



République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

# Mémoire de Master

## en Informatique

*Spécialité : Ingénierie des Systèmes d'Information et Logiciel*

## Thème

---

Modèle Deep learning pour différencier la pneumonie  
COVID-19 des autres pneumonies.

---

Encadré par

— M. ALIOUAT OUAHIBA.

Réalisé par

— MLE DOUMANI ROUMAÏSSA.

— MLE KERTOUS RADJA.

2020/2021.

# *Remerciements*

Avant de présenter ce travail, nous tenons à remercier Dieu tout puissant, de nous avoir permis d'arriver à ce niveau d'étude.

Nous tenons à exprimer par ces quelques lignes de remerciements notre gratitude envers tous ceux qui nous ont aidé, par leur présence, leur soutien, leurs disponibilité et leurs conseils afin d'accomplir ce projet.

Nous tenons encore à exprimer notre profonde gratitude et nos sincère remerciements à Mme. Aliouat Ouahiba qui nous a fait l'honneur de diriger ce travail et ses précieux conseils furent d'un apport considérable. Aussi nous tenons à lui reconnaître le temps précieux qu'elle nous a consacré.

Nos remerciements chaleureux et anticipés vont aux membres du jury qui ont consacré une partie de leur temps pour examiner et juger notre travail.

Enfin, nos sincères sentiments vont à tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet, en particulier nos chères familles et nos amis(es).

# *Dédicaces*

Avec l'expression de ma reconnaissance je dédie ce modeste travail :

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

À mes chers frères Anis et Amine, Islam, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour eux, ainsi que toute ma famille pour leur confiance et leur amour.

À mes chères sœurs Asmaa, Ahlem qui n'ont pas cessé de me conseiller, m'encourager et me soutenir dans toute ma carrière d'étude. Que Dieu les protège et leurs offre la chance et beaucoup de bonheur.

À mes très chères amies qui m'ont apporté la joie et beaucoup d'amour dans ma vie .

À ma chère binôme Roumaïssa.

*Radja Kertous.*

# *Dédicaces*

Avec l'expression de ma reconnaissance Je dédie ce modeste travail :

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

À mes deux chers frères Ahmed et Yacine . Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour eux, ainsi que toute ma famille pour leur confiance et leur amour.

À ma chère sœur Amira qui n'a pas cessé de me conseiller, m'encourager et me soutenir durant toute ma carrière d'étude. Que Dieu la protège et lui offre la chance et beaucoup de bonheur.

À mes très chères amies Meriem et Ahlem et mimi et Hayet qui m'ont apporté la joie et beaucoup d'amour dans ma vie .

À ma chère binôme Radja.

*Roumaïssa Doumani.*

# Résumé

Le nouveau coronavirus également appelé COVID-19 est originaire de Wuhan, en Chine, en décembre 2019 et s'est maintenant répandu dans le monde entier. Comme le nombre de cas augmente rapidement, la plupart des pays sont confrontés à une pénurie de kits de test et de ressources.

La quantité limitée de kits de test et un certain nombre de cas quotidiens, nous encourage à proposer un modèle d'apprentissage profond qui peut aider les radiologues et les cliniciens dans la détection des cas de COVID-19 et les autres types de pneumonie, à l'aide des images radiologies pulmonaires.

Les réseaux de neurones ont un fort potentiel dans ce domaine, ils résolvent les problèmes de manière identique avec l'humain.

L'objectif de ce travail est de proposer une approche du Deep Learning dans le domaine de la pathologie médicale pour détecter le COVID-19. Pour ce faire, nous avons choisi d'utiliser les réseaux de neurones convolutifs (CNN) grâce à son efficacité pour résoudre les problèmes de classification d'images.

**Mots clés :** CNN, Classification, COVID-19, Réseaux de neurones, Réseaux de neurones convolutifs . . .

# Abstract

The novel Coronavirus also called COVID-19 originated in Wuhan, China in December 2019 and has now spread across the world. As the number of cases are rapidly increasing, most of the countries are facing shortage of testing kits and resources. The limited quantity of testing kits and increasing number of daily cases encouraged us to come up with a Deep Learning model that can aid radiologists and clinicians in detecting COVID-19 cases using chest X-rays.

Neural networks have great potential in this area. They can solve problems the same way they can be solved by humans.

The objective of this work is to propose a deep learning approach in the field of medical pathology to detect for covid-19. To do this, we have chosen to use convolutional neural networks (CNN) thanks to its efficiency in solving image classification problems.

**Key words :** CNN, Classification, COVID-19, Neural networks, Convolutional neural networks ...

# Table des matières

- Table des matières** **i**
  
- Table des figures** **iv**
  
- Liste des tableaux** **vi**
  
- Liste des abréviations** **viii**
  
- Introduction générale** **1**
  
- 1 Présentation des pneumonies** **3**
  - 1.1 Introduction . . . . . 3
  - 1.2 Définition de la pneumonie . . . . . 3
  - 1.3 Classification de la pneumonie . . . . . 4
    - 1.3.1 Les germes responsables . . . . . 4
    - 1.3.2 Les différents types de pneumonie . . . . . 5
    - 1.3.3 La pneumonie covid 19 . . . . . 6
  - 1.4 Diagnostic étiologique des pneumonies . . . . . 6
  - 1.5 Imageries médicale et diagnostic des pneumonies . . . . . 7
    - 1.5.1 But de l'imagerie médicale . . . . . 7
    - 1.5.2 Techniques de diagnostic de la pneumonie . . . . . 8
  - 1.6 Conclusion . . . . . 8
  
- 2 Etat de l'art** **9**
  - 2.1 Intoduction . . . . . 9

2.2	Intelligence artificielle, machine learning et le deep learning . . . . .	9
2.3	L'apprentissage en profondeur (deep Learning) . . . . .	11
2.3.1	Définition . . . . .	11
2.3.2	Historique de deep learning . . . . .	11
2.3.3	Domaine d'application du Deep Learnig . . . . .	13
2.3.4	Principe de fonctionnement de deep learning . . . . .	13
2.3.5	Architectures et algorithmes . . . . .	14
2.4	Les réseaux de neurones convolutifs . . . . .	14
2.4.1	Principe d'architecture d'un CNN . . . . .	15
2.4.2	Les différentes couches de CNN . . . . .	16
2.5	La fonction du cout . . . . .	21
2.5.1	Définition . . . . .	21
2.5.2	Réduction de la fonction du cout . . . . .	21
2.6	Le problème overfitting . . . . .	22
2.6.1	Définition . . . . .	22
2.6.2	Techniques pour éviter Overfitting . . . . .	22
2.7	Les travaux de recherche basés sur l'architecture CNN pour la classification de pneumonie . . . . .	23
2.8	Conclusion . . . . .	28
<b>3</b>	<b>Solution proposée</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Présentation de données . . . . .	29
3.2.1	Distribution de notre jeux de données . . . . .	30
3.2.2	Prétraitement de données . . . . .	31
3.3	Architectures proposées . . . . .	32
3.3.1	Les modèles pré-entraînés utilisés . . . . .	32
3.3.2	Utilisation de l'architecture DenseNet201 . . . . .	33
3.3.3	Utilisation de l'architecture DenseNet169 . . . . .	34
3.3.4	Concaténation entre deux architectures InceptionV3 et Inception- ResNetV2 . . . . .	35
3.3.5	Concaténation entre deux architectures VGG16 et DenseNet169 . .	36
3.3.6	Entraînement de nos modèles . . . . .	37

3.4 Conclusion . . . . .	38
<b>4 Résultats et discussion</b>	<b>39</b>
4.1 Intoduction . . . . .	39
4.2 Environnement de développement . . . . .	39
4.2.1 Langage de programmation . . . . .	39
4.2.2 Plateforme de développement . . . . .	40
4.2.3 Bibliothèques utilisées . . . . .	41
4.3 Indicateurs de performances de la classification . . . . .	41
4.3.1 Matrice de confusion . . . . .	42
4.4 Implémentation . . . . .	43
4.4.1 Paramétrage des modèles . . . . .	43
4.4.2 Résultats expérimentaux obtenus et discussion . . . . .	45
4.5 obstacles . . . . .	56
4.6 Conclusion . . . . .	57
 <b>Conclusion générale</b>	 <b>58</b>
 <b>Bibliographie</b>	 <b>60</b>
 <b>A Fragments de code</b>	 <b>66</b>

# Table des figures

1.1	La différence entre poumon sain et pneumonie. [1]	4
1.2	Illustration des modalités d'imagerie médicale.[11]	7
2.1	La relation entre IA, le ML et le DL [15]	10
2.2	: le mécanisme d'activation d'un neurone [18]	13
2.3	:Architectures d'apprentissage en profondeur [20]	14
2.4	Architecture d'un réseau de neurone convolutif[24]	15
2.5	Représentation d'une RGB image.[25]	16
2.6	Exemple de principe du filtre convolutif [25].	17
2.7	Principe de la fonction ReLU[30].	18
2.8	Types de Pooling[25].	19
2.9	Types de Pooling [31].	20
2.10	Couche entièrement connectée (Fullyconnected)[31].	21
3.1	Distribution des données.	30
3.2	Quelque échantillons de notre base de données.	31
3.3	Architecture du DenseNet avec trois blocs.[56]	33
3.4	Notre architecture basé sur DenseNet201.	34
3.5	Notre architecture basée sur DenseNet169.	35
3.6	Notre architecture basée sur InceptionResNetV2 et InceptionV3.	36
3.7	Notre architecture basée sur VGG16 et DenseNet169.	37
4.1	Un aperçu sur le résultat initiale et final de la phase apprentissage.	45
4.2	Courbes accuracy et loss obtenus pour le modèle-1	46

4.3	Courbes accuracy et loss obtenus pour le modèle-2 . . . . .	46
4.4	Courbes accuracy et loss obtenus pour le modèle-3 . . . . .	47
4.5	Courbes accuracy et loss obtenus pour le modèle-4 . . . . .	47
4.6	Matrice de confusion du modèle-1. . . . .	48
4.7	Matrice de confusion du modèle-2. . . . .	49
4.8	Matrice de confusion du modèle-3. . . . .	50
4.9	Matrice de confusion du modèle-4. . . . .	51
4.10	Résultat de teste de notre modèle sur quelques exemple de teste. . . . .	55
4.11	Matrices de confusion de la classification binaire et trois classes . . . . .	56
A.1	L'augmentation de données. . . . .	66
A.2	Les couches de la partie classification. . . . .	66

# Liste des tableaux

- 2.1 Historique du deep learning. [17]. . . . . 12
- 2.2 Travaux réalisés dans la détection des différents types de pneumonie. . . . . 27
  
- 4.1 Métriques d'évaluations de notre modèle. . . . . 43
- 4.2 Paramétrage de nos modèles. . . . . 44
- 4.3 Mesure de performance du modèle-1. . . . . 48
- 4.4 Mesure de performance du modèle-2. . . . . 49
- 4.5 Mesure de performance du modèle-3. . . . . 50
- 4.6 Mesure de performance du modèle-4. . . . . 51
- 4.7 Comparaison entre les 4 modèles . . . . . 53
- 4.8 Comparaison entre notre modèle proposé et quelques travaux de littérature. 54



## Liste des abréviations

1D	One Dimension
2D	Two Dimensional
3D	Three Dimensional
ADN	Acide Désoxyribo Nucléique
Adam	Adaptive Moment Estimation
AI/IA	Intelligence Artificielle
API	Application Programming Interface
Acc	Accuracy
AUC	Area Under the ROC Curve
BS	Batch size
CPU	Central Processing Unit
CNN/ConvNet	Convolutional Neural Networks /Réseaux de Neurones Convolutifs
CT-scan	Computerized tomography-scanner
DL	Deep Learning
DNN	Deep Neural Networks
FC	Fully Connected
FN	False Negative
FP	False Positive
F1-Sc	F1-Score
HTML	HyperText Markup Language
GPU	Graphical Processing Unit
Lr	learning rate
ML	Machine Learning
MSVM	Multicategory Support Vector Machines
MH	Moyenne Harmonique...
Pre	Précision
RGB	Red-Green-Bleu

ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
Sp	Specificity
Se	Sensitivity
Sen	Sensitivité
SVM	Support Vector Machine
TPU	Tensor Processing Unit
TN	Correct Negative
TP	Correct Positive

# Introduction générale

Les pneumonies représentent une énorme menace pour l'humanité de nos jours, surtout avec la pandémie actuelle de COVID-19 qui provoque actuellement une grave crise mondiale avec des milliers de cas positifs et un nombre important de décès.

A l'heure actuelle, des essais pour détecter la présence de COVID-19 sont effectués sur la base de la réaction en chaîne par polymérase de transcription inverse (RT-PCR), qui prend généralement 24 heures pour donner le résultat. De plus, la disponibilité des kits de dépistage pose un autre problème.

Les différents types de pneumonie incluant le COVID-19 peuvent être détectés sur les clichés radiologiques pulmonaires.

Le domaine de l'imagerie médicale a connu des changements majeurs ces dernières années, en raison de la grande avancée dans le domaine de l'apprentissage profond et de la vision par ordinateur, l'imagerie médicale et l'intelligence artificielle jouent un rôle clef dans le diagnostic et le traitement de plusieurs maladies.

Les différents types de pneumonie représentent des anomalies dans le bilan radiologique, toutes ces anomalies sont remarquables par le radiologue mais ca reste une tâche fatigante, susceptible d'impliquer une erreur humaine et la chose la plus importante est qu'il faut tellement de temps pour savoir traiter le grand nombre des images radiologiques des personnes infectées.

Notre travail vise à résoudre le problème du temps en construisant un modèle de Deep Learning qui différencie automatiquement la pneumonie COVID-19 des autres types de

pneumonies dans les images radiologiques.

Nous réalisons dans notre travail un couplage entre deux domaines la médecine et l'intelligence artificielle, nous allons proposer une solution (architecture) basée sur le Deep Learning en utilisant les réseaux de neurones convolutifs (CNN) pour classifier les images de notre ensemble de données selon 4 classes (pneumonie bactérienne, pneumonie viral, pneumonie COVID-19, normal).

**Organisation du manuscrit :** Ce manuscrit est organisé comme suit :

**Chapitre 1 :** Est consacré à la partie théorique qui donne un aperçu général sur la pneumonie et ses différents types pour comprendre son aspect médical.

**Chapitre 2 :** Présente une étude théorique sur le Deep Learning et quelques travaux connexes réalisés dans le domaine de classification des différents types de pneumonie .

**Chapitre 3 :** Présente l'ensemble de données utilisé ainsi que le détail de nos architectures proposées.

**Chapitre 4 :** Est composé de deux parties :

- La première partie : présente les outils de programmation et l'implémentation,
- La deuxième partie : présente la discussion des résultats obtenus après une comparaison entre nos différents modèles.

En fin, nous terminerons ce travail par une conclusion générale.

# Présentation des pneumonies

## 1.1 Introduction

Le système respiratoire humain est sensible à de nombreuses inflammations et infections, dont la plus importante est la pneumonie.

La pneumonie est une pathologie du tissu pulmonaire qui peut atteindre des personnes de tout âge, mais le plus grand risque concerne les jeunes enfants, les personnes âgées, et les patients immuno-déprimés.

Dans ce chapitre, nous allons d'abord faire un petit rappel sur cette maladie, nous énumérerons les germes responsables de cette dernière, ses différents types et son diagnostic étiologique. Ceci nous permettra par ailleurs de mieux comprendre son aspect médical. Ensuite, nous définirons la pneumonie COVID-19. Enfin, nous présentons le rôle de l'imagerie médicale pour le diagnostic des différents types de pneumonie.

## 1.2 Définition de la pneumonie

La pneumonie est une infection des voies respiratoires inférieures qui affecte le système respiratoire. Elle peut avoir de nombreuses origines, les plus courantes étant les bactéries et les virus.

Généralement, les germes présents dans l'air pénètrent dans les poumons par les voies respiratoires lors de la respiration, lorsque ces derniers arrivent aux alvéoles, ils provoquent une inflammation. à ce stade on aura l'apparition des symptômes de pneumonie.[1]

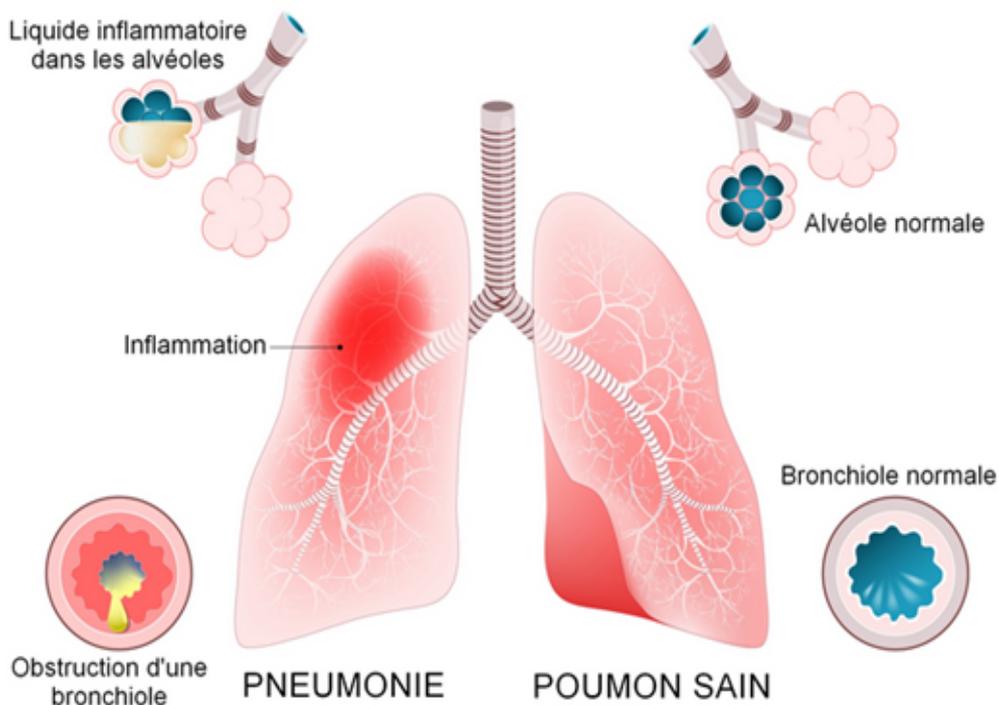


FIGURE 1.1 – La différence entre poumon sain et pneumonie. [1]

## 1.3 Classification de la pneumonie

### 1.3.1 Les germes responsables

La pneumonie peut être due à

**Des bactéries** : La bactérie est un micro-organisme unicellulaire et sans noyau dont le génome est constitué d'ADN. Celui-ci consiste en un chromosome, et on note éventuellement la présence de plasmides (petit morceau d'ADN circulaire) [2].

