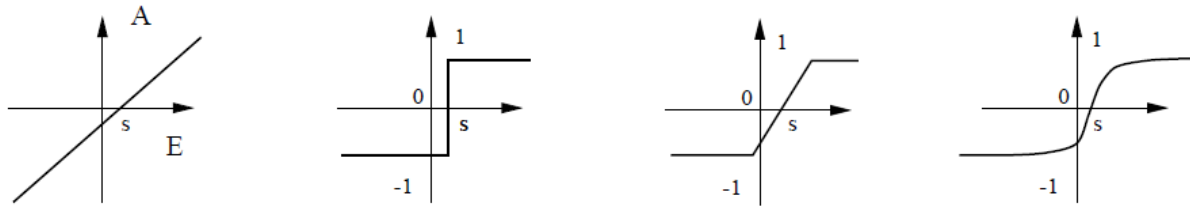
**Le seuil d'activation** : une fonction d'activation  $A$  gère l'état du neurone formel.

Génialement, si  $A \approx 1$  le neurone est excité et il est au repos si  $A \approx -1$  ou  $A \approx 0$  selon le cas. L'allure de cette fonction est généralement telle qu'il existe un seuil d'activation du neurone (fig. 4) : le neurone n'est excité que sil reçoit un signal d'entrée  $E$  supérieur à ce seuil  $s$ .

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumérées au tableau 1 [28].

Nom de la fonction	Relation d'entrée/sortie	Icone
Seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
Linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
Sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Compétitive	$a = 1$ si $n$ maximum $a = 0$ autrement	

**Tableau 2.1 : Quelques fonctions d'activation.**



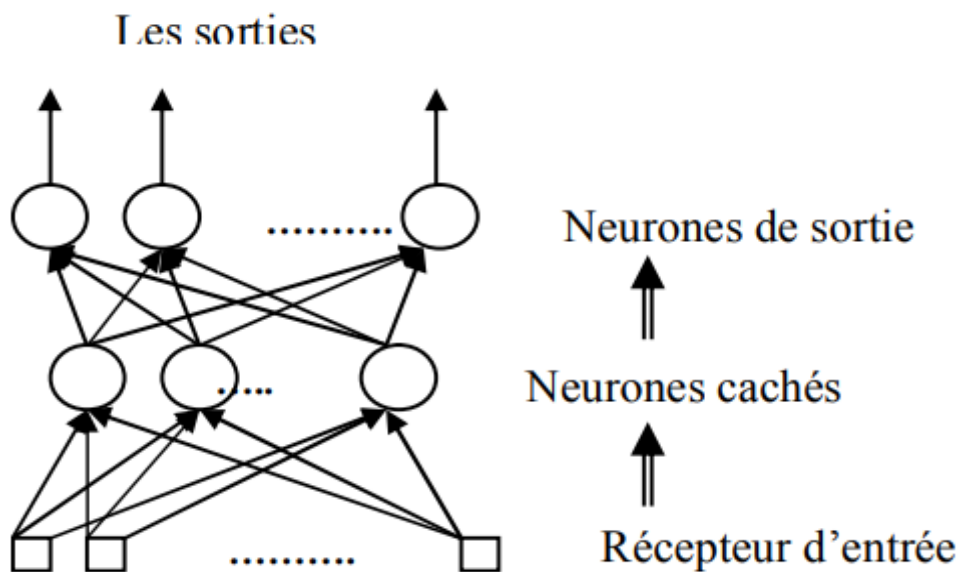
**Figure II.4 :Fonction de transfert : du neurone «linéaire» ; du neurone «seuil symétrique » ; du neurone «linéaire saturée symétrique» et du neurone «tangente hyperbolique» .**

### 2.3 Architecture des réseaux de neurones

Selon la topologie de connexion des neurones, on peut les classer en deux grandes catégories: réseaux non bouclés (statique ou feedforward) et réseaux bouclés (dynamique, feed back ou récurrent).

#### 2.3.1 Réseaux statiques "feed-forward "

Un réseau de neurones non bouclé (appelé aussi statique) est représenté comme un graphe dont les nœuds sont les neurones. L'information circule des entrées vers les sorties sans retour en arrière (Figure 4). Ce type de réseaux est utilisé pour effectuer des tâches d'approximation de fonction non linéaire, de la classification ou de la modélisation de processus statiques non linéaires [29].



**Figure: II.5 : Exemple d'un réseau de neurones non bouclé.**



### 2.3.2 Réseaux multicouches (ou Perceptron Multi Couche PMC)

C'est le réseau de neurones statique le plus utilisé. Les neurones sont arrangés par couche. Les neurones de la première couche reçoivent le vecteur d'entrée, ils calculent leurs sorties qui sont transmises aux neurones de la seconde couche qui calculent eux même leurs sorties et ainsi de suite de couche en couche jusqu'à celle de sortie. Chaque neurone dans la couche cachée est connecté à tous les neurones de la couche précédente et de la couche suivante, et il n'y a pas de connexions entre les cellules d'une même couche.

Il peut résoudre des problèmes non linéairement séparables et il suit un apprentissage supervisé avec la règle de correction de l'erreur.

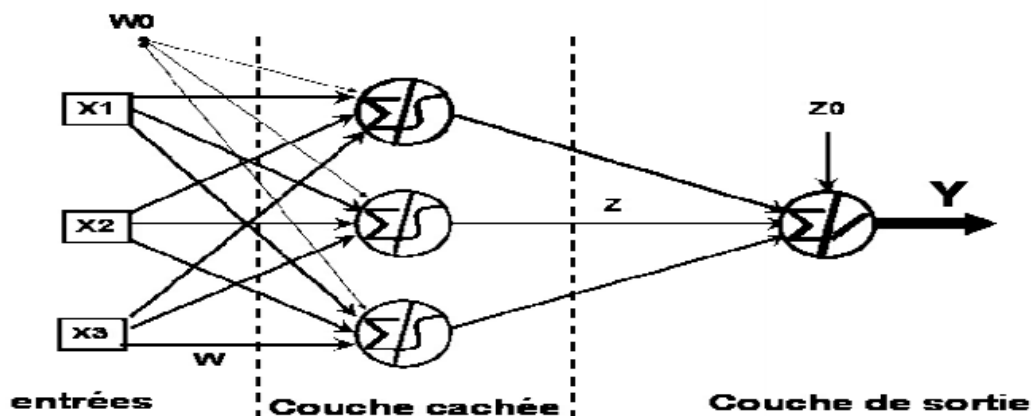


Figure II.6 : Perceptron Multi Couche PMC.