**Des virus** : Les virus sont des agents infectieux microscopiques possédant un seul type d'acide nucléique (ADN ou ARN) [3].

**Des levures** : Les levures sont des micro-organismes unicellulaires qui appartiennent au domaine des champignons [4].

**Des parasites** : Un parasite est un organisme vivant qui ne peut survivre et se développer que dans un autre organisme, il utilisera donc ses cellules, sa nourriture et ses tissus pour se développer et assurer sa progéniture [5].

### 1.3.2 Les différents types de pneumonie

La pneumonie est une infection pulmonaire qui constate plusieurs types selon le germe, les types les plus connus sont :

**La pneumonie virale :** C'est une forme qui est généralement peu symptomatique où les deux poumons sont souvent atteints.

Les virus les plus fréquemment mis en cause dans la pneumonie virale sont les virus influenza et parainfluenzae, le virus respiratoire syncytial, ou les rhinovirus [1].

**La pneumonie bactérienne :** La pneumonie bactérienne est la principale cause de décès infectieux, cette maladie affecte une partie des deux poumons.

La pneumonie bactérienne la plus courante est causée par *Streptococcus pneumoniae* (abrégié en pneumocoque), qui est la cause bactérienne de la pneumonie qui peut être mortelle [6].

**La pneumonie parasitaire :** Ce type se rencontre surtout chez les personnes immuno-déprimées.

L'aspergillose pulmonaire, ainsi que la pneumocystose sont des exemples [1].

Autres types :

**La pneumonie d'aspiration :** Ce type de pneumonie se produit lorsqu'un peu de liquide contenu dans l'estomac est aspiré dans les poumons, ce liquide cause une infection à travers les bactéries contenues dans ce dernier. Ce phénomène survient généralement après une anesthésie ou à la suite d'un trouble neurologique touchant le réflexe de déglutition ou d'inhalation de vomissements [1].

**Les pneumonies secondaires à une autre infection :** La pneumonie est ainsi une complication très fréquente de la grippe surtout chez les personnes très jeunes ou très âgées qui sont très fragiles. On note surtout dans ce type la bronchite qui se complique en pneumonie [6].

On constate aussi d'autres inflammations pulmonaires comme celles causées par des produits toxiques (peinture, vernis ...), une allergie à des produits employés en milieu professionnel, cette inflammation est d'origine non infectieuse [1].

### 1.3.3 La pneumonie covid 19

#### 1.3.3.1 Notion épidémiologique

Fin décembre 2019, une série de cas de pneumonie virale provoquée par un nouveau corona virus est apparue à Wuhan, en Chine, et s'est rapidement propagée sur tous les continents, ce corona virus, identifié sur prélèvements des voies aériennes, a été nommé SARS-CoV-2 pour Severe Acute Respiratory Syndrome CoronaVirus-2 par l'ICTV (International Committee On Taxonomy of Viruses) [7].

La maladie qu'il provoque a été dénommée COVID-19 pour COronaVirusDisease 2019 par l'OMS. Le SARS-CoV-2 appartient à la famille des Corona viridae, tout comme le SARS-CoV et le MERS-CoV qui furent responsables d'épidémies de grande ampleur en 2003 et 2012 respectivement [7].

#### 1.3.3.2 Les voies de contamination

Selon les informations disponibles à ce jour, le COVID-19 se propage par :

Les gouttelettes respiratoires générées lorsqu'une personne infectée parle ou tousse.

- Un contact rapproché entre les personnes (moins de 2 mètres, plus de 15 minutes).
- Contact indirect, à travers des surfaces et des objets contaminés.
- Dans les espaces confinés, une mauvaise ventilation, une forte densité d'occupants,

le risque de transmission augmente [8].

## 1.4 Diagnostic étiologique des pneumonies

Le diagnostic étiologique des pneumonies consiste à identifier l'agent responsable de la maladie, ce diagnostic peut être effectué en utilisant plusieurs techniques sur un prélèvement, on note :

**L'examen directe** : Désigne l'examen d'un échantillon pour visualiser des bactéries, des parasites ou champignons.

**Polymérase Chain Réaction(PCR)** : Une réaction enzymatique qui permet de sélectionner puis d'amplifier en une très grande quantité un fragment d'ADN particulier, parmi des milliers d'autres fragments [9].

**La sérologie** : La sérologie est une méthode biologique permettant d'établir des diagnostics, via l'étude des sérums et de ce qu'ils contiennent, notamment des anticorps

spécifiques pouvant être liés à la présence de certains agents pathogènes [10].

## 1.5 Imageries médicale et diagnostic des pneumonies

L'imagerie médicale est un domaine de médecine qui a connu un bouleversement ces vingt dernières années, elle regroupe différents techniques qui permettent de voir l'intérieur du corps humain[11].

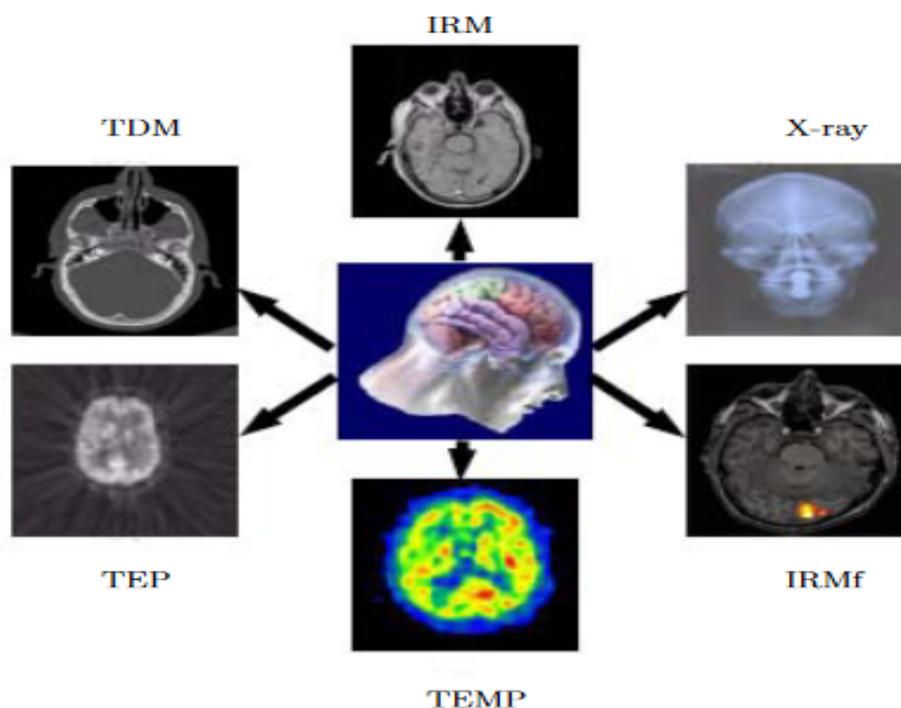


FIGURE 1.2 – Illustration des modalités d'imagerie médicale.[11]

### 1.5.1 But de l'imagerie médicale

Le but de l'imagerie médicale est de fournir une représentation visuelle claire du corps humain (structure et fonction des organes) sans intervention directe, et de fournir un suivi très précis du processus de la maladie [11].

L'apport de l'imagerie médicale dans la prise en charge des maladies depuis leur diagnostic jusqu'à leur traitement est devenu irremplaçable. Le recours à une imagerie médicale de plus en plus performante, en scanographie notamment, améliore la qualité du diagnostic et permet de mieux orienter la stratégie thérapeutique et d'évaluer l'efficacité des

traitements[11].

### 1.5.2 Techniques de diagnostic de la pneumonie

Les différents appareils d'acquisition d'images médicales sont nombreux et permettent d'obtenir des informations différentes selon le procédé physique utilisé pour observer les tissus des poumons et détecter les anomalies qui sont provoquées aux niveaux de ces derniers.

**La radiographie du thorax :** Une technique qui consiste à traduire l'opacité par une teinte plus ou moins claire et visualiser le thorax et ses composants pour diagnostiquer les pathologies qui peuvent l'attaquer, elle consiste à projeter des rayons X sur le corps afin de créer une image sur le tableau de capture numérique.

Elle peut donner une probabilité de pneumonie sur les seules constatations radiologiques.

**Le scanner du thorax :** Il s'agit d'un examen utilisant des rayons X pour explorer les structures anatomiques du thorax , notamment le thorax.

La tomodensitométrie thoracique est largement utilisée pour détecter la pneumonie car elle peut :

- diagnostiquer diverses lésions pulmonaires.
- évaluer la réponse thérapeutique.

**Échographie :** L'échographie est une technique d'imagerie par ultrasons, elle présente comme avantages la :

- elle est non invasive, non irradiante, disponible au lit du patient
- elle peut donner un diagnostic précoce des formes sévères du COVID-19 et de suivre l'évolution des lésions pulmonaires [13].

## 1.6 Conclusion

Dans ce chapitre, nous avons présenté des généralités expliquant la pneumonie et ses différents types. Aussi nous avons parler de l'imagerie médicale et son rôle dans le diagnostic des pneumonies.

Dans le chapitre suivant, nous allons présenter le Deep Learning (DL) ainsi que ses différentes méthodes d'apprentissage.

# Chapitre 2

## Etat de l'art

### 2.1 Introduction

Le Deep Learning est un ensemble de méthodes automatiques tentant de modéliser avec haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.

Dans ce chapitre, nous allons présenter quelques notions du Deep Learning, nous détaillerons par la suite les réseaux de neurones convolutifs et on va citer quelques travaux qui sont déjà réalisés au sein de la classification des pneumonies à partir des images Chest-X-Ray et CT-scan.

### 2.2 Intelligence artificielle, machine learning et le deep learning

**Intelligence artificielle (IA) :** La notion a vu le jour dans les années 1950 grâce au mathématicien Alan Turing. Ce dernier soulève la question d'apporter aux machines une forme d'intelligence, consistant à mettre en œuvre un certain nombre des sciences, théories et techniques visant à permettre aux machines d'imiter les capacités cognitives d'un être humain. L'IA est un domaine général qui englobe l'apprentissage automatique et l'apprentissage en profondeur, mais qui comprend également de nombreuses autres approches qui n'impliquent aucun apprentissage. [14].

**Machine Learning(ML) :** L'apprentissage automatique est un sous-domaine de l'intelligence artificielle, qui donne à un système une capacité d'apprendre et de faire des

prédictions grâce à ses algorithmes à partir de données, et par cela les ordinateurs apprennent à résoudre des tâches spécifiques, sans avoir besoin de les programmer.[14].

Les différents types d'algorithmes Machine Learning peuvent être répartis en différentes catégories :

- Apprentissage supervisé (les algorithmes de classification et régression).
- Apprentissage non supervisé (les algorithmes de clustering et d'association ainsi réduction dimensionnelle).
- Apprentissage par Renforcement.

**Deep Learning (DL) :** L'apprentissage en profondeur est une forme d'intelligence artificielle dérivée du Machine Learning, il permet de résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance d'objets à travers son amélioration du développement des algorithmes traditionnels, son adaptation à n'importe quelle quantité de données telles que les big data et à n'importe quel type de problème avec une grande puissance de calcul et flexibilité, et son efficacité à l'extraction de caractéristiques automatiquement.

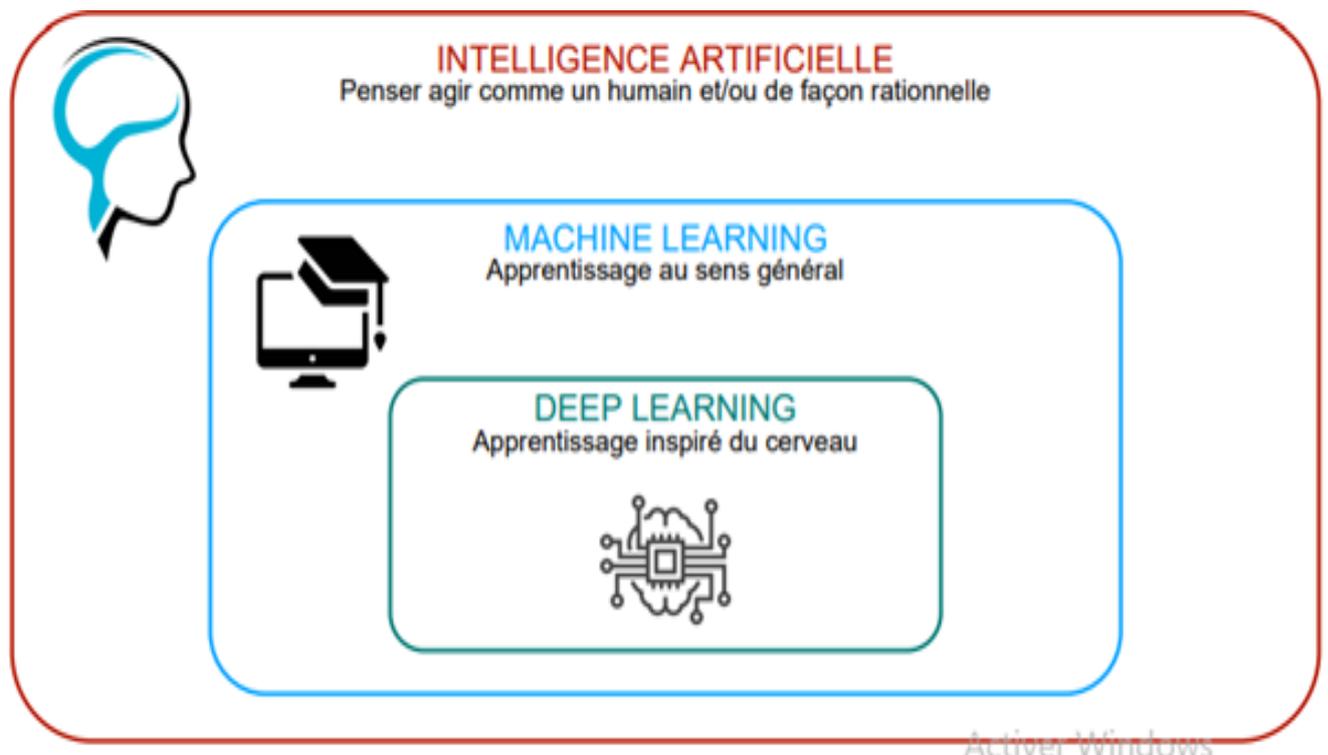


FIGURE 2.1 – La relation entre IA, le ML et le DL [15]

## 2.3 L'apprentissage en profondeur (deep Learning)

### 2.3.1 Définition

Selon les fondateurs Yann LeCun, Yoshua Bengio Geoffrey Hinton [16] : L'apprentissage profond permet aux modèles informatiques composés de plusieurs couches de traitement d'apprendre des représentations de données avec plusieurs niveaux d'abstraction.

Deep learning ou l'apprentissage profond, est un type d'apprentissage automatique dans lequel un modèle apprend à effectuer des tâches de classification directement à partir d'images, de texte ou de son. L'apprentissage profond est généralement implémenté à l'aide d'une architecture de réseau neuronal.[14].

### 2.3.2 Historique de deep learning

Le Deep learning est un nouveau concept qui émerge depuis les années 2000, mais ce n'est pas le même cas des réseaux de neurones artificiels (le concept de base du Deep Learning).

Toutes les étapes de l'évolution sont résumées dans le tableau suivant : .

Année	Contributeur	Contribution
1943	McCulloch et Pitts.	Introduction du McCullochPitts (MCP) modèle considéré comme L'ancêtre des réseaux de neurones artificielles.
1949	Donald Hebb.	Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes.
1958	Frank Rosenblatt.	Introduction du premier perceptron.
1974	Paul Werbos.	Introduction de la retro propagation.
1980	Teuvo Kohonen et Kunihiko Fukushima.	Introduction des cartes auto organisatrices Introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif
1986	Michael I. Jordan.	Définition et introduction des réseaux de neurones récurrent.
1990	Yann LeCun.	Introduction de LeNet qui a montré la capacité des réseaux de neurones profond.
1997	Schuster and Paliwal	Introduction des réseaux de neurones récurrent bidirectionnelles .
2012	Krizhevsky et al Googel.	Introduction de AlexNet qui a remporté le challenge ImageNet. Google dévoile son projet « Google Brain », un programme capable d'analyser des images.
2014	Szegedy et al	début du GoogLeNet qui était le gagnant du concours ImageNet de cette année.
2015	He et al.	Résiduel Net (ResNet)réseau de 152 couches qui explore une structure plus profonde , il a remporté le défi ImageNet à un niveau sans précédent..