La sortie du réseau a pour expression :

$$Y = g[Z[f(WX + W_0)] + Z_0]$$

Avec :  $f$  et  $g$  les fonctions de transfert, des couches cachées et de sorties respectivement,  $Y$  sortie du réseau,  $X$  vecteur des entrées.  $W$  : Matrice des poids de connexions liant la couche d'entrée à la couche cachée.  $W_0$  : Vecteur des biais des cellules de la couche cachée.  $Z$  : Matrice des poids des connexions liant la couche cachée à la couche de sortie.  $Z_0$  : Vecteur des biais des cellules de la couche de sortie.

### 2.3.3 Réseau récurrents "Feed-back"

Un réseau bouclé (récurrent), régi par une ou plusieurs équations différentielles, résulte de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions[30].

Ces réseaux sont utilisés pour effectuer des tâches de modélisation des systèmes dynamiques, de commande de processus ou de filtrage. Le comportement dynamique d'un réseau de neurones bouclé peut être décrit par une représentation d'état paramétrée par les coefficients  $C$ , comme représentée ci-dessous :

$$\begin{cases} S(k+1) = \psi(S(k), I(k); C) \\ Y(k) = \xi(S(k), I(k); C) \end{cases}$$

Où  $I(k)$  est le vecteur des entrées externes,  $S(k)$  le vecteur des variables d'état,  $Y(k)$  le vecteur des sorties,  $\psi$  ( $\cdot, \cdot; C$ ) et  $\xi$  ( $\cdot, \cdot; C$ ) représentent les fonctions réalisées par le réseau de neurones statiques de la forme canonique interconnectés avec les coefficients  $C$ .

Nerand a démontré dans ses articles [12] [15] que : tout réseau de neurones bouclé aussi complexe soit-il peut être mis sous une forme canonique comportant un réseau de neurones non bouclé dont certaines sorties (les variables d'état) sont ramenées aux entrées par des bouclages de retard unité.

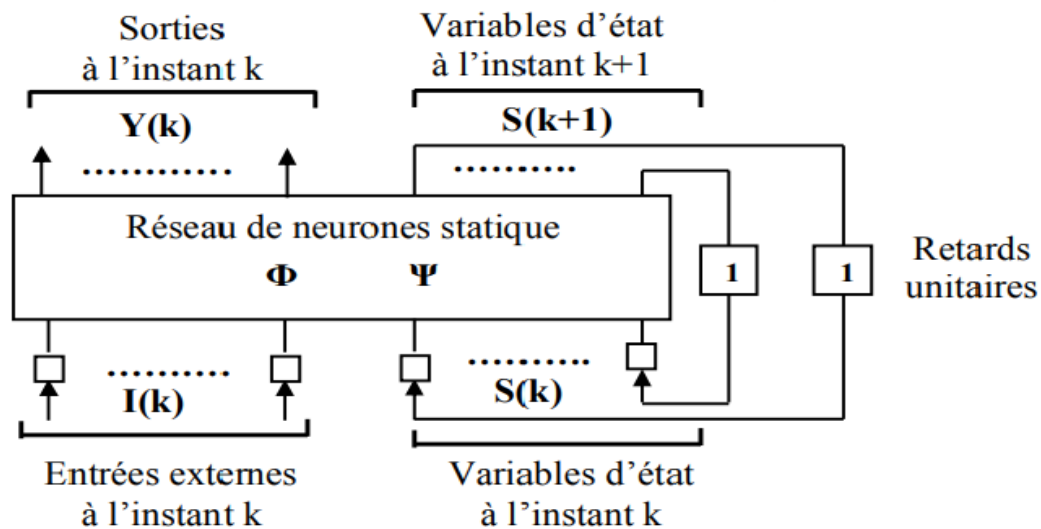


Figure II.7 : Forme canonique d'un réseau de neurones bouclé.

## 2.4 Apprentissage d'un réseau de neurone

L'apprentissage est une étape très importante du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié itérativement jusqu'à l'obtention du comportement désiré, et ce par l'ajustement des poids (connexion ou synapse) des neurones à une source d'information bien définie (Hebb 1949; Grossberg 1982; Rumelhart et al. 1986). L'apprentissage consiste également à extraire des conformités parmi les données utilisées pour l'entraînement du réseau mais l'objectif essentiel de l'apprentissage est la résolution du problème par la prévision, la classification ... etc (Weiss et Kulikowski 1991).

Pour un RNA, l'apprentissage peut être regardé également comme étant le processus de la mise à jour des poids (connexion) au sein du réseau dans le but d'ajuster la réponse du réseau à l'expérience et aux exemples. Il existe plusieurs types de règles d'apprentissage qui peuvent être rassemblées en deux catégories qui sont l'apprentissage supervisé et l'apprentissage non supervisé.

### 2.4.1 Apprentissage supervisé

Dans ce type d'apprentissage, on cherche à imposer au réseau un fonctionnement donné en forçant les sorties des réseaux à prendre des valeurs bien données (choisie par l'opérateur) et ce en modifiant les poids synaptiques[31].

Le réseau se comporte alors comme un filtre dont les paramètres de transfert sont ajustés à partir des couples entrée-sortie présentés (Hassoum 1995).

### 2.4.2 Apprentissage non supervisé

L'adaptation des paramètres du réseau s'effectue à partir d'un algorithme d'optimisation, l'initiation des poids synaptiques étant le plus souvent aléatoire. Des exemples d'utilisations de différentes règles d'apprentissage, architectures et algorithmes qui sont reliés à ce type supervisé sont illustrés dans le tableau 2..).

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Supervisé	Perception Simple ou Multi-Couches	Correction d'erreur	Perceptron, Rétro-Propagation, Adaline, Madaline	Classification, Approximation de fonctions, Prédiction, Contrôle
	Récurrente	Boltzmann	Apprentissage de Boltzmann	Classification
	Multi-Couches, non bouclés	Hebb	Analyse de discriminants linéaires	Analyse de données Classification
	à compétition	Par compétition	LVQ	Catégorisation au sein d'une classe, Compression de données
	ART			ARTMap

**Tableau 2.2 : Exemple de différents types d'apprentissages en mode supervisé**

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Non supervisé	Multi-Couches non bouclé	Correction d'erreur	Projection de Sam mon	Analyse de données
	Non bouclé ou à compétition	Hebb	Analyse en corn posantes	Analyse de données
	à compétition	Par compétition	VQ	Catégorisation, Compression de données
	Carte de Kohonen		SOM	Catégorisation, Analyse de données
	ART			ART-1, ART-2

**Tableau 2.3 : Exemple de différents types d'apprentissages en mode non supervisé**

## 2.5 Apprentissage Profond

### 2.5.1 Introduction :

L'apprentissage en profondeur est une forme d'apprentissage automatique qui permet des ordinateurs pour apprendre de l'expérience et comprendre le monde en termes de hiérarchie des concepts.

Un graphique de ces hiérarchies serait profond de plusieurs couches.