TABLE 2.1 – Historique du deep learning. [17].

### 2.3.3 Domaine d'application du Deep Learning

Le Deep Learning est un domaine très utilisé notamment dans la détection d'objets et la classification, la restauration et la segmentation d'images. Il est déjà utilisé dans un certain nombre de secteurs telle que la Conduite autonome, la reconnaissance faciale et vocale, ainsi que la médecine.[18]

### 2.3.4 Principe de fonctionnement de deep learning

Le principe de fonctionnement du deep learning repose sur les réseaux de neurones artificiels, ces neurones sont regroupés en trois types de couches différents et chaque connexion entre ces neurones est associée à un poids :

- Une couche d'entrée : Elle est constituée de l'ensemble des variables d'entrée.
- Une couche de sortie : Elle est constituée de l'ensemble des neurones de sortie du réseau, c'est cette couche-là qui fournit les sorties principales.
- Une ou plusieurs couches cachées : Ce sont les couches qui se trouvent entre la couche d'entrée et la couche de sortie, elles définissent l'activité interne du réseau.

L'entraînement du réseau de neurones consiste à déterminer des valeurs de poids  $W_{ij}$ , qui permettent à la couche de sortie de classer précisément les images de l'ensemble d'entraînement. [19]

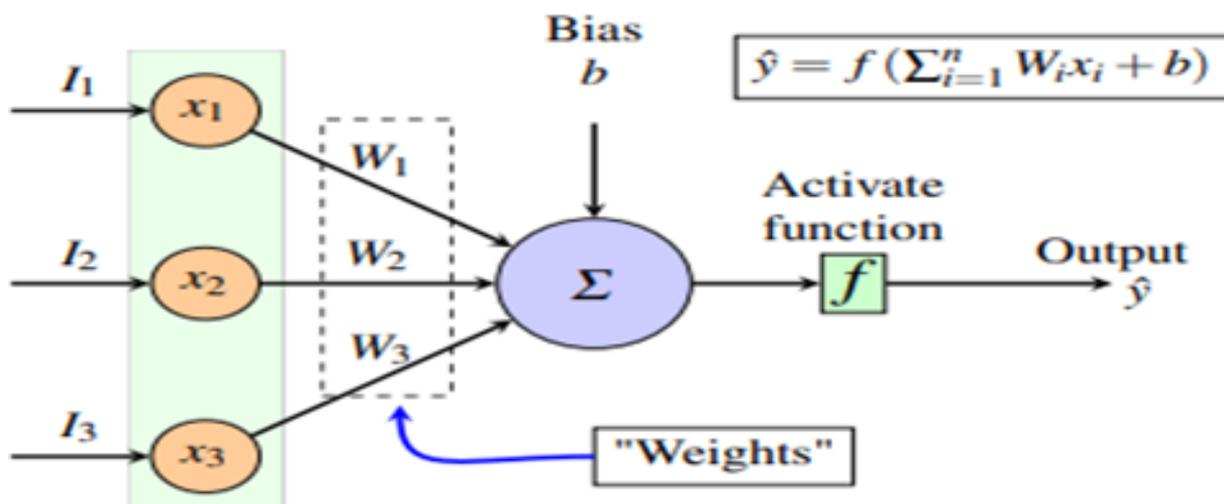


FIGURE 2.2 – : le mécanisme d'activation d'un neurone [18]

### 2.3.5 Architectures et algorithmes

Le Deep Learning est un domaine à croissance rapide, il englobe un grand nombre de diverses architectures profondes.[20].

la figure ci-dessous présente trois architectures supervisées de deep learning. Dans notre travail nous somme intéressées aux modèles de classification d'image CNN qui définit une sous-catégorie de réseaux de neurones.

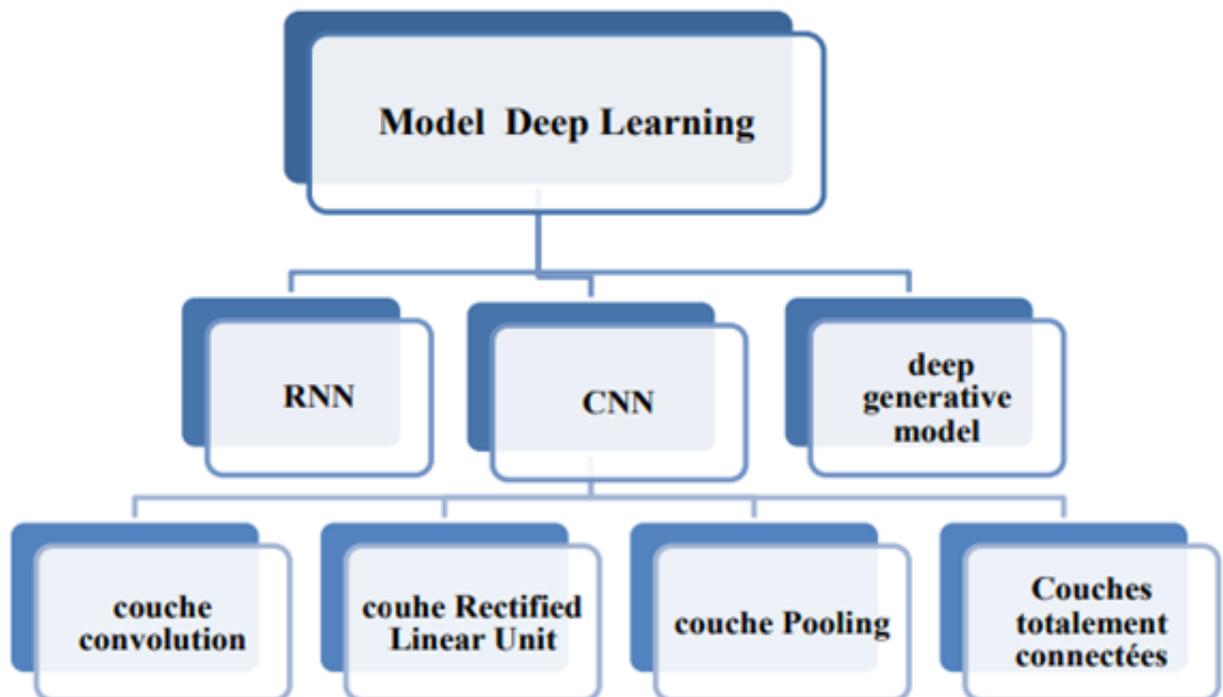


FIGURE 2.3 – :Architectures d'apprentissage en profondeur [20]

## 2.4 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN pour Convolutional Neural Networks) sont des types de réseau de neurones qui sont spécialisés dans des tâches de reconnaissance de forme, CNN a réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures [21].

### 2.4.1 Principe d'architecture d'un CNN

Le rôle du CNN est de réduire les images sous une forme plus facile à traiter, sans perdre les fonctionnalités essentielles pour obtenir une bonne prédiction. Son mode de fonctionnement est simple : l'utilisateur fournit en entrée une image sous la forme d'une matrice de pixels. celle-ci dispose de 3 dimensions :

- Deux dimensions pour une image en niveaux de gris.
- Une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu) [22].

La première partie de l'architecture consiste à faire passer l'image en une succession de couches de convolution et des filtres, créant de nouvelles images appelées cartes de convolutions et d'agrégation qui diminuent grandement la taille des données. Cette partie est dédiée à l'extraction automatique de caractéristiques, tandis que la seconde partie est composée de couches de neurones complètement connectés, est destinée à la classification, pour un apprentissage plus rapide, des fonctions d'activations seront utilisées, une fois que ces couches sont regroupées, une architecture CNN a été construite[23].

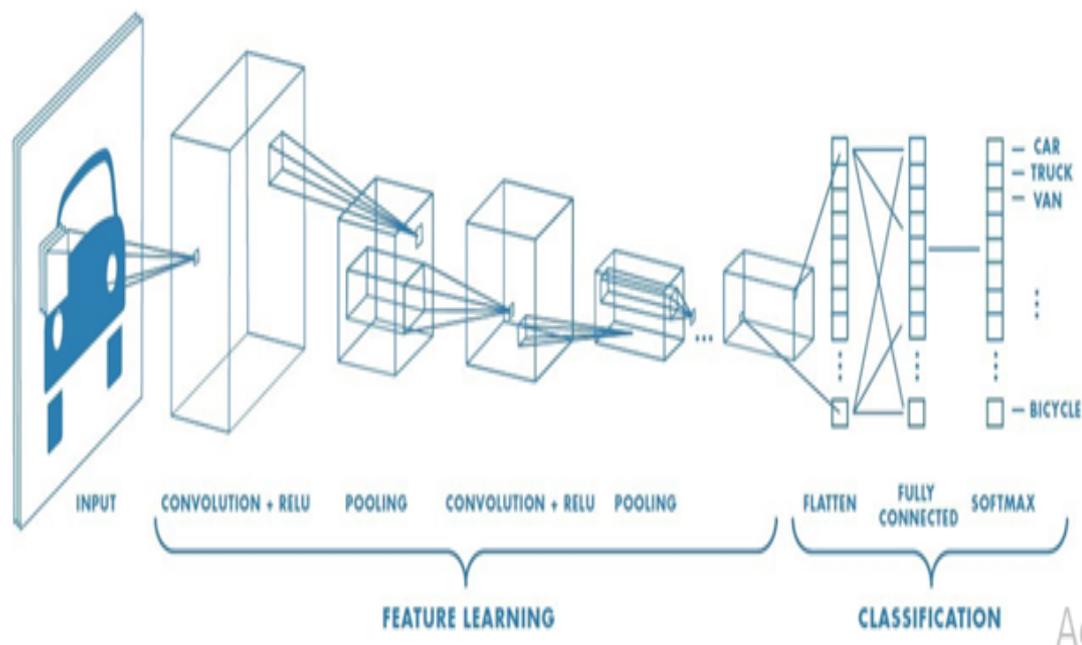


FIGURE 2.4 – Architecture d'un réseau de neurone convolutif[24]

## 2.4.2 Les différentes couches de CNN

Il existe quatre principales opérations dans le CNN, tout en ayant un rôle bien défini.

### 2.4.2.1 Couche d'entrée du CNN :

Contient des données décrivant généralement l'image, ces dernières sont considérées comme une matrice de valeurs du pixels tridimensionnelle qui doit être en général remodelée en une seule colonne (représentation vectorielle).



FIGURE 2.5 – Représentation d'une RGB image.[25]

### 2.4.2.2 Couche de convolution :

Est la composante clé des réseaux de neurones convolutifs, très utilisée en retouche d'image, car elle permet d'en faire ressortir des caractéristiques à partir des images d'entrées, afin d'appliquer un bon filtre.[26]

Tout d'abord, une partie de l'image est connectée à la première couche pour effectuer une opération de convolution et calculer le produit scalaire entre le champ récepteur (c'est une région locale de l'image d'entrée ayant la même taille que celle du filtre) et le filtre. La sortie de cette couche est l'image entrée avec des modifications qui est souvent appelée une carte de caractéristique (FeatureMap). Cette opération est répétée par le même processus

encore et encore jusqu'à ce que toute l'image soit parcourue et nous permet de trouver des parties de l'image qui pourraient être intéressantes [26].



FIGURE 2.6 – Exemple de principe du filtre convolutif [25].

La couche de convolution contient également l'activation ReLU, Ainsi on constate trois hyper paramètres permettent de dimensionner le volume de cette couche :

- **Profondeur de la couche** : Nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- **Stride** : Est le nombre de pixels par lequel nous glissons notre matrice de filtre sur la matrice d'entrée pour contrôler le chevauchement des champs récepteur.
- **Padding** : Parfois, il est recommandé de mettre des zéros à la frontière du volume d'entrée, la taille de ce zero-padding est le troisième hyper paramètre.

Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée[27].

— **Fonction d'activation** : La fonction d'activation est une composante essentielle du réseau neuronal, c'est une transformation non linéaire de la valeur d'entrée, elle calcule la somme pondérée des entrées et ajoute le biais.

Son but principal est de pouvoir permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire. De nombreuses fonctions d'ac-

tivation ont été proposées, parmi lesquelles on trouve : sigmoïde Tangente hyperbolique, ReLU et Softmax [28].

Rectified Linear Unit (ReLU) : Cette opération élémentaire doit être appliquée à chaque pixel d'une image après convolution en laissant toutes les valeurs positives passer inchangées et attribuent simplement 0 aux valeurs négatives, au cas où aucune fonction d'activation n'est appliquée la fonction créée sera linéaire et le problème XOR persiste .

ReLU est très utilisée dans les réseaux de neurones à convolution car il s'agit d'une fonction rapide à calculer, sa performance est donc meilleure que d'autres fonctions où des opérations coûteuses doivent être effectuées[29].

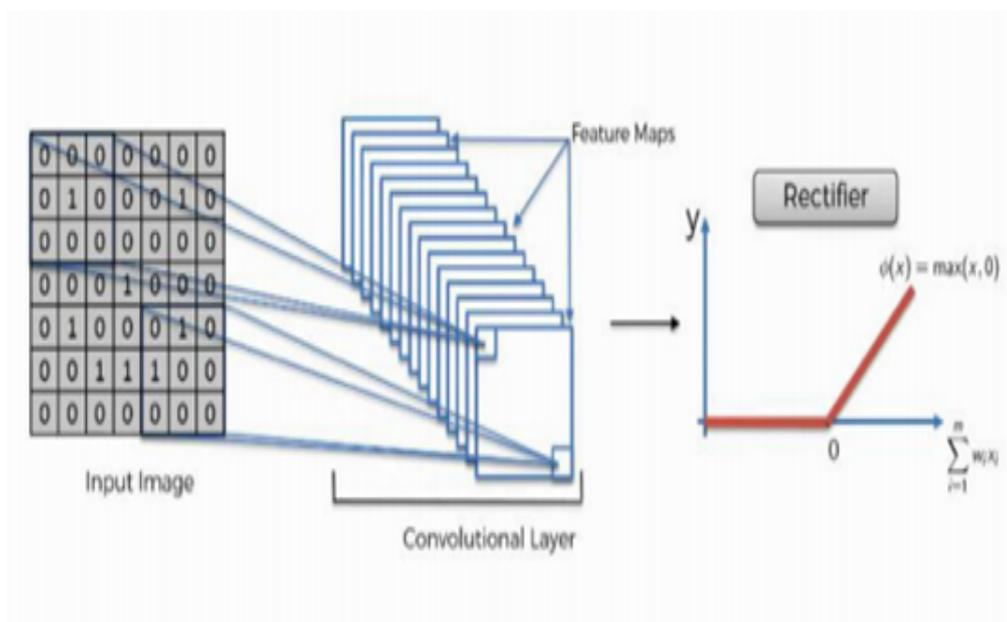


FIGURE 2.7 – Principe de la fonction ReLU[30].

#### 2.4.2.3 Couche de sous-échantillonnage (Pooling) :

Il est courant d'insérer périodiquement une couche Pooling entre les couches de convolution successives dans une architecture CNN, sa fonction est de réduire progressivement la taille spatiale de la représentation afin de réduire la quantité de paramètres et de calculs dans le réseau tout en préservant leurs caractéristiques importantes, et donc de contrôler également le sur-ajustement. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionner spatialement, on ob-

tient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites. En utilisant deux types de regroupement : le regroupement maximale et le regroupement moyen.

Ainsi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage[25].

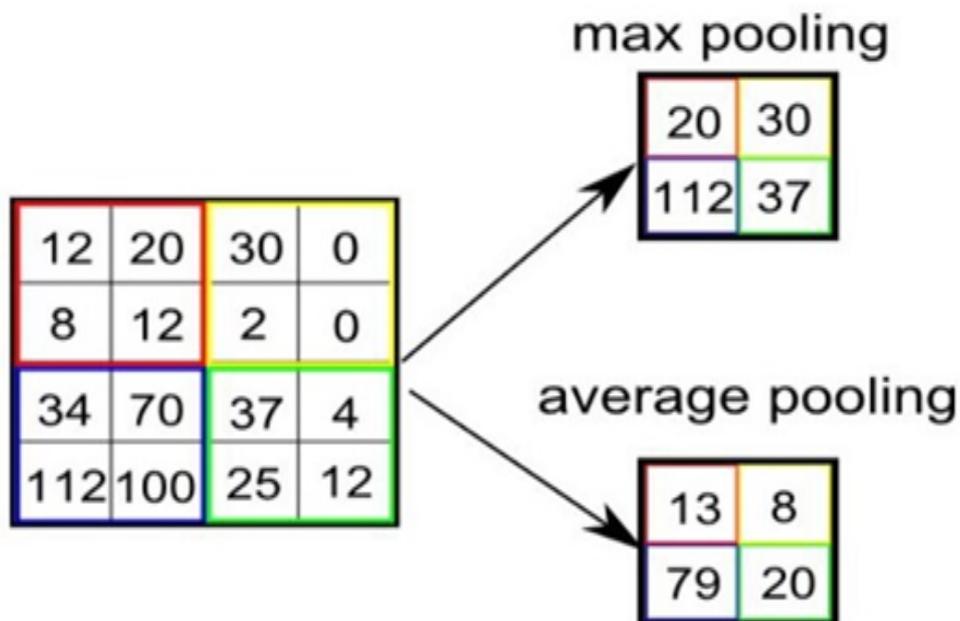


FIGURE 2.8 – Types de Pooling[25].