### 2.5.2 Deep Learning DNNs (Réseaux de neurones profonds):

C'est une technique d'apprentissage basée sur les réseaux de neurones artificiels, permettant à un programme, par exemple, de reconnaître le contenu d'une image ou de comprendre le langage parlé.

Il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de deep learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques à travers chaque couche avec une intervention humaine. Les algorithmes permettent de propager les données de la couche d'entrée vers la couche de sortie[32].

Deux des plus éminents inventeurs et promoteurs du deep learning (DL) ont été recrutés par Facebook (Yann LeCun, université de New York) et Google (Geoffrey Hinton, université de Toronto)

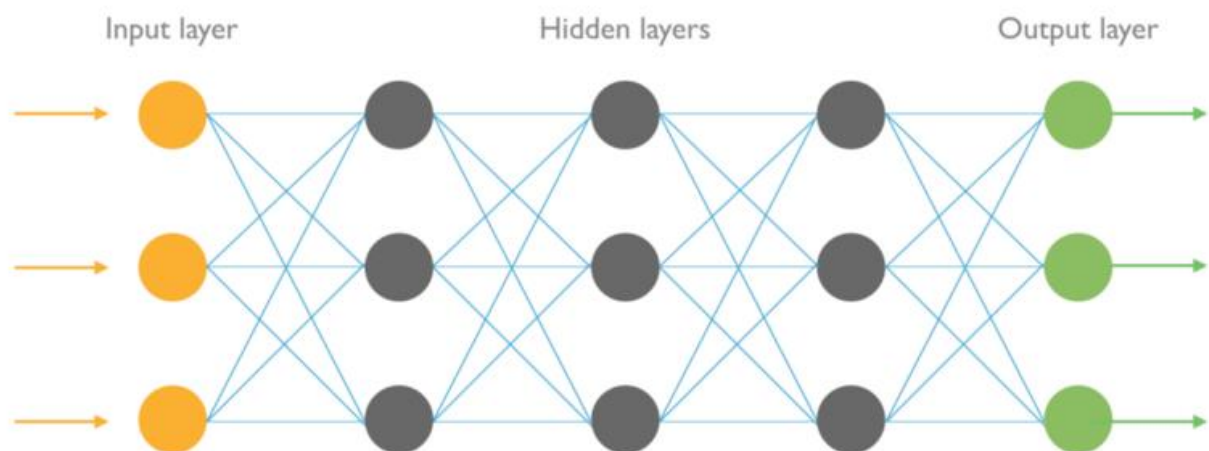


Figure II.8 : schéma d'un réseau de neurone profond

### 2.5.3 Fonctionnement du deep learning :

Les neurones sont regroupés en trois types de couches différents:

1. Couche d'entrée : reçoit des données d'entrée et transmet les entées a la première couche masqué.
2. Couches cachées : effectuent des calculs mathématiques sur nos entrées. L'un des défis de la création de réseaux neuronaux est de déterminer le nombre de couches cachées, ainsi que le nombre de neurones pour chaque couche.
3. Couche de sortie : renvoie les données de sortie.

Chaque connexion entre neurones est associée à un **poids**. Ce poids détermine l'importance de la valeur d'entrée. Les poids initiaux sont définis aléatoirement.

Chaque neurone a une fonction d'activation . Ces fonctions sont difficiles à comprendre sans raisonnement mathématique.

En termes simples, l'un de ses objectifs est de «normaliser» la sortie du neurone. Une fois qu'un ensemble de données d'entrée est passé à travers toutes les couches du réseau neuronal, il renvoie les données de sortie via la couche de sortie.[33]

Pour former un réseau neuronal il faut :

- Un grand ensemble de données.
- Une grande quantité de puissance de calcul.

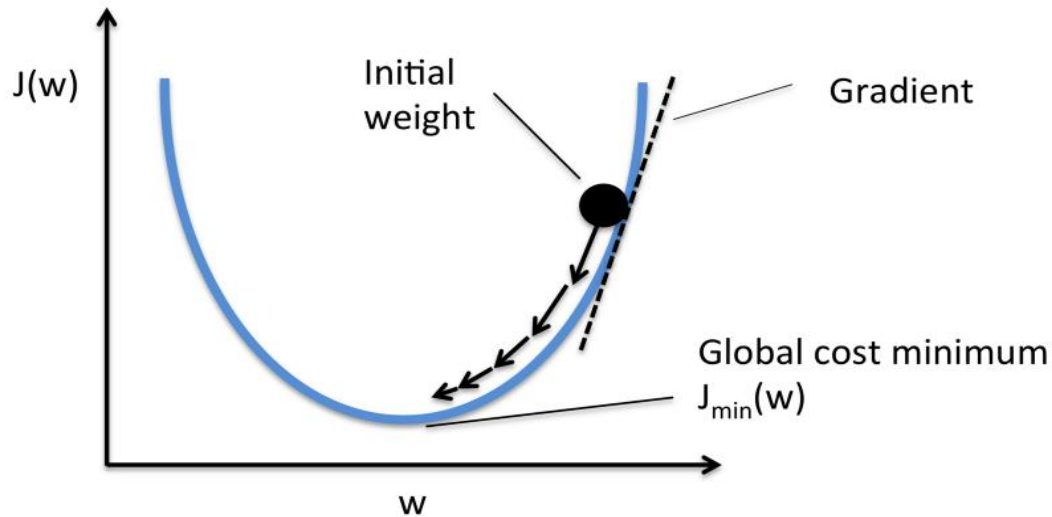
Pour former l'intelligence artificiel « IA », on lui donne les informations de l'ensemble de données et comparer ses résultats avec les résultats de l'ensemble de données. Puisque l'IA n'est toujours pas entraînée, ses sorties seront fausses.

Une fois que l'ensemble des données est parcouru, on peut créer une fonction qui montre à quel point les sorties de l'IA étaient erronées par rapport aux sorties réelles. Cette fonction s'appelle la fonction de coût . Elle que cette fonction soit nulle. C'est à ce moment que les sorties sont identiques aux sorties du jeu de données.

#### ❖ La réduction de la fonction du cout :

Modifier les poids entre les neurones et les changer au hasard jusqu'à ce que la fonction de coût soit faible, mais ce n'est pas très efficace.

Ou bien, utiliser une technique appelée Gradient Descent (qui est une technique qui permet de trouver le minimum d'une fonction). Cela fonctionne en **changeant les poids** par petits incréments **après chaque itération de jeu de données**. En calculant la dérivée (ou le gradient) de la fonction de coût pour un certain ensemble de poids, et voir dans quelle direction le minimum est.



**Figure II.9 : Gradient Descent.**

Pour minimiser la fonction de coût, il faut parcourir plusieurs fois le jeu de données. C'est pourquoi on a besoin d'une grande puissance de calcul. La mise à jour des poids en utilisant la descente de gradient est effectuée **automatiquement**.

#### 2.5.4 Applications du Deep Learning :

- ✓ la classification des images
- ✓ la reconnaissance des actions humaines
- ✓ la reconnaissance des gestes de la main
- ✓ l'analyse des scènes

#### 2.5.5 Domaine du système deep Learning

il existe plusieurs domaines du système du deep learning [34] :

- ✓ les télécommunications,
- ✓ les finances,
- ✓ la sécurité,
- ✓ les réseaux sociaux,
- ✓ les assurances et l'industrie automobile.

## 2.5.6 Thèmes récurrents de l'apprentissage en profondeur

### Programmation dynamique pour apprentissage supervisé / renforcement (SL / RL)

Un thème récurrent de Deep Learning est la programmation dynamique (DP) (Bellman, 1957), qui peut aider à faciliter l'attribution de crédits selon certaines hypothèses.

Par exemple, dans les NN de SL, la propagande elle-même peut être considérée comme une méthode dérivée de DP. Dans les RL traditionnelles basées sur de fortes hypothèses markoviennes, les méthodes dérivées du DP peuvent aider à réduire considérablement la profondeur du problème (Sec. Les algorithmes DP sont également essentiels pour les systèmes qui combinent des concepts de NN et des modèles graphiques, tels que les modèles de Markov cachés (HMMS) et la maximisation des attentes[35].

### Apprentissage non supervisé (UL) Facilitant SL et RL

Un autre thème d'enregistrement est la manière dont UL peut faciliter à la fois SL est normalement utilisé pour encoder des données entrantes brutes telles que des flux vidéo ou vocaux sous une forme plus pratique pour un apprentissage ciblé conséquent. En particulier, les codes qui décrivent les données d'origine dans une manière moins redondante ou plus compacte peut être introduite dans les machines SL ou RL quels espaces de recherche peuvent donc devenir plus petits que ceux nécessaires pour traiter les données brutes. UL est étroitement lié aux thèmes de la régularisation et de la compression.

### Apprentissage des représentations hiérarchiques par SL profond, UL, RL :

De nombreuses méthodes d'intelligence artificielle à l'ancienne (GOFAI) (Nilsson, 1980) ainsi que des approches plus récentes de l'IA et de l'apprentissage automatique apprennent des hiérarchies de représentations de données de plus en plus abstraites. .

Par exemple, certaines méthodes de reconnaissance de formes syntaxiques telles que l'induction grammaticale découvrent des hiérarchies de règles formelles modéliser des observations.

Le mathématicien automatisé / EURISKO partiellement supervisé apprend continuellement des concepts en combinant des concepts préalablement appris. Un tel apprentissage de la représentation hiérarchique est également un thème récurrent de NN DL pour SL et RL hiérarchique

Souvent, les représentations hiérarchiques abstraites sont des sous-produits naturels de la compression de données.

### Occam's Razor: Compression et Minimum Description Length (MDL)

Le rasoir d'Occam privilégie les solutions simples par rapport aux solutions complexes. Étant donné un langage de programmation.

Le principe de la description minimale (MDL) peut être utilisé pour mesurer la complexité d'une solution candidat par la longueur du programme le plus court qui le calcule (par exemple, Solomonoff, 1964;

Kolmogorov, 1965b; Chaitin, 1966; Wallace et Boulton, 1968; Levin, 1973a; Solomonoff, 1978).

Quelques méthodes tenir compte explicitement de l'exécution du programme (Allender, 1992);

Beaucoup ne considèrent que les programmes à exécution constante, écrits en programmation non universelle langues (par exemple, Rissanen, 1986; Hinton et van Camp, 1993). Dans le cas NN, le principe MDL suggère que la faible complexité en poids de NN correspond à une probabilité élevée de

NN chez les Bayésiens vue (E., MacKay, 1992; Buntine et Weigend, 1991) la performance de généralisation (par exemple, Baum et Haussler, 1989), sans dépasser les données de formation. De nombreuses méthodes ont été proposées pour régulariser les NN, c'est-à-dire rechercher des solutions informatiques. mais simple, faible complexité SLNN (Sec. 5.6.3) et RL NNs (Sec. Ceci est étroitement lié à certaines méthodes UL.

### **Unités de traitement graphique (GPU) rapides pour DL en NN**

Alors que le millénaire précédent a connu plusieurs tentatives de création de matériel rapide spécifique à NN (par exemple, Jackel et al, 1990) et à l'exploitation de matériel standard (par exemple, Anguita et Gomes, 1996), le nouveau millénaire a acquis une percée en matière de DL sous forme de multiprocesseurs bon marché cartes graphiques ou GPU.

Les GPU sont largement utilisés pour les jeux vidéo, un énorme et compétitif marché qui a fait baisser les prix du matériel. Les GPU excellent à la matrice rapide et aux multiplications vectorielles nécessaire non seulement pour des mondes virtuels convaincants, mais aussi pour la formation NN, où ils peuvent accélérer apprendre par un facteur de 50 et plus. Une partie de l'implémentation FNN basée sur le GPU ont largement contribué aux récents succès dans les concours de reconnaissance de formes , image. Segmentation et détection d'objet.

## **2.6 Conclusion**

Dans ce chapitre nous avons parlé globalement des réseaux x de neurones et réseaux de neurones profond afin de les utilisé dans le chapitre suivant.



# Chapitre 3

## 3.1 Introduction

Après avoir étudié dans la première partie l'état de l'art sur les capteurs, le traitement d'image et les réseaux de neurone et le du Deep Learning, nous allons réaliser notre classificateur et créer notre modèle, pour appliquer par la suite ce model sur la base d'image qu'on a choisi. Et afin de réaliser tout ça nous utiliserons le langage python et les différentes bibliothèques comme Tensorflow et Keras, pour l'apprentissage et la classification.

## 3.2 conception

Pour commencer nous avons utilisé deux tâches : l'entraînement et le test.