#### 2.4.2.4 Flattening :

Cette étape consiste à aplatir la carte d'entité regroupé obtenu a travers les deux étapes précédentes dans une colonne comme dans la figure 2.9, pour le but d'insérer ces données dans un réseau neuronal artificiel plus tard [31].

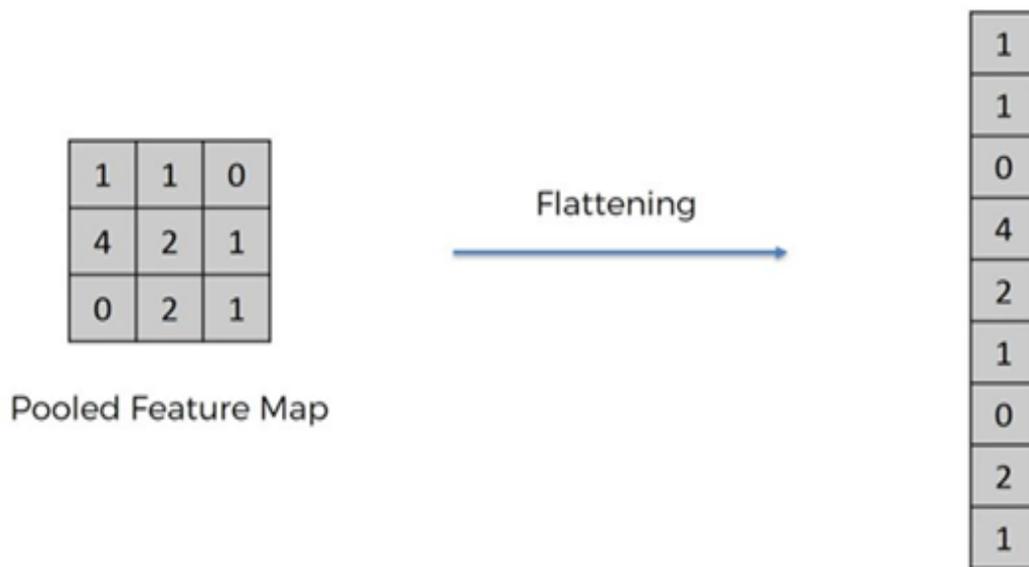


FIGURE 2.9 – Types de Pooling [31].

#### 2.4.2.5 Couche entièrement connectée (Fullyconnected)

Le terme « entièrement connecté » implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante.

Cette couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie de  $K$  dimensions, où  $K$  représente le nombre de classes que le réseau sera capable de prédire[27].

Pour compter les probabilités, la couche FC multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si nombre de classes=2, softmax si nombre de classes plus grand que 2) [32].

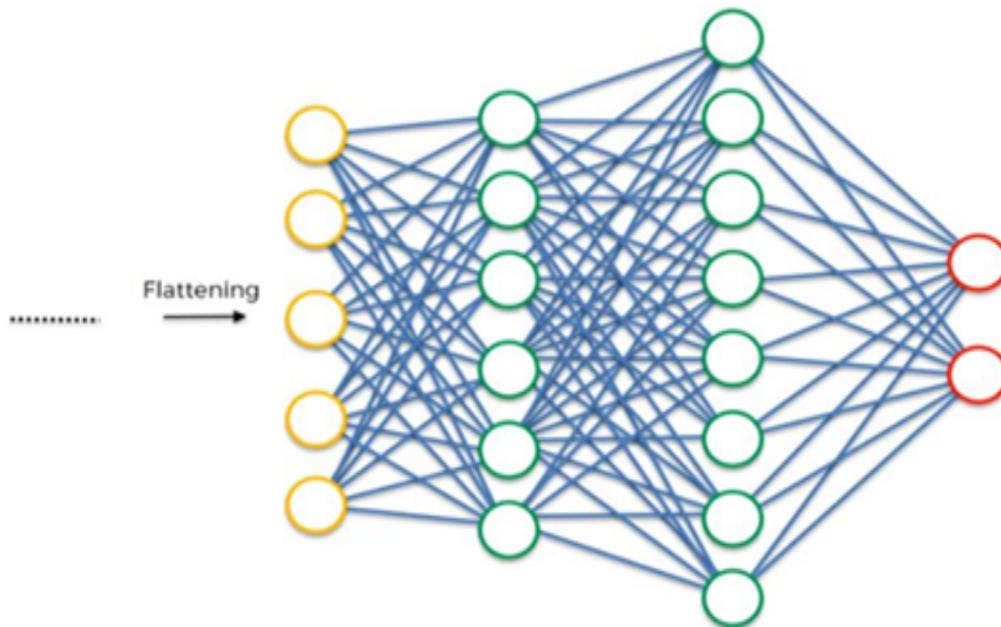


FIGURE 2.10 – Couche entièrement connectée (Fullyconnected)[31].

#### 2.4.2.6 Couche de sortie (output layer)

La couche de sortie contient l'étiquette qui est sous forme codée.

## 2.5 La fonction du cout

### 2.5.1 Définition

C'est une fonction qui peut être créée après le parcours de l'ensemble de données afin de montrer dans quelle mesure les sorties de modèle étaient erronées par rapport aux sorties réelles.

### 2.5.2 Réduction de la fonction du cout

- **La descente de gradient** : C'est l'algorithme le plus célèbre d'optimisation dans lequel se sert de trouver les poids ou les coefficients des algorithmes d'apprentissage automatique qui minimisent l'erreur de modèle sur l'ensemble de données.

La façon dont cela fonctionne est de permettre au modèle de faire des prédictions sur les données d'entraînement et d'utiliser l'erreur dans la prédiction pour mettre

à jour le modèle afin de réduire l'erreur et augmenter la précision, pour ce faire, il modifie le modèle en le déplaçant le long de la pente d'erreur jusqu'à la valeur d'erreur minimale.

il existe trois variantes de cette méthode, en fonction de la quantité de données : Batch gradient descent, descente de gradient stochastique, mini-batch gradient descent .[33].

## 2.6 Le problème overfitting

### 2.6.1 Définition

Overfitting ou learning-by-heart désigne l'incapacité de modèle à généraliser sur des données de test, car les données d'entraînement sont apprises par cœur. Autrement dit, se produit soit à cause des limites des données d'apprentissage, qui peuvent avoir une taille limitée ou inclure beaucoup de bruits, soit des contraintes d'algorithmes, qui sont trop compliqués et nécessitent trop de paramètres.

### 2.6.2 Techniques pour éviter Overfitting

Il existe plusieurs techniques pour éviter ce problème , parmi lesquels :

- **Cross-validation** : Une des méthodes les plus efficaces pour éviter overfitting, elle permet de régler les hyper-paramètres avec uniquement l'ensemble d'entraînement d'origine ou on le divise en plusieurs groupes. L'idée en suite est d'entraîner le modèle sur tous les groupes sauf un.

Si on a  $k$  groupes, on entraînera le modèle  $k$  fois avec à chaque fois un nouveau groupe de test ensuite, on calcule une estimation de la performance sur les différents entraînements. Cette technique de validation croisée est appelée  $k$ -fold [34].

- **Data augmentation** : Il s'agit d'un ensemble de techniques utilisées pour augmenter artificiellement la taille d'un ensemble de données en appliquant des transformations aux données existantes.

Par exemple, la rotation des images , la transformation en niveaux de gris ou bien la modification de la saturation des couleurs. Donc l'augmentation des données peut être un outil très puissant, mais elle nécessite un examen et une compréhension

attentive des données[35].

- **Méthodes de régularisation** : Cette technique est explicitement conçue pour réduire l'erreur de test, visent à limiter overfitting pour conduire à de favori performances.

Il existe diverses méthodes de régularisation parmi lesquels le Dropout.

— **Le Dropout** : C'est une technique où les nœuds choisis au hasard sont ignorés dans le réseau pendant la phase d'apprentissage à chaque étape et cela signifie que leur contribution à l'activation des neurones qui leur succède est temporairement supprimé lors de la phase de propagation. Ce taux d'abandon est généralement de 0,5 et l'abandon peut être réglé pour produire les meilleurs résultats et améliore également la vitesse d'entraînement. Cette méthode de régularisation réduit les interactions nœud à nœud dans le réseau, ce qui conduit à l'apprentissage de fonctionnalités importantes et aide également à mieux généraliser les nouvelles données et ce qui peut aider à protéger contre le sur-apprentissage [36].

- **Le transfer learning** : Au lieu ou on commence notre traitement de zéro, on va utiliser un réseau préformé sur une quantité vraiment énorme de données. Ensuite, on utilise nos données relativement petites pour affiner le modèle pré-entraîné. Le principal avantage de l'apprentissage par transfert est qu'il atténue le problème des données de formation insuffisantes, qui est l'un des raisons du surajustement. Aussi il augmente la productivité et réduit le temps de formation[36].

## 2.7 Les travaux de recherche basés sur l'architecture CNN pour la classification de pneumonie

Dans cette section nous allons présenter quelques travaux de recherche dans le domaine de la classification des différents types de pneumonie. .

Work	Data-set	Moda-lité	Prepro-cessing	DNN	Classifieur	nombre-classes	Performance
2017 [37]	Chest-Xray-14	X-ray	Random Horizontal Flipp Normalizing	CheXNet	Sigmoïde	Binaire (pneumonie, non penu- monie))	F1-Sc= 0.435
2018 [38]	Chest-Xray-14	X-ray	Resizing, Non DA	Chest-net avec (ResNet- 152 Feature Extraction et At- tention branch)	Sigmoïde	Multi-class (14 mala- dies))	AUC=0,781
2020 [39]	COV-IDX	X-ray	DA	COVID-Net	Softmax	Multi-class (Normal, Non- Covid19 , Covid-19)	Acc=93.3 Sen=91.0
2020 [40]	Co-hens Git- Hub	X-ray	Rescaling Non DA	COVIDX- Net(VGG19, Dense- Net201)	Softmax	Binaire( Covid- 19,non Covid-19)	Acc=90 Pre= 0.83 F1-Sc= 0.91

<b>2020</b> [41]	Diffé- rants Data sets	X- ray et CT- Scan	Cropping, Resizing	AlexNet	SoftMax	Binaire (Covid-19, normal)	Acc=98 Sen=100 Spe=96
<b>2020</b> [42]	Diffé- rants Data sets	X- ray	Non défini	ResNet50	SVM	Multi-class (Pneumo- nie, Covid- 19, normale)	Acc=95.33 Sen=95.33 F1-Sc=95.34
<b>2020</b> [43]	Diffé- rants Data sets	X- ray	DA, Rescaling	Coronet (basé sur Xception)	Softmax	Multi-class (bacte- rienne, viral, Covid-19, normale)	Acc=89.5 Pre=90 Spe=96.4 F1-Sc=98
<b>2020</b> [44]	Diffé- rants Data sets	X- ray	DA	Conca- tenation de Xcep- tion et Re- sNet50V2	Softmax	Multi-class (Normal, Covid, Pe- numonie)	Acc=91.4
<b>2020</b> [45]	Clini- cal (3 Da- ta- sets)	X- ray	Reshaping, Normaliza- tion	Multi- Resolution CovXNet	Non défini	Multi-class (bacte- rienne, viral, Covid-19, normale)	Acc=90.2 Pre=90.8 Spe=89.1 F1-Sc =90.4

<b>2021</b> [46]	Diffé- rants Data sets	X- ray	DA Normali- zation	DenResCov- 19 (basé sur DenseNet- 121 et ResNet- 50)	SoftMax	Multi- classes (normal , pneumo- nie, COVID- 19, tuberculose)	Pre=82.90 AUC-ROC=95 F-1 Sc=75.75
<b>2021</b> [47]	COV- IDR	X- ray et CT- Scan	Data shuf- fling	CoroDet	Sigmoid	4-classes : (Covid-19, Normal, Viral, bac- terienne)	4-classes : Acc= 91.2 Sen= 91.76 Spe= 93.48 Pre= 92.04 F-1 Sc= 90.04
<b>2021</b> [48]	COV- ID- CT	CT- Scan	Resizing	CNN à partir de 0	Softmax	Binaire(cov- id, non covid)	Acc=93
<b>2021</b> [49]	Diffé- rants Data sets	X- ray et CT- scan	Resizing	COVID- detection- Net (basé sur Shuf- fleNet et Squeeze- Net)	MSVM	4 classes : (CO- VID,Normal, Bacte- rienne ,Viral)	Acc=94.44 Spe=98.15 Pre=94.42 F1-Sc=94.4

<b>2021</b> <b>[50]</b>	Diffé- rants Data sets	X- ray	Non défini	AlexNet	SoftMax	Binaire(Cov- id-19, normal) 3 classes : (COV ID-19, Bacte- rienne, normal) 4-classes : (Covid-19, viral, Bac- terienne, normal)	2 classes : Acc= 99.16 Sen= 97.44 Spe= 100 3 classes : Acc= 95 Sen= 91.30 Spe= 84.78 4 classes : Acc= 93.42 Sen= 89.18 Spe= 98.92
<b>2021</b> <b>[51]</b>	Diffé- rants Data sets	X- ray	Resizing, DA, Normali- zation	Concate- nation( Efficient- Net, GoogLeNet, Xception)	SoftMax	Binaire(CO- VID-19, normal) Multi- classes : (Covi-d19, pneu- monie, tuber- culose, normal)	2 classes : Acc= 98.95 4 classes : Acc= 99.21

TABLE 2.2 – Travaux réalisés dans la détection des différents types de pneumonie.

La recherche bibliographique réalisée a montré que :

- Un nombre significatif d'articles réalisés dans ce domaine ou la majorité des travaux proposent un modèle de réseau de neurones convolutif profond pour identifier la présence du COVID 19 et le distinguer des autres formes de grippe et de pneumonie.
- Toutes les architectures CNN ont été proposées :
  - Soit avec une approches de transfer learning.
  - Soit des modèles hybridés avec machine learning .
  - Soit avec la construction des modèles a partir de 0.
- Le COVID 19 étant une nouvelle maladie, il n'y a pas de BDD de taille appropriée disponible pouvant être utilisé, les BDD générées sont des dataset des images scanner thoracique (CT) ou radiographies pulmonaires X-Ray, ces BDD sont préparées en combinant et en modifiant différents référentiels de données disponibles publiquement.

## 2.8 Conclusion

Nous avons présenté dans ce chapitre le Deep Learning avec ses différents domaines d'applications et quelques types des réseaux de neurones. Nous avons notamment détaillé l'architecture CNN dans laquelle se base notre travail.

Dans le prochain chapitre, nous allons détailler notre solution.

## Solution proposée

### 3.1 Introduction

Récemment, l'apprentissage profond a révolutionné le diagnostic et la gestion automatique des maladies en analysant, identifiant et classant avec précision les maladies dans les images médicales, la raison de ce succès est que ces algorithmes apprennent automatiquement des fonctionnalités à partir des données elles-mêmes.

Dans ce chapitre nous allons présenter l'ensemble de données utilisé ainsi que l'architecture de nos modèles proposés.

### 3.2 Présentation de données

Dans notre étude ,on a utilisé la base de donn e de l'article [43].

 tant donn e que COVID-19 est une nouvelle maladie, il n'y a pas d'ensemble de donn es de taille appropri e disponible publiquement, Par cons equent, l'auteur a d u cr er un ensemble de donn es en collectant des images radiologiques pulmonaires provenant de deux bases de donn es d'images accessibles au public. Les images radiologique COVID-19 sont disponibles sur un Github open source, r f rentiel par Joseph et al. Les auteurs ont compil  les images radiologiques provenant de diverses sources authentiques (Soci t  de radiologie d'Am rique du Nord (RSNA), Radiopaedia etc) la plupart des  tudes sur le COVID-19 utilisent les images de cette source.[52]

les images normales, bact riennes et pneumonie virale ont  t  obtenues   partir du r f rentiel Kaggle « Chest X-Ray Images (Pneumonia) » [53].

### 3.2.1 Distribution de notre jeux de données

Notre jeu de données est composé de deux sous-ensembles :

**L'ensemble de données d'entraînement** : qui contient globalement 1396 images (300 images COVID-19, 368 images normal, 374 images pneumonie bactérienne et 354 image pneumonie viral)

**L'ensemble de données de teste** : qui contient globalement 242 images (20 images de cas COVID-19, 77 images normal, 75 images pneumonie bactérienne, 70 images de pneumonie viral).

L'ensemble de données d'entraînement a ensuite été divisé en deux ensembles : train et validation comprenant respectivement 80% et 20% des données totales.

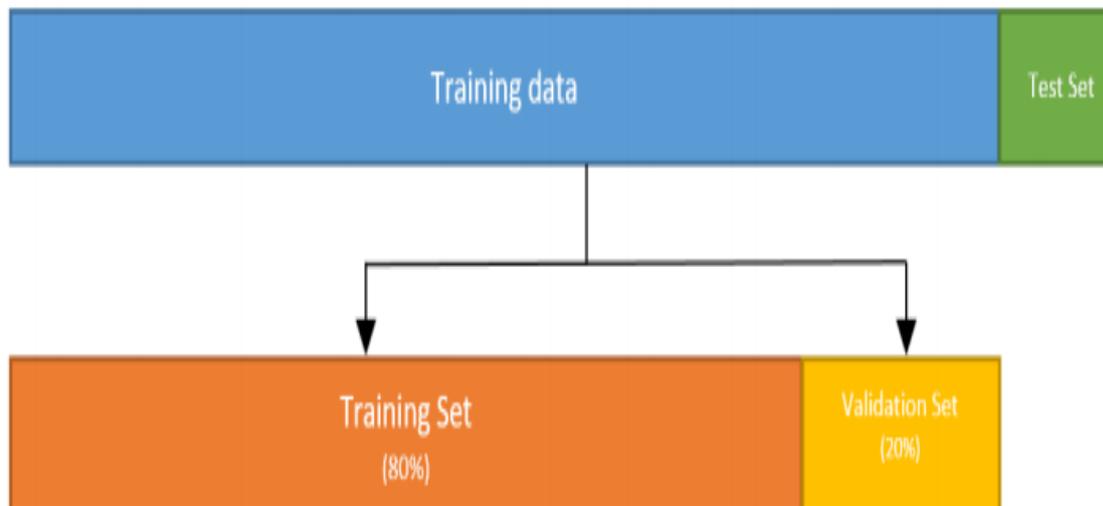


FIGURE 3.1 – Distribution des données.

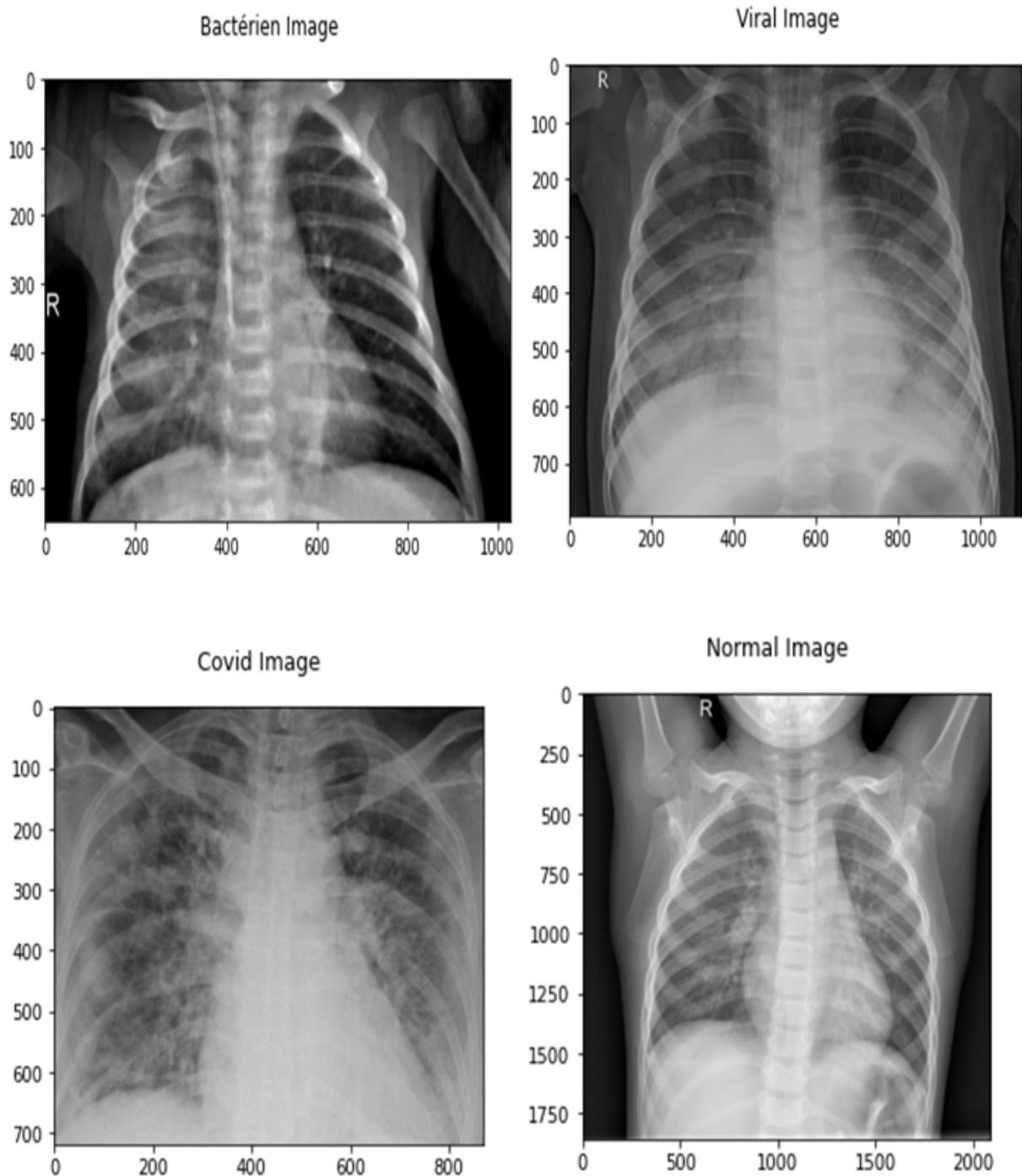


FIGURE 3.2 – Quelques échantillons de notre base de données.

### 3.2.2 Prétraitement de données

#### 3.2.2.1 Meilleure transformation des données pour la modélisation

Il est très important de rogner notre images en gardant que la région qui influence l'étiquette de l'image, (224\*224) donne meilleur résultat que (150\*150) et (96\*96).

### 3.2.2.2 Augmentation de données

La quantité et la qualité de nos données jouent un rôle très important pour l'élaboration d'un bon modèle. Parmi les augmentations qu'on a utiliser :

- Retournement aléatoire horizontal et vertical (random flip).
- Générer des images avec différents niveaux de zoom (zoom\_range).
- Spécifier la limite supérieure de la fraction de la largeur totale par laquelle l'image doit être décalée de manière aléatoire, vers la gauche ou la droite.(width\_shift\_range).
- La normalisation par la dévision sur 255.

## 3.3 Architectures proposées

Dans notre projet, nous allons proposer quatre architectures basées sur le transfer learning, dans chacune d'elles un modèle sera créé et formé. Le transfer learning est largement utilisé dans le domaine de classification des images et cela pour la haute performance des modèles pré-entraînés avec un temps d'entraînement réduit.

Sachant qu'il existe différent stratégies d'appliquer le transfer learning, On a choisir la stratégie Fine-tunning total car c'est la plus adapté a notre problème. Cette stratégie Consiste à remplacer la dernière couche FC du réseau pré-entraîné par un classificateur adapté au nouveau problème, puis entrainer toutes les couches sur les nouvelles images, elle est utilisé lorsque la taille de la nouvelle data est moyenne.[54]

le résumé de nos propositions est le suivant :

### 3.3.1 Les modèles pré-entraînés utilisés

**Modèles DenseNet201 [55] et DenseNet169 [55]** : Sont des réseaux de neurones convolutifs de 201 et 169 couches de profondeur, ces réseaux ont été entraînés sur la base de données ImageNet pour classer ces images en 1000 catégories d'objets, cette dernière est une base de données qui contient 1,2 million d'images et 50 000 images supplémentaires pour la validation et 100 000 images pour le test. En conséquence, ces réseaux ont appris de riches représentations de caractéristiques pour un large éventail d'images. Chaque bloc dense a une connexion avec toutes les couches suivantes.

Pour faciliter à la fois le sous-échantillonnage dans l'architecture et la concaténation des fonctionnalités, les auteurs ont divisé ces réseaux en plusieurs blocs denses densément

connectés.

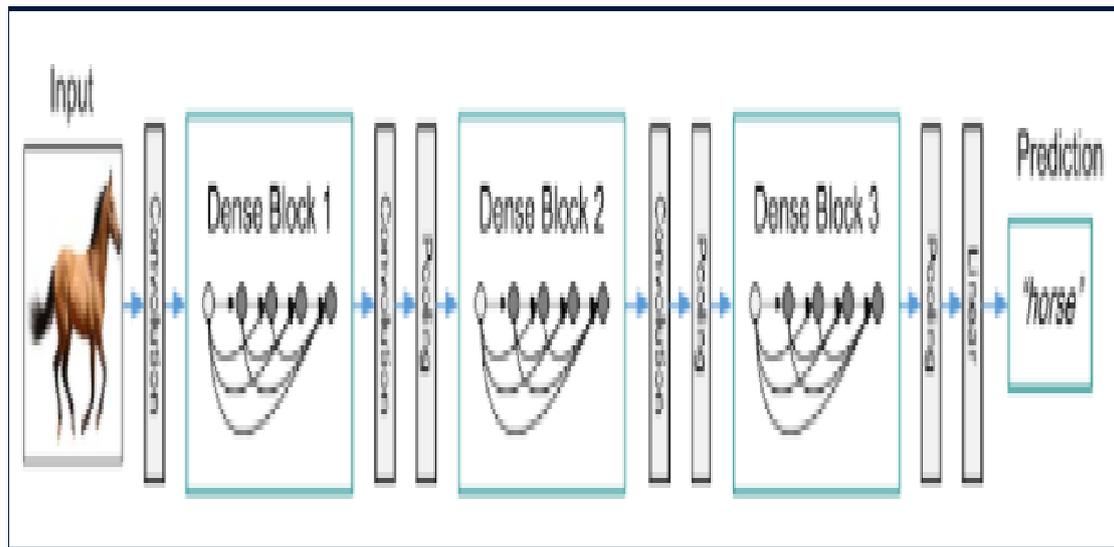


FIGURE 3.3 – Architecture du DenseNet avec trois blocs.[56]

**Le modèle InceptionResNetV2 [57]** : Inception-ResNet-v2 est un réseau de neurones convolutifs de 164 couches de profondeur qui est entraîné sur la base de données ImageNet, c'est un hybride technique qui combine la structure Inception et Residual connections.

**Le modèle Inception V3 [57]** : Inception v3 est un modèle de reconnaissance d'images couramment utilisé qui a démontré sur l'ensemble de données ImageNet, il est constitué de plusieurs couches de convolutions, AveragePooling, MaxPooling, concaténations, Droupout et des couches Dense.

**Modèle VGG16 [58]** :C'est un modèle pré-formé sur ImageNet, il s'agit d'une architecture simple possède 16 couches pondérés qui utilise uniquement des blocs composés d'un nombre incrémentielle de couches de convolution et des blocs de MaxPooling sont intercalés entre les blocs convolutifs.

### 3.3.2 Utilisation de l'architecture DenseNet201

Nous avons créé un modèle CNN basé sur le modèle pré-entraîné DenseNet201.

Tout d'abord la couche d'entrée du modèle reçoit une image (224\*224\*3), l'extraction de caractéristiques se fait aux niveau de DenseNet201.

La sortie de la couche d'extraction de caractéristiques est ensuite transmise à une couche Flatten pour convertir la forme des données en un vecteur unidimensionnel, suivi d'une couche Dropout pour éviter le problème de overfitting, deux couches de Batch Normalization pour optimiser les paramètres du modèle et améliorer la vitesse de l'entraînement.

La partie classification se compose d'une couche Dense avec 100 neurones, la sortie finale est produite à partir d'une couche Dense avec quatre neurones et une fonction d'activation Softmax, qui classe l'image de sortie dans l'une des classes COVID-19, pneumonie bactérienne, pneumonie viral ou normal.

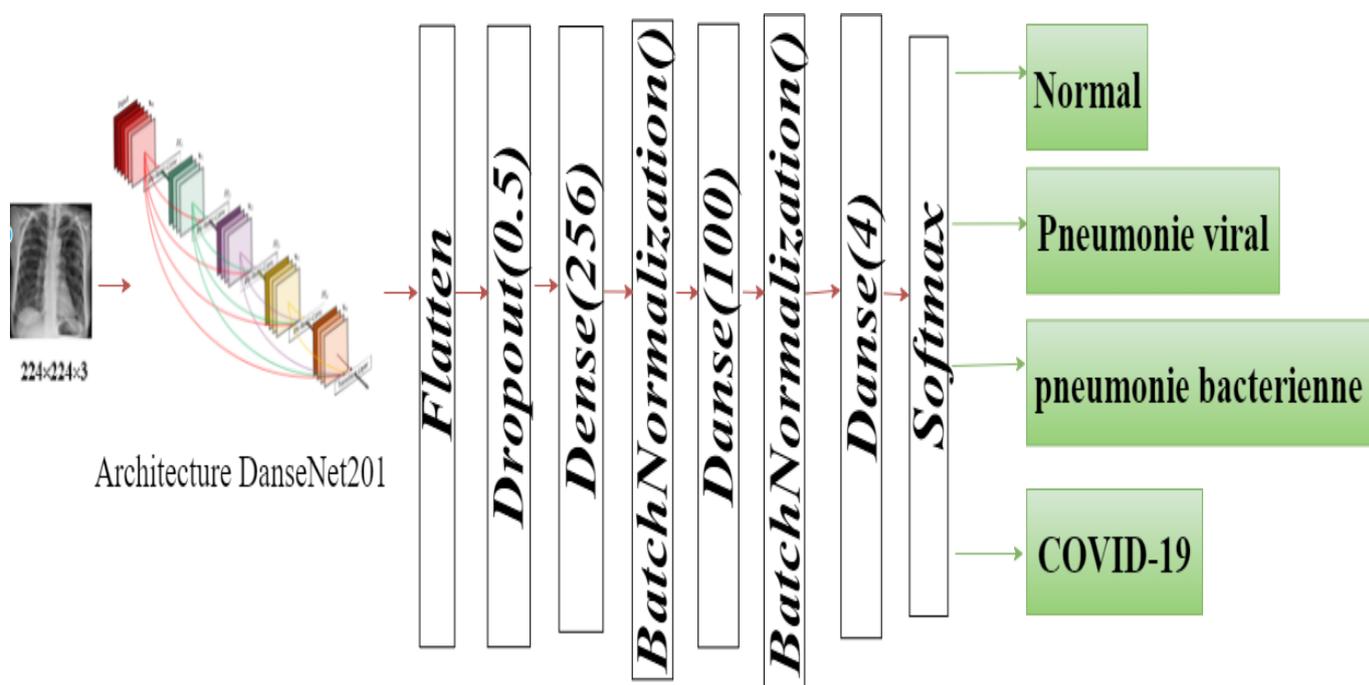


FIGURE 3.4 – Notre architecture basé sur DenseNet201.

### 3.3.3 Utilisation de l'architecture DenseNet169

Premièrement la couche d'entrée du modèle reçoit une image (224\*224\*3), l'extraction de caractéristiques se fait aux niveau de DenseNet169.

La sortie de la couche d'extraction de caractéristiques est ensuite transmise à une couche MaxPooling2D qui sert à prendre le maximum de chaque carte de caractéristiques, suivi par trois couches de Batch Normalization et trois couches Dropout.

La partie classification se compose de trois couches Dense la première avec 256 neu-

rones, la deuxième avec 128 neurones et la sortie finale est produite à partir de la troisième couche avec quatre neurones et une fonction d'activation Softmax, qui classe l'image de sortie dans l'une des maladies pulmonaires ou bien normal.



FIGURE 3.5 – Notre architecture basée sur DenseNet169.

### 3.3.4 Concaténation entre deux architectures InceptionV3 et InceptionResNetV2

Dans cette architecture chacun des modèles InceptionV3 et InceptionResNetV2 génère une carte de caractéristiques dans sa dernière couche d'extraction de caractéristiques à partir de l'image d'entrée, nous avons concaténé ces dernières pour améliorer l'apprentissage.

Ces cartes de caractéristiques sont transmises à deux couches convolutives pour aider le réseau à mieux apprendre les fonctionnalités extraites de InceptionResNetV2 et InceptionV3.

La partie classification se compose de trois couches Dense, la première avec 128 neurones, la deuxième avec 64 neurones et la troisième couche avec quatre neurones et une fonction d'activation Softmax qui est responsable de la classification finale.

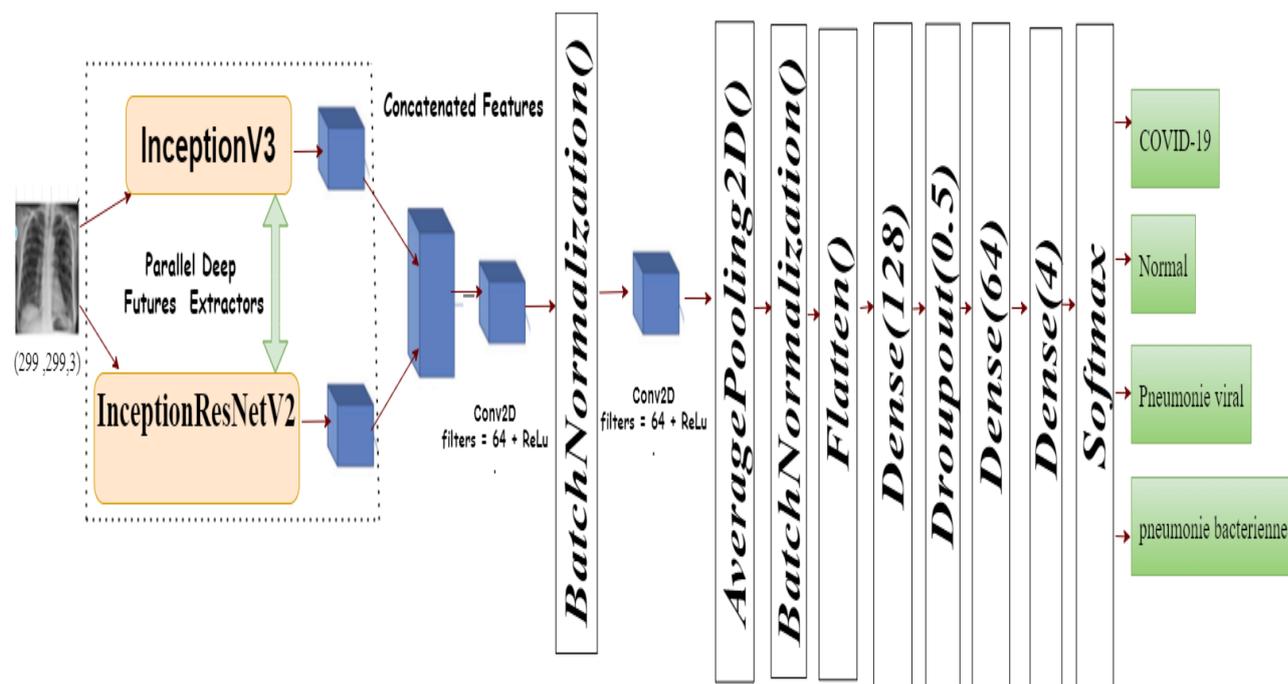


FIGURE 3.6 – Notre architecture basée sur InceptionResNetV2 et InceptionV3.