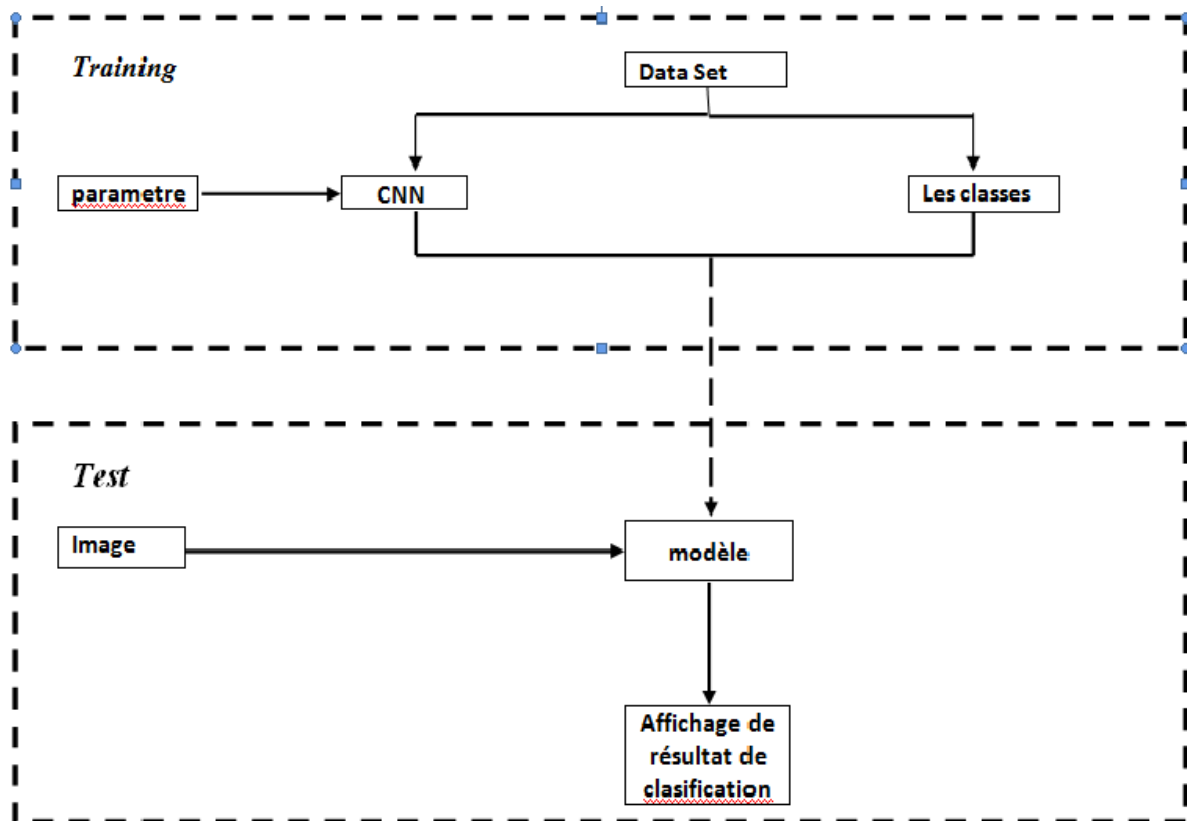


Figure III.1: Conception du classificateur

### A. L'entraînement :

- ❖ Pour l'entraînement qu'on va créer notre modèle grâce à des différentes configurations, comme la base de données des images « Dataset » et l'algorithme qui va nous permettre de créer notre réseau de neurone.
- ❖ Les classes : c'est un fichier texte qui porte les noms des classes de notre DataSet qui sont : poumons normaux et poumons malades.
- ❖ Après l'exécution des Dataset sur notre algorithme qui sera paramétrée pour générer un modèle, qui sera à sa part utilisé pour faire le test.

### B. Le test :

Dans cette tâche on a besoin de deux choses importantes :

- L'image : on a besoin d'une ou plusieurs dans notre test comme entrées.
- Le modèle : c'est un fichier généré dans notre training.
- L'affichage de la classification : on va afficher le résultat sortie du modèle.

## 3.3 Présentation des outils

### 3.3.1 Le software :

#### Python :

Apparu en 1990 par Guido van Rossum, développé par Python Software Fondation.

C'est un langage de programmation objet et multiplateforme. Il favorise la programmation impérative et orienté objet, il est doté d'un typage dynamique fort' une gestion automatique de la mémoire, il fonctionne sur la plupart des plateformes informatiques. Il est conçu pour optimiser la productivité de programme en offrant des outils du haut niveau et une syntaxe plus simple a utilisée. [36]



Plusieurs Framework open sources sont disponibles dans la littérature, la grande majorité supporte le langage Python. Voici un exemple :

#### TensorFlow :

TensorFlow crée par l'équipe Google Brain pour mener des recherches sur le ML et le Deep Learning. Il est considéré comme une version moderne de Theano.

C'est une bibliothèque de logiciels open source pour le calcul numérique à l'aide de graphes de flux de données. Les nœuds du graphe représentent des opérations mathématiques, tandis que les arêtes du graphe représentent les tableaux de données multidimensionnels (tenseurs) communiqués entre eux.

- Avantages

- ✓ Python
- ✓ supporté par Google
- ✓ Une très grande communauté
- ✓ Le support du multi-GPU

- Inconvénients

- ✓ Plus lent que les autres framework dans de nombreux benchmarks, bien que Tensorflow se rattrape.
- ✓ Le soutien des RNN est encore surclassé par Theano [37]

**Keras :**

Le framework le plus haut niveau, le plus convivial de la liste. Il permet aux utilisateurs de choisir si les modèles qu'ils construisent sont exécutés sur TensorFlow ou un autre framework. Il est écrit et entretenu par Francis Chollet, un autre membre de l'équipe Google Brain.

## Avantages

- Python
- Le backend par excellence pour TensorFlow
- Interface haut niveau, intuitive

- Inconvénients

- Moins flexible que les autres API

**La base de données Chest-xray-pneumonia :**

La base de données « chest-xray-pneumonia » est un ensemble des images de poumon qui sont divisées en deux parties : normal et anormal (malade).

Elle se compose de 60000 images en mode gris 64x64.

On trouve 5218 images pour l'entraînement tel qu'on a 1324 images de poumons normaux.

Et 3876 images de poumons malades.

Et pour le test y'on a 624 images : 234 des poumons normaux, 390 anormaux (malades).

Voici des figures qui montrent les deux cas qu'on a utilisés :