### 3.3.5 Concaténation entre deux architectures VGG16 et DenseNet169

Le processus de cette architecture est comme suit :

Le modèle VGG16 et DenseNet169 reçoivent en entrée une image de notre jeu de données, l'extraction des caractéristiques se fait en utilisant plusieurs caractéristiques extraites par ces deux réseaux.

Ces cartes de caractéristiques sont transmises à deux couches convolutives pour aider le réseau à mieux apprendre les fonctionnalités extraites de VGG16 et DenseNet169.

La partie classification se compose de trois couches Dense la première avec 128 neurones, la deuxième avec 64 neurones et la troisième couche avec quatre neurones et une fonction d'activation Softmax qui est responsable de la classification finale.

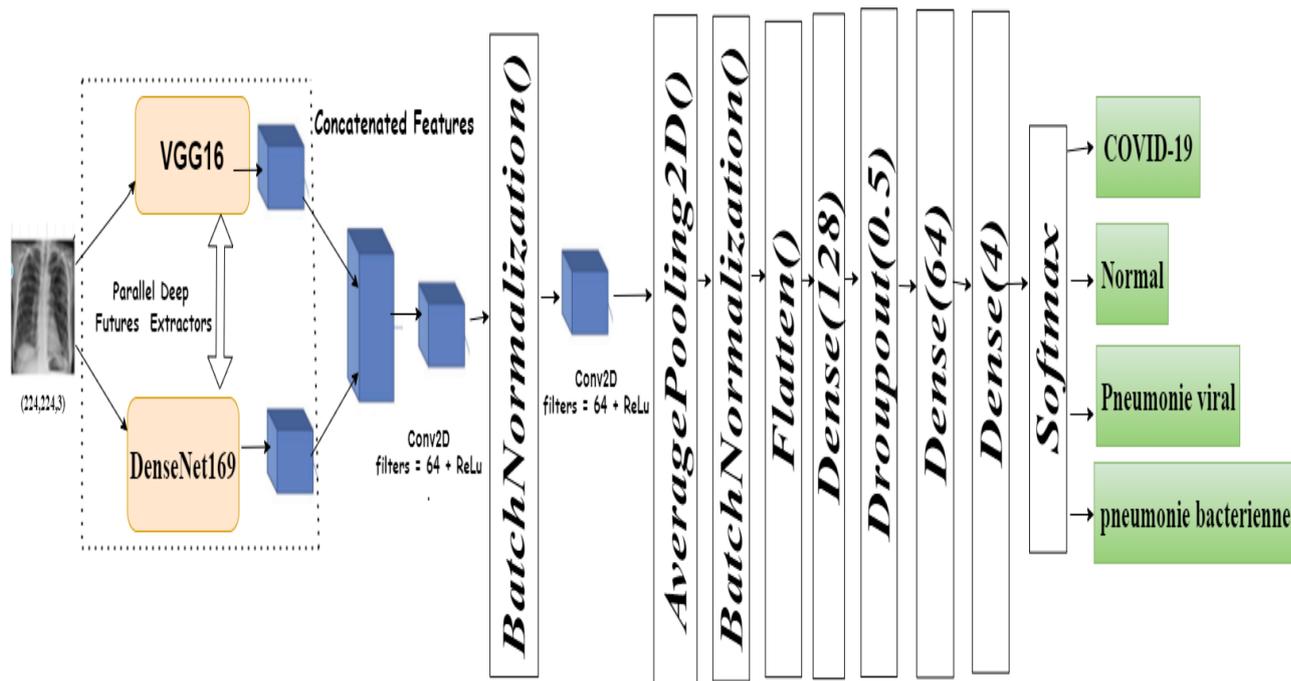


FIGURE 3.7 – Notre architecture basée sur VGG16 et DenseNet169.

### 3.3.6 Entraînement de nos modèles

Pour l'entraînement de nos modèles, nous avons choisi :

**La fonction d'erreur :** La fonction que nous avons utilisé est "catégorical cross-entropy", par ce qu'elle permet la réduction de l'erreur tout en utilisant l'algorithme de la descente de gradient.

**L'algorithme d'optimisation de la descente de gradient :** On a utilisé l'algorithme Adam.

Adam est un algorithme d'optimisation permet de réduire le poids des erreurs, les avantages de l'utilisation d'Adam comme suit :

- Simplicité de mise en œuvre.
- Efficacité du calcul.
- Nécessite peu de mémoire [59].

**Early stopping :** Cette callback permet d'arrêter l'entraînement lorsqu'une métrique surveillé a cessé de s'améliorer. [60]

**ReduceLRonPlateau :** Cette callback sert à réduire learning rate lorsqu'une métrique a cessé de s'améliorer[61].

## **3.4 Conclusion**

Dans ce chapitre, nous avons présenté l'ensemble de données utilisé, les pré-traitements effectués sur ces derniers ,et notamment le détail de notre solution.

Les résultats et les outils de cette implémentation seront présenté dans le chapitre suivant.

## Résultats et discussion

### 4.1 Introduction

Dans ce chapitre, nous allons parler d'une manière générale des différents moyens, techniques utilisés pour implémenter notre solution.

Ensuite nous ajoutons les critères choisis pour évaluer la performance de nos modèles, à la fin, nous allons discuter les résultats obtenus.

### 4.2 Environnement de développement

Afin d'implémenter notre modèle, des différents bibliothèques, environnement de développement et langage de programmation ont été utilisées.

#### 4.2.1 Langage de programmation

Le langage de programmation choisi pour implémenter nos modèles présentés précédemment s'est porté sur le langage python.

Python est un langage de programmation très puissant en plus il est facile à apprendre. Il a une structure de données de haut niveau, il permet d'utiliser des méthodes simples et efficaces, c'est un langage idéal pour les scripts et le développement rapide d'applications dans plusieurs domaines et sur la plupart des plateformes.

Python offre plusieurs librairies (packages) de traitement de données, calculs matriciels, d'analyses et de visualisation de données. Pour le Deep Learning, python possède des environnements de travail comme caffe, tensorflow, keras, pytorch. Ces environne-

ments de travail sont très utiles car ils permettent de gérer facilement l'algorithme de rétro-propagation pour les réseaux de neurones de grande taille, les CNN, RNN, etc. Ils assurent aussi la parallélisation des calculs sur le GPU.[62]

Le choix de ce langage présente les avantages suivants :

- C'est entièrement gratuit.
- Il est orienté objet mais n'impose pas ce type de programmation.
- Il fonctionne sur tous les principaux systèmes d'exploitation et plates-formes informatiques.
- Il crée des logiciels de qualité professionnelle.

## 4.2.2 Plateforme de développement

Tous entrainement du réseau de neurone nécessite une forte puissance du calcul. Notre méthode est basée sur un réseau convolutif profond, ce qui implique un grand nombre de points à entrainer. L'entrainement sur le PC portable aurait également pris beaucoup de temps. Nous avons utilisé le Google Colab.

**Google Colab** Ou Colaboratory est un service cloud, offert par Google (gratuit) basé sur Jupyter Notebook, pour entraîner des modèles de Machine Learning avec une puissance de calcul énorme. en plus qu'il est facile à utiliser, le colab est flexible dans sa configuration.[63]

- Prise en charge de Python 2.7 et Python 3.6.
- GPU gratuit.
- Bibliothèques préinstallées : Toutes les principales bibliothèques Python comme TensorFlow, Numpy, Matplotlib parmi beaucoup d'autres sont préinstallées et prêtes à être importées.
- Permet aux développeurs d'utiliser et de partager des blocs-notes entre eux sans avoir à le faire télécharger, installer ou exécuter autre chose qu'un navigateur.

**Jupyter Notebook** : Jupyter Notebook est une application Web open source qui permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations, et du texte. Les utilisations incluent le traitement et la visualisation des données, la modélisation statistique, l'apprentissage automatique. Elle est disponible par défaut dans la distribution Anaconda.[64]

### 4.2.3 Bibliothèques utilisées

Dans notre travail, nous avons besoin des bibliothèques suivantes :

**TensorFlow** TensorFlow est une plateforme open source de bout en bout (end-to-end) pour le Machine Learning. TensorFlow est un framework de programmation pour le calcul numérique développé par l'équipe Google en novembre 2015.

Depuis, TensorFlow a continué à gagner une grande popularité et devenu un framework très utilisés pour le Deep Learning .

Aujourd'hui, ce framework est utilisé par les grandes entreprises comme Intel, Twitter, Lenovo, Nvidia, etc et les principaux produits de Google sont basés sur TensorFlow : Gmail, Google Photos, Reconnaissance de voix, etc.[65]

TensorFlow est très populaire pour ses nombreux avantages que nous citons ci-dessous :

- Multi-plateformes (Linux, Mac , Windows et même Android ).
- Temps de compilation très courts.
- Exécution efficace d'opérations sur CPU, GPU ou TPU.

**Keras** Keras est une bibliothèque open source pour les réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow. Il a été développé en mettant l'accent sur l'expérimentation rapide. Keras est utilisé dans un grand nombre de startups, de laboratoires de recherche (dont Microsoft Research et la NASA) et de grandes entreprises telles que Netflix, Square, Uber, Google, etc [66].

**Matplotlib** Matplotlib est une bibliothèque de traçage Python 2D qui produit des figures dans différents formats en haute qualité, c'est un environnement interactif sur toutes les plateformes.[67]

**Numpy** Numpy est un package utilisé pour les calculs scientifiques en Python. Il est idéal pour les opérations liées à l'algèbre linéaire, aux transformations de Fourier, ou au crunching de nombres aléatoires. [68]

**Pandas** Pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et facile à utiliser.[69]

## 4.3 Indicateurs de performances de la classification

Afin d'analyser correctement les performances d'un modèle, diverses mesures de performance telles que l'accuracy, le score F1, la précision et le rappel (sensibilité) sont utilisées.

Ces mesures d'évaluation sont particulièrement utilisées pour évaluer les performances d'un modèle de classification.

### 4.3.1 Matrice de confusion

Une matrice de confusion (Confusion Matrix ) est une matrice  $N * N$  permettant de mesurer les performances d'un modèle en vérifiant notamment à quelle fréquence ses prédictions sont exactes par rapport à la réalité dans des problèmes de classification, cela nous donne une vue de la performance de notre modèle de classification et des types d'erreurs qu'il commet [70].

Pour mesurer les performances, il est important de distinguer quatre types d'éléments classés pour la classe souhaité :

- TP(Correct Positive) : classe positive considéré comme positive.
- TN(Correct Negative ) : classe négative considéré comme négative.
- FP(False Positive) : classe négative considéré comme positive.
- FN(False Negative) : classe positive considéré comme négative.

les autres mesures utilisées sont expliquées dans le tableau 4.1

Indicateur	Description	Équation
<b>Accuracy</b>	Pourcentage de prédictions correctes	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
<b>Précision</b>	Représente le rapport entre le nombre d'exemple correctement positif étiqueté et le nombre de tous les positifs étiquetés	$Precision = \frac{TP}{TP + FP}$
<b>Rappel(sensibilité)</b>	Proportion d'éléments bien classés par rapport au nombre d'éléments de la classe à prédire.	$Rappel = \frac{TP}{TP + FN}$
<b>F1 score</b>	Représente la moyenne harmonique entre la précision et le rappel	$MH = \frac{2 * TP}{2 * TP + FN + FP}$

TABLE 4.1 – Métriques d'évaluations de notre modèle.

## 4.4 Implémentation

### 4.4.1 Paramétrage des modèles

Grâce à notre étude approfondie de la technique transfer learning nous avons créé plusieurs modèles, qui sont basés sur les hypèr-paramètres optimisés.

Modèle-1 : Architecture basée sur la combinaison entre VGG16 et DenseNet169.

Modèle-2 : Architecture basée sur DenseNet169.

Modèle-3 : Architecture basée sur DenseNet201.

Modèle-4 : Architecture basée sur la combinaison de Inception-ResNet V2 et InceptionV3.

Le détail de choix des paramètres est représenté dans le tableau 4.2

Modèle	Epoch	BS	Lr	Optimiseur	Input shape	Nombre- de pa-ramètres
Modèle-1	20	16	0.0001	Adam	224,224	Total params = 28,665,284 Trainable params = 28,506,628 Non-trainable params = 158,656
Modèle-2	20	16	0.0001	Adam	224,224	Total params = 16,577,988 Trainable params = 16,385,796 Non-trainable params = 192,192
Modèle-3	40	20	0.0001	Adam	224,224	Total params = 42,434,248 Trainable params = 42,204,480 Non-trainable params = 229,768
Modèle-4	20	16	0.0001	Adam	299,299	Total params = 78,258,244 Trainable params = 78,163,012 Non-trainable params = 95,232

TABLE 4.2 – Paramétrage de nos modèles.

## 4.4.2 Résultats expérimentaux obtenus et discussion

### 4.4.2.1 Phase d'entraînement

Les performances d'entraînement des modèles à l'étude ont été évaluées en termes de paramètres importants, c'est-à-dire training accuracy, validation accuracy, training loss, and validation loss à différentes epoch. Ces paramètres sont calculés pour estimer overfitting et under fitting des modèles entraînés.

La figure 4.1 montre le training et validation accuracy avec leurs loss valeurs pour la classification en quatre classes basées sur les quatre modèles pré-entraînés. Dans laquelle le nombre d'entraînement pour la plupart des modèles pré-entraînés a été mené jusqu'à la 20eme epoch pour éviter l'overfitting, aussi on constate qu'entre ces deux itérations que l'accuracy de l'apprentissage et de la validation augmente avec le nombre d'epoch. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch cela signifie que notre modèle a été bien entraîné.

#### Modèle 1(VGG16+DenseNet169) :

```
Epoch 1/20
69/69 [=====] - 414s 5s/step - loss: 1.2152 - acc: 0.4461 - val_loss: 1.1233 - val_acc: 0.5110

Epoch 20/20
69/69 [=====] - 73s 1s/step - loss: 0.2602 - acc: 0.9130 - val_loss: 0.3817 - val_acc: 0.8676
```

#### Modèle 2 (DenseNet169) :

```
Epoch 1/20
69/69 [=====] - 420s 5s/step - loss: 1.8118 - acc: 0.3228 - val_loss: 1.2737 - val_acc: 0.4154

Epoch 20/20
69/69 [=====] - 55s 789ms/step - loss: 0.4419 - acc: 0.8531 - val_loss: 0.3517 - val_acc: 0.8676
```

#### Modèle 3 (DenseNet201) :

```
Epoch 1/40
55/55 [=====] - 74s 980ms/step - loss: 1.0244 - acc: 0.6233 - val_loss: 2.5916 - val_acc: 0.3654

Epoch 29/40
55/55 [=====] - 50s 908ms/step - loss: 0.1148 - acc: 0.9609 - val_loss: 0.4012 - val_acc: 0.9038
```

#### Modèle 4 (InceptionV3+Inception-ResNetV2) :

```
Epoch 1/20
69/69 [=====] - 203s 2s/step - loss: 1.1584 - acc: 0.5041 - val_loss: 0.8307 - val_acc: 0.6875
Epoch 20/20
69/69 [=====] - 148s 2s/step - loss: 0.1756 - acc: 0.9483 - val_loss: 0.3466 - val_acc: 0.9007
```

FIGURE 4.1 – Un aperçu sur le résultat initiale et final de la phase apprentissage.

Les graphes accuracy et loss obtenus pour le modèle-1, modèle-2, modèle-3 et modèle-4 :

Les figures 4.2 à 4.5 montre les tracés de acuuracy et de perte sur les ensembles d’entraînement et de validation au cours des epochs d’entraînement pour nos quatre modèles.