Figure III.2 : des poumons normaux

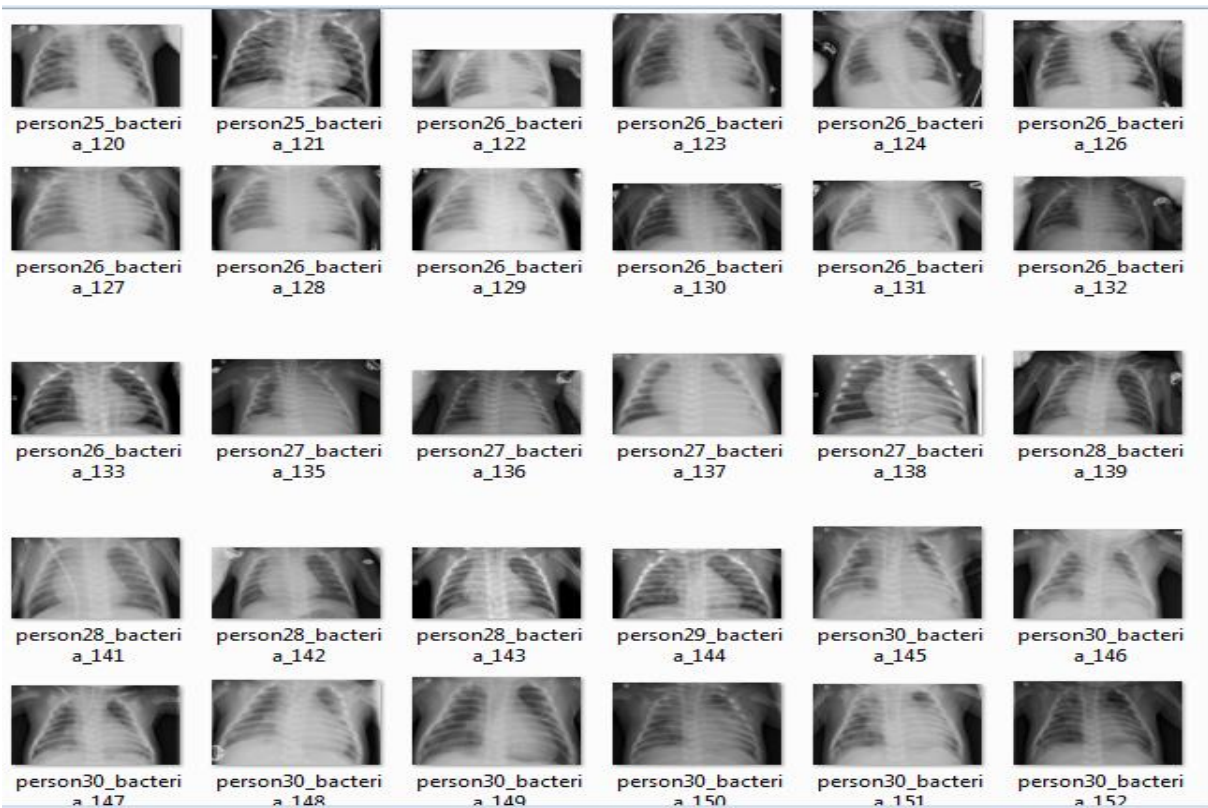


Figure III.3 : des poumons malades

### 3.3.2 Le hardware

Le Deep Learning est un domaine avec des exigences en calculs intenses et la disponibilité des ressources (surtout en GPU) dédiés à cette tâche vont fondamentalement influencer sur l'expérience de l'utilisateur car sans ses ressources, il faudra trop de temps pour apprendre de ses erreurs ce qui peut être décourageant. Les expérimentations ont tous été effectuées sur une machine qui offre des performances acceptables dont voici les caractéristiques :

- Un PC portable lenovo i7 CPU 2.40 GHZ
- RAM de taille 8 GO
- Disque dur de taille 1 TO.
- Système d'exploitation 64 bits Windows 7

### 3.3.3 L'implémentation

Dans ce qui suit, nous donnerons le code source de notre implémentation:

```
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
import matplotlib.pyplot as plt

# Initialising the CNN
classifier = Sequential()
# Step 1 - Convolution
classifier.add(Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation = 'relu'))
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Adding a second convolutional layer
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Step 3 - Flattening
classifier.add(Flatten())
# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
# Part 2 - Fitting the CNN to the images
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range =
0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```

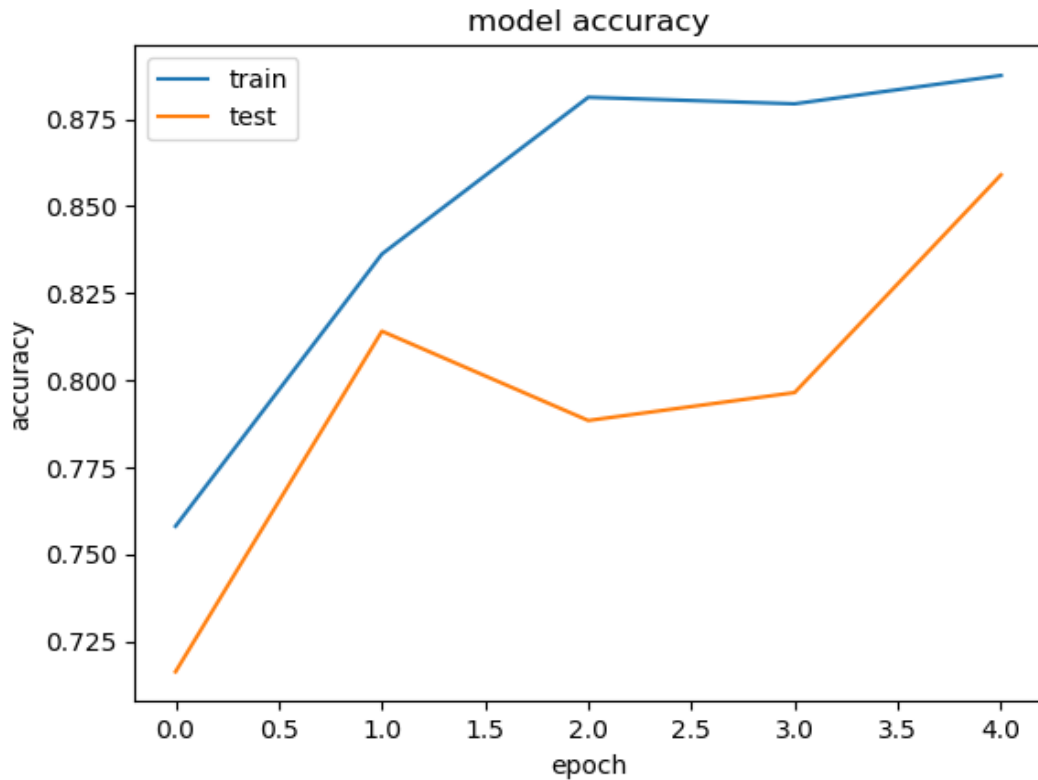
training_set = train_datagen.flow_from_directory('/media/a/NOUVEAU
NOM/Nouveaudossier/train',target_size =(64, 64),batch_size=32,class_mode='binary')
test_set = test_datagen.flow_from_directory('/media/a/NOUVEAU
NOM/Nouveaudossier/test',target_size = (64, 64),batch_size = 32,class_mode = 'binary')
history = classifier.fit_generator(training_set,steps_per_epoch = 50,epochs =
5,validation_data = test_set,validation_steps = 50)
classifier.save('pneumonia.h5')
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss m0h')
plt.ylabel('loss e5i')
plt.xlabel('epoch d5.')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('/media/a/NOUVEAU NOM/Nouveaudossier/image',
target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = "NORMALE"
else:
    prediction = "PNEUMONIA"
print(prediction)

```

### 3.3.4 Validation

Après nos expérimentations, nous avons créé un modèle à l'aide des DataSet, et par la suite on a appliqué sur ce modèle la base d'images chest-xray-pneumonia. Après la création du model, on va lui faire un test, et on obtient les deux graphes suivants :

- ✓ Graphe de précision  
d'après ce graphe on remarque que la précision training et du test augment avec le nombre d'époques:



**Figure III.4 : graphe de précision pour le modèle.**

✓ Graphe d'erreur

Ce graphe montre que l'erreur du training et du test diminue jusqu'à la première époque, et se stabilisent de 1 ères au 2 Emme époque, et apres la 2 époque,le test s'augmente et le training se diminue.