Ces graphiques illustrent que nos modèles proposés ont été parfaitement entraînés et qu’ils ne sont ni sous-ajustés ni sur-ajustés

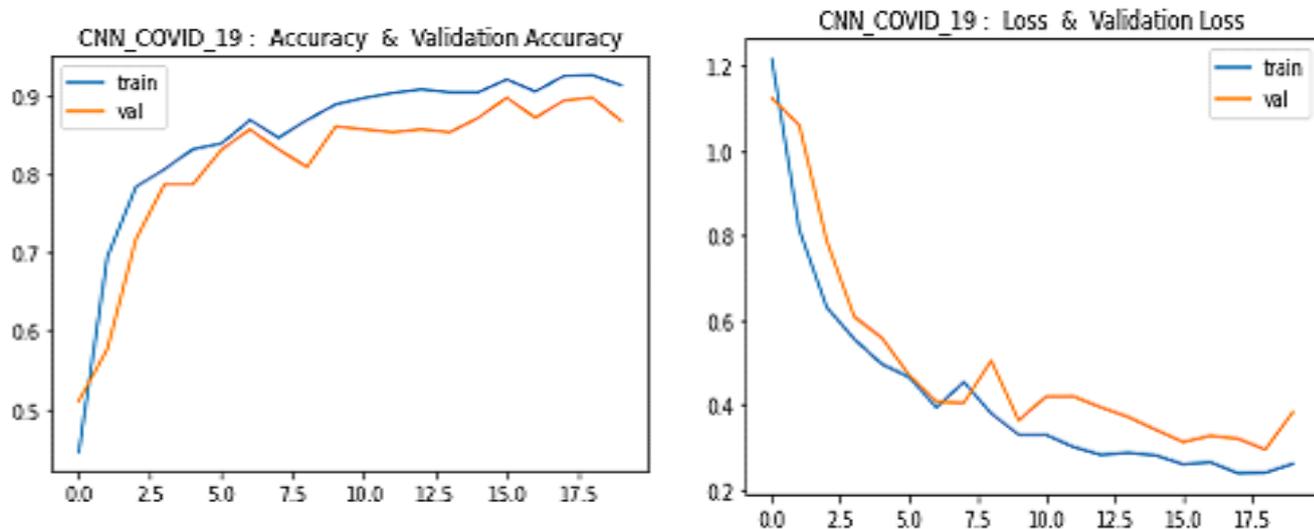


FIGURE 4.2 – Courbes accuracy et loss obtenus pour le modèle-1

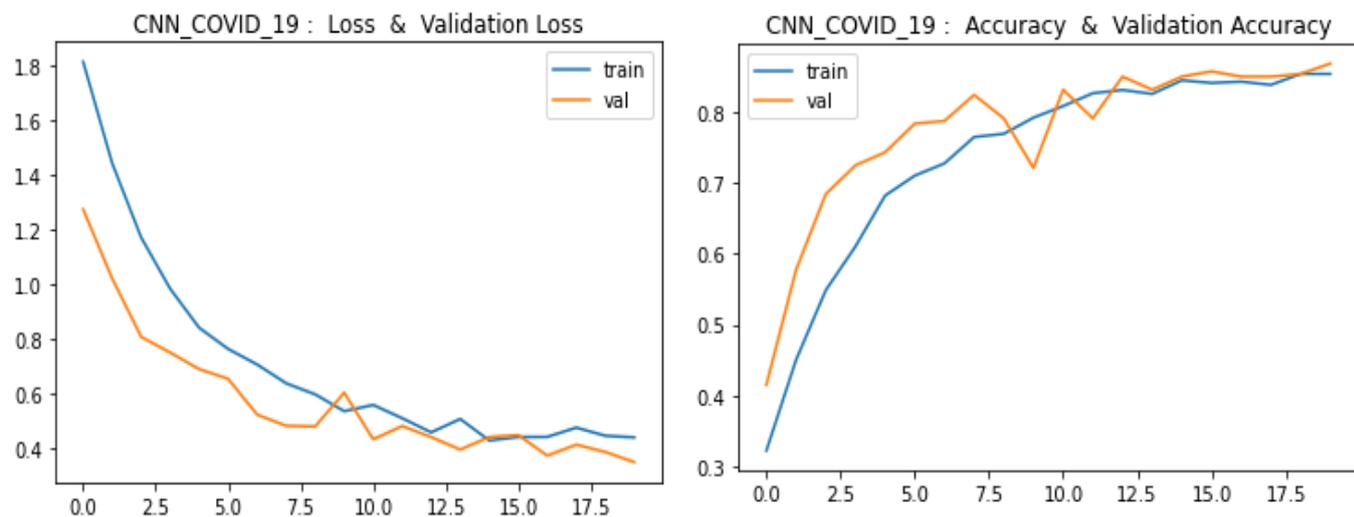


FIGURE 4.3 – Courbes accuracy et loss obtenus pour le modèle-2

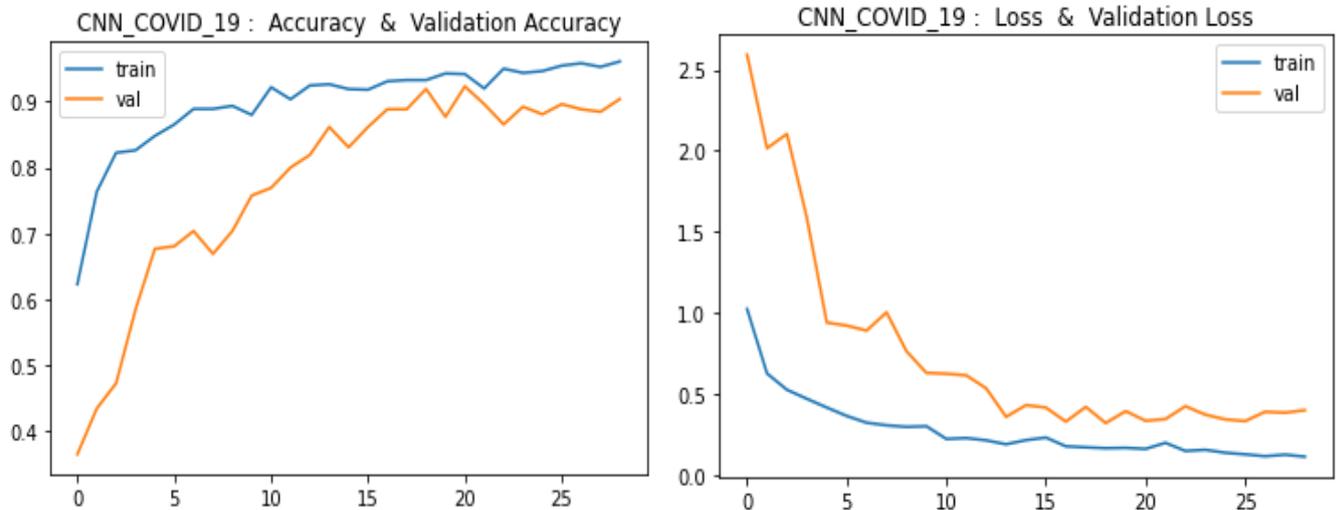


FIGURE 4.4 – Courbes accuracy et loss obtenus pour le modèle-3

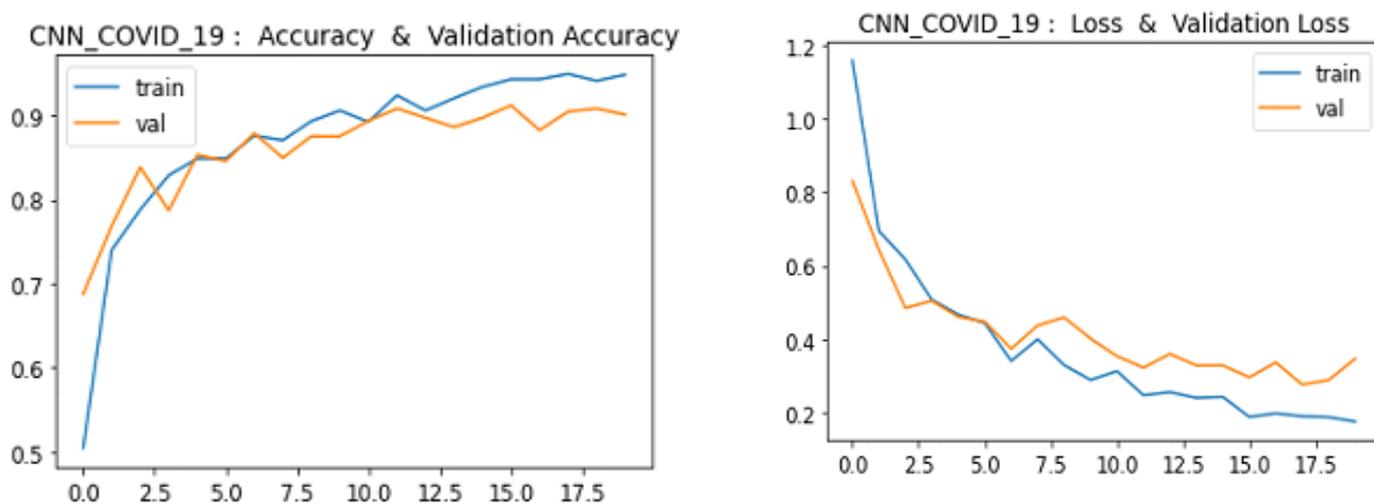


FIGURE 4.5 – Courbes accuracy et loss obtenus pour le modèle-4

#### 4.4.2.2 Phase du teste

##### Matrice de confusion

Les performances de classification multi-classes de nos modèles proposés ont été évaluées en calculant les matrices de confusion pertinentes pour chaque modèle, les résultats ont indiqués que les performances des modèles n'étaient pas loin les uns des autres.

Les résultats de la précision, du rappel, de la mesure F1 pour la classification quatre-classes et les matrices de confusion sont donnés dans les tableaux de 4.3 à 4.6 et les figures de 4.6 à 4.9.

**Modèle-1 :**

D'après la figure 4.6 on remarque que la totalité des images mal classées est de 23 images et les images bien classées est de 219.

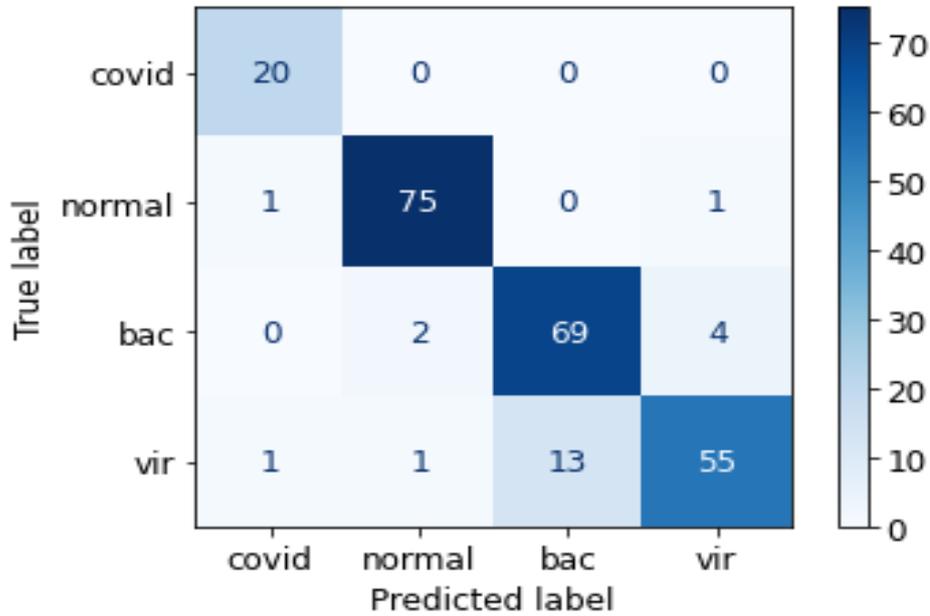


FIGURE 4.6 – Matrice de confusion du modèle-1.

Class	Précision	Recall	F1 score
<b>COVID-19</b>	<b>91%</b>	<b>100%</b>	<b>95%</b>
<b>Normal</b>	<b>96%</b>	<b>97%</b>	<b>97%</b>
<b>Pneumonie bacterienne</b>	<b>84%</b>	<b>92%</b>	<b>88%</b>
<b>Pneumonie vi- ral</b>	<b>92%</b>	<b>79%</b>	<b>85%</b>

TABLE 4.3 – Mesure de performance du modèle-1.

**Modèle-2 :**

D’après la figure 4.7 on remarque que la totalité des images mal classées est de 28 images et les images bien classées 214, et nous remarquons aussi que le premier modèle est mieux que le modèle-2.

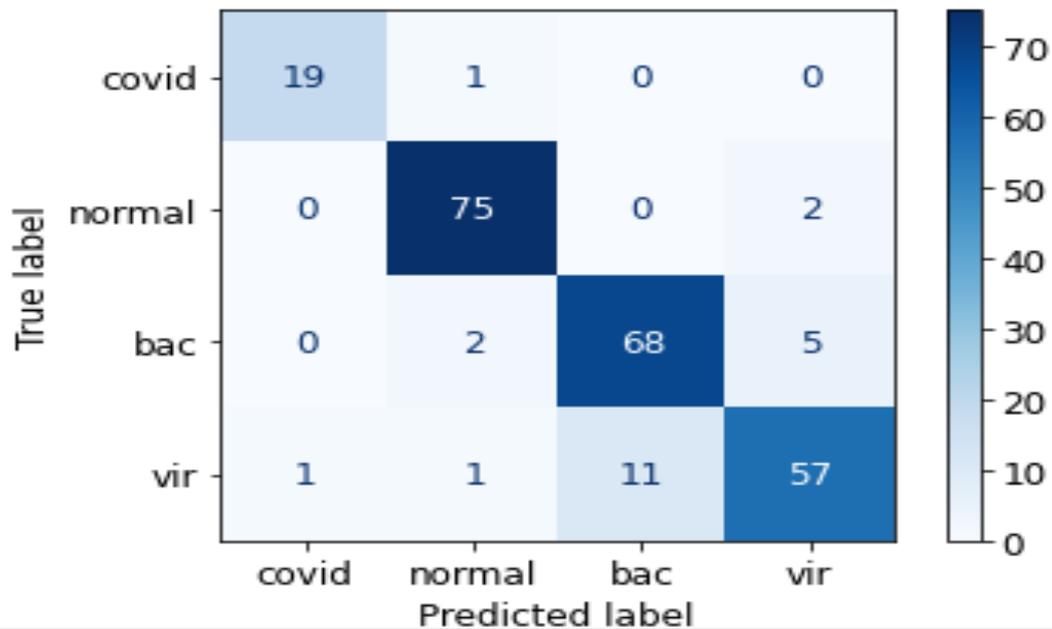


FIGURE 4.7 – Matrice de confusion du modèle-2.

Class	Précision	Recall	F1 score
COVID-19	95%	100%	98%
Normal	88%	99%	93%
Pneumonie bacterienne	86%	87%	86%
Pneumonie vi- ral	90%	76%	82%

TABLE 4.4 – Mesure de performance du modèle-2.

### Modèle-3

D'après la figure 4.8 on remarque que la totalité des images mal classées est de 26 images et les images bien classées est de 216, ainsi que que le troisième modèle est moins performant que les deux précédents modèles, et nous remarquons aussi que le premier modèle est mieux que modèle-2 et le modèle-3.

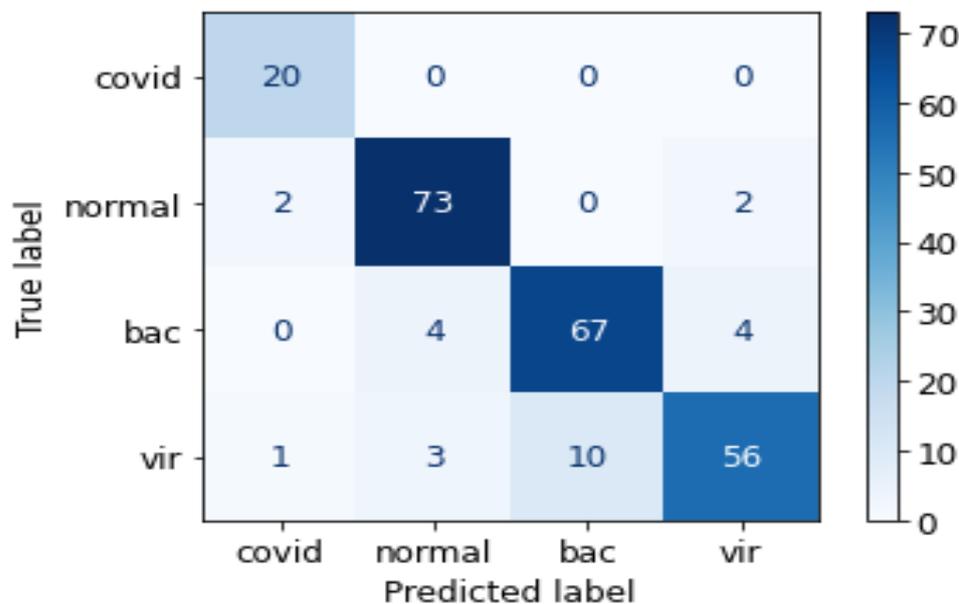


FIGURE 4.8 – Matrice de confusion du modèle-3.

Class	Précision	Recall	F1 score
COVID-19	87%	100%	93%
Normal	91%	95%	93%
Pneumonie bacterienne	87%	89%	88%
Pneumonie vi- ral	90%	80%	85%

TABLE 4.5 – Mesure de performance du modèle-3.

**Modèle-4**

D'après la figure 4.9 on remarque que la totalité des images mal classées est de 27 images et les images bien classées est de 215, ainsi que le quatrième modèle est moins performant que les deux modèles modèle-1 et le modèle-3 et plus performant que modèle-2.