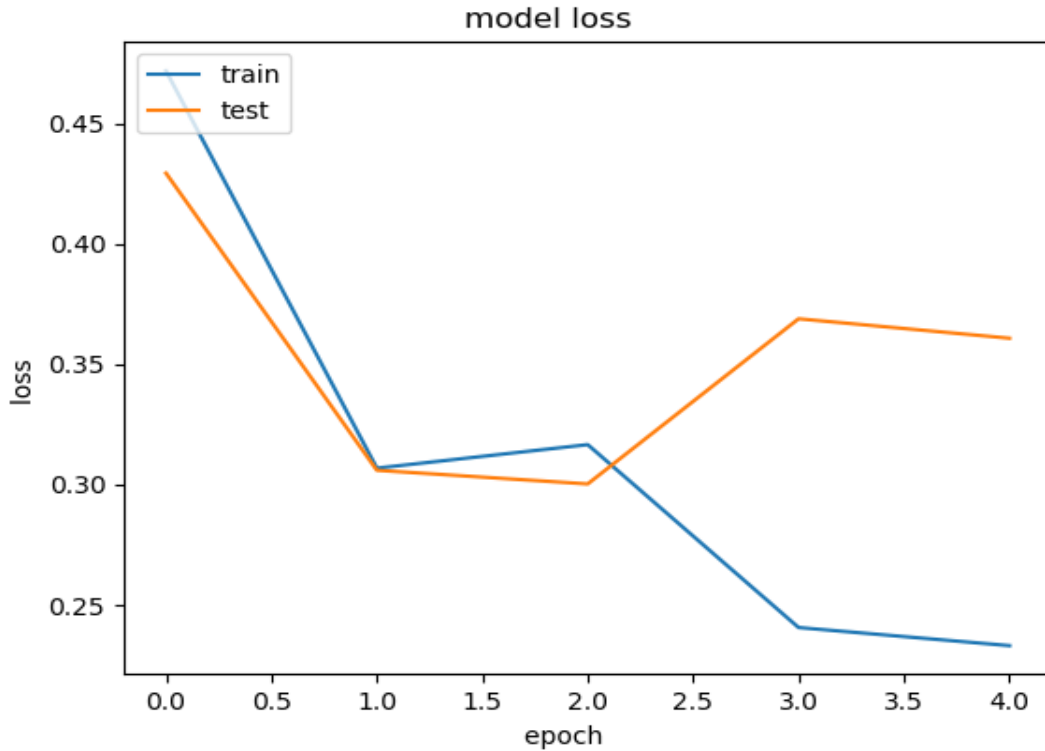


Figure III.5 : graphe d'erreurs pour le modèle.

Et après la classification on obtient le pourcentage de précision et d'erreurs, et la classe ou il appartient l'image donnée :

```

22/50 ..... ETA: 27s - loss: 0.2336 - acc: 0.9348
23/50 ..... ETA: 24s - loss: 0.2332 - acc: 0.9322
24/50 ..... ETA: 24s - loss: 0.2334 - acc: 0.9310
25/50 ..... ETA: 25s - loss: 0.2336 - acc: 0.9325
26/50 ..... ETA: 25s - loss: 0.2340 - acc: 0.9330
27/50 ..... ETA: 24s - loss: 0.2317 - acc: 0.9339
28/50 ..... ETA: 25s - loss: 0.2314 - acc: 0.9329
29/50 ..... ETA: 23s - loss: 0.2330 - acc: 0.9337
30/50 ..... ETA: 23s - loss: 0.2330 - acc: 0.9340
31/50 ..... ETA: 21s - loss: 0.2349 - acc: 0.9332
32/50 ..... ETA: 21s - loss: 0.2377 - acc: 0.9334
33/50 ..... ETA: 20s - loss: 0.2391 - acc: 0.9338
34/50 ..... ETA: 20s - loss: 0.2399 - acc: 0.9339
35/50 ..... ETA: 19s - loss: 0.2357 - acc: 0.9351
36/50 ..... ETA: 19s - loss: 0.2479 - acc: 0.9351
37/50 ..... ETA: 18s - loss: 0.2440 - acc: 0.9375
38/50 ..... ETA: 17s - loss: 0.2439 - acc: 0.9359
39/50 ..... ETA: 17s - loss: 0.2393 - acc: 0.9359
40/50 ..... ETA: 16s - loss: 0.2405 - acc: 0.9351
41/50 ..... ETA: 16s - loss: 0.2370 - acc: 0.9379
42/50 ..... ETA: 15s - loss: 0.2365 - acc: 0.9355
43/50 ..... ETA: 14s - loss: 0.2367 - acc: 0.9353
44/50 ..... ETA: 14s - loss: 0.2363 - acc: 0.9377
45/50 ..... ETA: 13s - loss: 0.2363 - acc: 0.9353
46/50 ..... ETA: 12s - loss: 0.2412 - acc: 0.9351
47/50 ..... ETA: 12s - loss: 0.2417 - acc: 0.9355
48/50 ..... ETA: 11s - loss: 0.2343 - acc: 0.9353
49/50 ..... ETA: 10s - loss: 0.2347 - acc: 0.9359
50/50 ..... 46s 79ms/step - loss: 0.2425 - acc: 0.9337 - val loss: 0.3471 - val acc: 0.8413 38/50
INFO:tensorflow:
>>>
    
```

Figure III.6 : résultat de la classification.



### **3.4 Conclusion**

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones convolutionnels, pour cela on a créé un modèle et nous lui avons fait des tests afin d'obtenir des résultats en termes de précision et d'erreur. L'implémentation que nous avons proposé permet de détecter les images anormales des images normales avec un taux de précisions élevé et une erreur faible.

# Conclusion générale

Le travail que nous avons réalisé dans le cadre de ce mémoire porte essentiellement sur la détection des maladies avec le traitement d'image. et on a discuté des notions fondamentales de l'apprentissage les plus populaires, des réseaux de neurones en général et des réseaux de neurones convolutionnels en particulier. Nous avons introduit ces réseaux de neurones convolutionnels en présentant les différents types de couches utilisées dans la classification.

On a implémenté le réseau neurones convolutifs et on a fait un test et afficher à la fin les résultats.

L'implémentation a été faite avec le langage de programmation python et on a utilisé des bibliothèques pour faciliter la tâche de création de nos modèles et pour l'accélération du training.

Finalement, pour avoir des meilleurs résultats, on peut améliorer ce travail on ajoutant une qui permet d'exécuter en temps réel.

# Bibliographie

- [1] : Conception et optimisation de capteurs à courants de Foucault pour la détection de défauts profonds dans des matériaux amagnétiques, Ecole Centrale de Lyon, 1995, Frédéric Thollon.
- [2] : Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie // Thèse présentée pour l'obtention du titre de Docteur de l'Université Henri Poincaré, Nancy 1 en Sciences, spécialité Automatique, Traitement du Signal et Génie Informatique par Cristian Duran-Faundez .
- [3] : Architectures et Protocoles des Réseaux Claude Duvallat Université du Havre UFR Sciences et Techniques.
- [4] : Quelques contributions dans les réseaux de capteurs sans fil , l'Université d'Avignon , Clément SAAD
- [5] : Modélisations et Analyses de Réseaux de Capteurs, Ludovic SAMPER le 7 avril 2008, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE.
- [6] : H. Karl, A. Willig. Protocols and architectures for wireless sensor networks. Wiley, 2005.
- [7] : E.H. Callaway. Wireless sensor networks: architectures and protocols. CRC Press, 2004.
- [8] : N. Badache, *La mobilité dans les systèmes répartis*, Techniques et sciences informatiques, Volume 17-n°8, pp 969-997, 1998.
- [9] : Les réseaux de capteurs : état de l'art Lyes KHELLADI & Nadjib BADACHE, Février 2004.
- [10] : étude de standard IEEE 802.11 dans le cadre des réseaux AdHoc : de la simulation à l'expérimentation, L'institut national des sciences appliquées de LYON, Dominique Dhoutaut.
- [11] : Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie Cristian Duran-Faundez Université Henri Poincaré - Nancy I, 2009.
- [12] : Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie Cristian Duran-Faundez ? Nancy I, 2009.
- [13] : Contributions to the energy optimization for security, localization and routing in wireless sensor networks université Abdelmalek Essaâdi; Faculté des Sciences et Technique de Tanger, 2013. English.
- [14] : Conception et validation de plateformes de communication autour du corps humain, à l'échelle de l'individu et du groupe Réseaux et télécommunications [cs.NI]. INSA de Lyon, 2015. Français
- [15] : LES CAPTEURS CCD/CMOS, Jérôme SIX Léo MEIGNAN
- [16] : image processing and analysis-variational , PDE, wavelet, and stochastic methods/ TONY F. CHAN JIANHONG (JACKIE) SHEN.

[17] :Digital Image Processing EE368/CS232/ Bernd Girod, Gordon Wetzstein Department of Electrical Engineering Stanford University

[18] :Détection automatique de phénomènes présents dans une image satellitaire en composition colorée par abdelazizammadi faculte des sciences rabat universitemohammed 5 - master en informatique telecom imagerie 2006

[19] : Traitement de l'image De la numérisation à l'archivage et la communication ; Coordonné par Olivier Hélénon/Joël Chabriaux/Bernard Gibaud/Denis Mariano-Goulart

[20] :<https://www.engineersgarage.com/articles/image-processing-tutorial-applications>

[21] :Une méthode de développement d'applications de traitement d'images.  
Régis Clouard

[22] :Digital Image Processing.  
Bernd Girod, Gordon Wetzstein! Department of Electrical Engineering! Stanford University 2013/2015.

[23] :Introduction au traitement mathématique des images-méthodes déterministes Directeurs de la collection: J. Garnier et V. Perrier MATHÉMATIQUES & APPLICATIONS Comité de Lecture 2012–2015/Editorial Board 2012–2015.

[24] :Anomaly detection technique for sequential data.  
Muriel Pellissier Université de Grenoble, 2013

[25] :Monitoring et détection d'anomalie par apprentissage dans les infrastructures virtualisées. Carla Sauvanaud

[26] :<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=2ahUKEwikub7i4LLdAhVCr6QKHhAOeCnYQFjAEegQIBhAC&url=https%3A%2F%2Fraw.githubusercontent.com%2Fmda-sw%2Flibrary%2Fmaster%2Finform%2Fthese-philippe-poincot%2Fchap3.pdf&usg=AOvVaw2S3nBEpzAxDWUBFlpOYq6d>

[27]. RESEAUX DE NEURONES, par Marc Parizeau, Hiver 2006

[28]. Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels : une approche dans l'espace d'état, MERZOUKA NOURESSADAT, 2009

[29] : Marc Parizeau. RESEAUX DE NEURONES. 2004.

[30] : Pat Nakamoto. Neural Networks & Deep Learning: Deep Learning explained to your granny – A visual introduction for beginners who want to make their own Deep Learning Neural Network (Machine Learning). 2017.

[31] : The neural network of the basal ganglia as revealed by the study of synaptic connections of identified neurones, A. David Smith and J. Paul Bolam

[32] : L. Deng, D. Yu, et al., "Deep learning : methods and applications," Foundations and Trends R in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.

**[33]** : revu ;Deep learning, Yann LeCun<sup>1,2</sup>, Yoshua Bengio<sup>3</sup> & Geoffrey Hinton

**[34]** : Article Est-ce que le « Deep Learning » peut améliorer votre résultat net ? Juin 2016

**[35]** : Deep Learning in Neural Networks: An Overview, Jurgen Schmidhuber, 8 October 2014

**[36]** : The Python Language Reference, Guido van Rossum, June 27, 2018

**[37]** : Deep Learning made easy with Deep Cognition, Favio Vázquez, 2017

## Resumé

En utilisant un système de classification, on affecte automatiquement à chaque image une classe. On retrouve ainsi la classification des objets, des empreintes digitales, la reconnaissance des visages. Ce projet consiste à utiliser et créer un classificateur d'images avec un modèle de réseaux neurones convolutionnels et d'avoir des meilleurs résultats de classification.

## Abstract

Using a classification system, each image is automatically assigned a class. We thus find the classification of objects, fingerprints, the recognition of faces. This project involves using and creating an image classifier with a convolutional neural network model and having better classification results.

## ملخص

باستخدام نظام تصنيف ، يتم تعيين فئة لكل صورة تلقائيًا. وهكذا نجد تصنيف الأشياء وبصمات الأصابع والاعتراف بالوجوه. يتضمن هذا المشروع استخدام وتصنيف مصنف للصور مع نموذج شبكة عصبية ولتحقيق نتائج تصنيف أفضل.