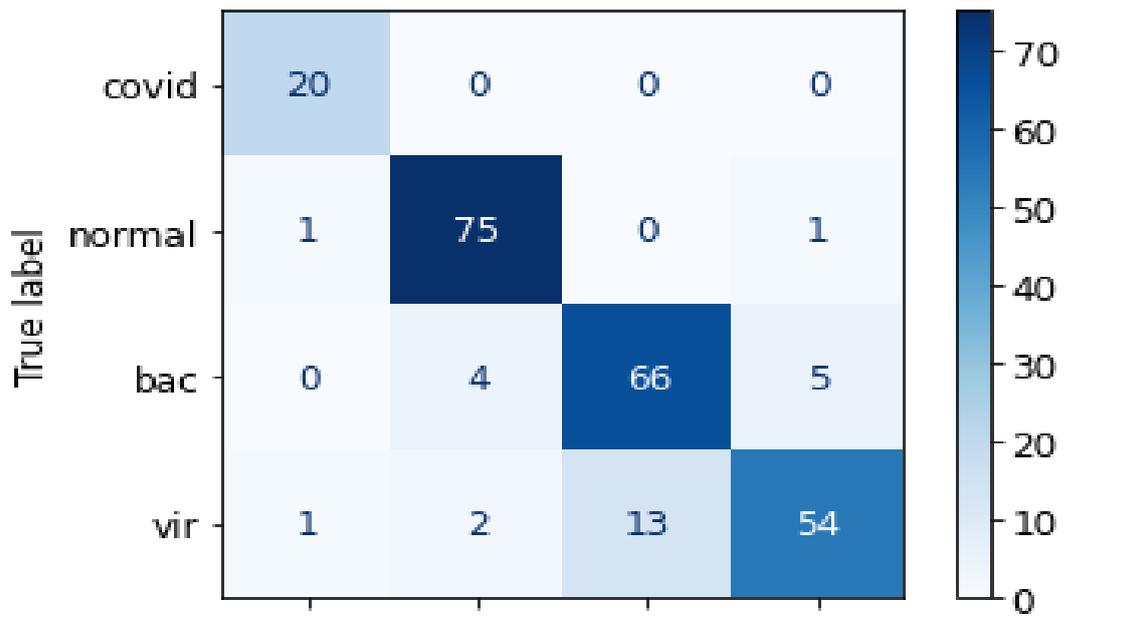


FIGURE 4.9 – Matrice de confusion du modèle-4.

Class	Précision	Recall	F1 score
<b>COVID-19</b>	<b>91%</b>	<b>100%</b>	<b>95%</b>
<b>Normal</b>	<b>93%</b>	<b>97%</b>	<b>95%</b>
<b>Pneumonie bacterienne</b>	<b>84%</b>	<b>88%</b>	<b>86%</b>
<b>Pneumonie vi- ral</b>	<b>90%</b>	<b>77%</b>	<b>83%</b>

TABLE 4.6 – Mesure de performance du modèle-4.

D'une manière générale, en analysant les matrices de confusion, on peut observer que les modèles-1 et le modèle-3 ont atteint une valeur d'accuracy de 90.4% et 89.2% respectivement, même si les modèles ont été entraînés sur un petit ensemble de données, les résultats semblent prometteurs, indiquant qu'avec plus de données, ces deux modèles peuvent obtenir de meilleurs résultats avec un pré-traitement minimum des données.

Une autre observation positive des résultats est que le taux de f1 score pour la classe COVID-19 pour le troisième modèle est de 93% ce qui est généralement bon, concernant le premier modèle est de 95% ce qui est parfait, aussi on prend en considération une valeur de rappel de 100% pour la classe COVID-19 dans le premier modèle qui signifie que tous les 20 exemples de COVID-19 ont été correctement classés avec une précision de 91%. On peut conclure que le premier modèle peut différencier la pneumonie COVID-19 des autres types avec un taux de faux négatifs nul et c'est l'objectif principal de notre travail.

De plus, le f-1 score pour la classe normale de troisième modèle est de 93% qui est identique à la classe COVID-19, concernant le premier modèle est de 97% qui est légèrement supérieure à la classe COVID-19, pour les pneumonies non COVID-19 (pneumonie bactérienne et pneumonie virale) sont comparativement inférieures à celles des deux autres classes dans les deux modèles et contribuent à la précision globale inférieure.

Si on combine la classe pneumonie bactérienne et la classe pneumonie virale en une seule classe en tant que classe non-COVID-19, alors la précision globale augmente considérablement.

L'objectif principal de notre recherche est de réaliser un modèle performant (atteindre un rappel, précision, F1 score élevés) pour la détection de chaque type de pneumonie, Et sur la base de ces résultats, il est évident que malgré la quantité de données de formation limitées, le premier modèle fonctionne bien et réalise ce but attendu, et à mesure que la quantité de données deviennent disponibles, les performances peuvent être encore améliorées.

<b>Modèles</b>	<b>Précision</b>	<b>Recall</b>	<b>F1 score</b>	<b>Teste-accuracy</b>
<b>Modèle-1</b>	<b>91%</b>	<b>92%</b>	<b>91%</b>	<b>90.4%</b>
<b>Modèle-2</b>	<b>90%</b>	<b>90%</b>	<b>90%</b>	<b>88%</b>
<b>Modèle-3</b>	<b>89%</b>	<b>91%</b>	<b>90%</b>	<b>89.2%</b>
<b>Modèle-4</b>	<b>89%</b>	<b>91%</b>	<b>90%</b>	<b>88.8%</b>

TABLE 4.7 – Comparaison entre les 4 modèles .

#### 4.4.2.3 Comparaison de performances entre notre modèle et quelques travaux connexes :

Le COVID-19 est une maladie récente et il n'y a pas beaucoup de littérature et de données disponibles dans le domaine public pour la classification des pneumonies en quatre classes (COVID-19, normale, pneumonie bactérienne, pneumonie viral).

La rareté des données et de la littérature dans le domaine public a limité notre comparaison à quelques techniques.

Works	Teste-accuracy	Precision	Recall	F1-Score
CNN algorithm avec transfer learning (Basé sur VGG16) [71]	88%	Non défini	91%	Non défini
COVID-Net [39]	83.5%	67.0%	81.9%	73.7%
CoroNet [43]	89,5%	90%	82%	86%
Stacked Multi-Resolution CovXNet[45]	90.2%	90.8%	89.9%	90.4%
modèle basé sur Mobile Net[72]	80.95%	82.52 %	82.20%	82.23%
modèle basé sur GoogleNet[73]	80.56%	84.17 %	80.56%	82.32%
Notre modèle proposé ( basé sur VGG16+Dense-Net169)	90,2%	91 %	92%	91%

TABLE 4.8 – Comparaison entre notre modèle proposé et quelques travaux de littérature.

Notre modèle a été entraîné sur un ensemble de données avec un nombre d'exemples de COVID-19 similaire à CoroNet, CovXNet et légèrement plus élevé à Covid-Net, MobileNet, GoogleNet .

Aussi Covid-Net, avec 116 millions de paramètres a atteint une accuracy de 83,5% , CoroNet a atteint une accuracy de 89,5% avec 33 millions de paramètres, notre modèle proposé a atteint une accuracy de 90.2% avec seulement 28 millions de paramètres ce qui représente une énorme capacité de calcul.

On remarque que notre modèle proposé a obtenu de meilleurs résultats en terme de

accuracy 90.2 % et de puissance, ce qui représente une énorme amélioration en terme de calcul et performance .

La figure 4.10 montre le résultat de notre modèle proposé sur quelques exemples d'images de l'ensemble de test.

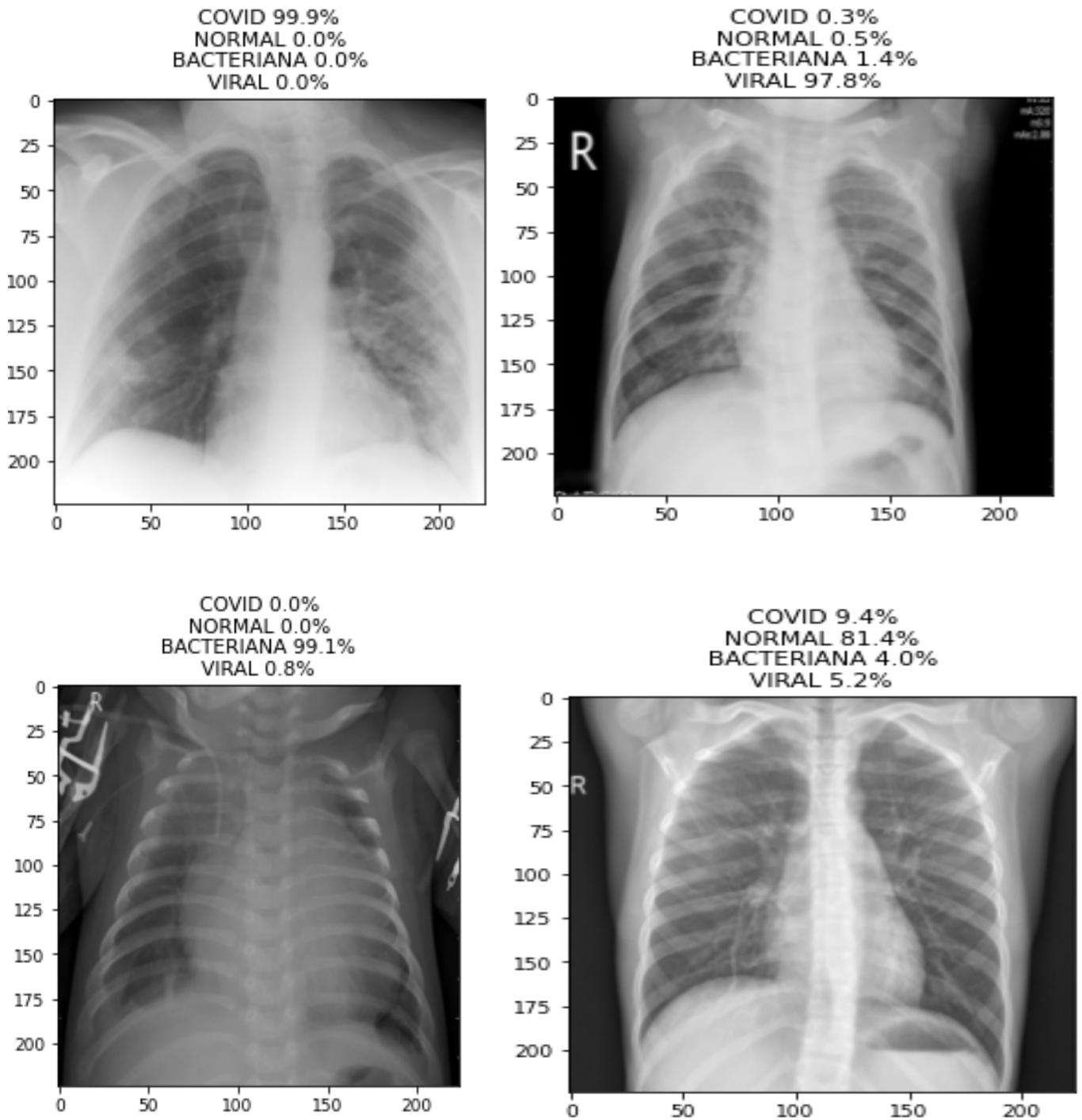


FIGURE 4.10 – Résultat de teste de notre modèle sur quelques exemple de teste.

#### 4.4.2.4 Classification Accuracy de Dataset-1 et Dataset-2

Pour vérifier la robustesse, nous avons testé notre modèle proposé sur d'autres deux datasets préparés par le même ensemble de données utilisé dans notre modèle proposé.

**Dataset-1 :** On a gardé seulement les données de la classe pneumonie bactérienne, pneumonie COVID-19, normal cela pour le train et le test.

**Dataset-2 :** On a combiné les classes de la Pneumonie bactérienne et virale et normal dans une classe nommée non-covid et on les a comparé à la classe COVID-19 pour le train et le test.

Après une légère modification et le ré-entraînement de notre modèle proposé, ce dernier a atteint une accuracy globale de 93.6% pour la classification de 3 classes et de 97.9% pour la classification binaire.

Les résultats sont illustrés dans les matrices de confusion de la figure 4.11.

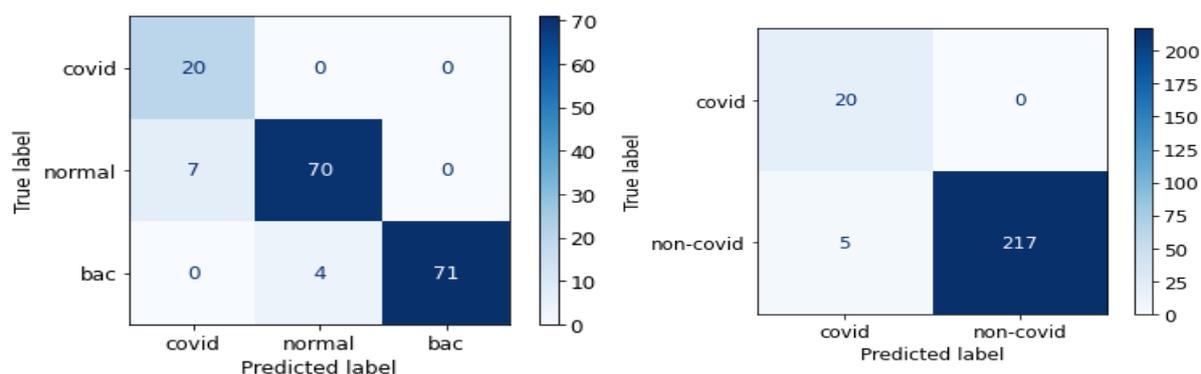


FIGURE 4.11 – Matrices de confusion de la classification binaire et trois classes .

## 4.5 obstacles

Il est connu que la mise en œuvre d'algorithmes d'apprentissage profond nécessite des ressources telles que de grands ensembles de données et de solides ressources matérielles informatiques (CPU, GPU et espace de stockage) pour former des modèles.

Au cours de la mise en œuvre de notre projet, nous avons rencontré de nombreux problèmes en raison de nos ressources en tant qu'étudiant nous citons :

- Certains ensembles de données (pneumonie virale, bactérienne, COVID-19, normal) ne sont pas disponibles publiquement, et certains ne sont pas fiables.

Google Colab était le service le plus approprié pour mettre en œuvre notre projet, mais toujours avec limitations et à partir d'eux nous mentionnons :

- Seulement 12 heures de travail continu par session.
- Utilisation d'un seul GPU par session.

Colab a également besoin d'une connexion internet stable comme tous les services cloud mais malheureusement internet en Algérie est instable, ce qui rend la mise en œuvre de notre projet encore plus difficile.

## 4.6 Conclusion

Nous avons présenté dans ce dernier chapitre les différents langages et outils de développement que nous avons utilisé afin d'implémenter notre solution. Ensuite, nous avons présenté les indicateurs de performances de classification, Finalement nous avons effectué une comparaison entre notre modèle et d'autres modèles de littérature. Nous avons terminé ce chapitre avec un point important, à savoir les obstacles et les problèmes auxquels nous avons été confrontés au cours de notre mise en œuvre du projet.

# Conclusion générale

La classification d'images est une tâche importante dans le domaine de la vision par ordinateur, la reconnaissance d'objets et l'apprentissage automatique.

Pendant la pandémie mondiale de COVID-19 et l'énorme augmentation du nombre de personnes infectées en raison de sa transmission rapide. La seule solution pour lutter contre ce virus est de différencier cette pneumonie des autres types, et mettre les personnes infectés sous surveillance.

Puisque COVID-19 est une pandémie mondiale, la disponibilité des kits de test pose un autre problème même pour les pays développés comme Les états-unis d'Amérique. Pour cela, les chercheurs ont essayé de trouver des solutions alternatives comme la détection des différents types de pneumonies incluant le COVID-19 à partir des images radiologique à travers les anomalies qui présentent chaque types.

C'est dans ce cadre que s'inscrit notre travail, nous avons proposé une architecture pour différencier la pneumonie COVID-19 des autres types (classification quatre classes), notre solution est basé sur le deep learning et la méthode de classification CNN, cela est justifié par la simplicité et l'efficacité de la méthode.

Pour réaliser cette classification (pneumonie viral, pneumonie bactérienne, pneumonie COVID-19, personnes sains), nous avons construit quatre modèles (modèle CNN basé sur l'architecture DanseNet201 et modèle CNN basé sur l'architecture DanseNet 169 et un autre basé sur la concaténation entre deux architectures InceptionV3 et InceptionRes-

NetV2, et le dernier est basé sur la concaténation entre le modèle VGG16 et DanseNet169 ), tous ces modèles ont été entraînés sur une partie de notre ensemble de données des images radiologique afin d'obtenir des meilleurs résultats en terme Accuracy et loss.

Notre modèle proposé basé sur la concaténation des deux architectures VGG16 et DanseNet 169 atteindre des meilleurs résultat par rapport aux d'autres modèles en terme de teste accuracy avec 90.2% ,93.6% et 97.9% pour quatre classe, trois classe et deux classe respectivement.

Les résultats obtenu lors de la phase du test confirme l'efficacité de notre solution.

Dans ce projet, nous avons présenté des généralités sur la pneumonie. Après cela, nous avons parlé d'IA, ML, DL et les réseaux de neurones convolutionnels. Ensuite, nous avons détaillé notre solution et discuté les résultats obtenus en implémentant et en testant nos modèles.

#### **Perspective et travaux futurs :**

Comme perspectives de recherches futures, nous envisageons de :

- Comparer et/ou hybrider notre modèle avec d'autres méthodes de classification.
- Tester notre modèle sur d'autre jeux de données qui contient plus de donnés.
- Proposer d'autres architectures pour améliorer les résultats.

# Bibliographie

- [1] <https://www.passeportsante.net/fr/Maux/Problemes/Fiche.aspx?doc=pneumonie-pm>, consulté le 01/04/2021.
- [2] <https://www.futura-sciences.com/sante/definitions/medecine-bacterie-101>, consulté le 15/03/2021.
- [3] Gladys kostyrka, « La place des virus dans le monde vivant », Thèse du doctorat en Philosophie, L'université de paris , 2010.
- [4] <https://www.aquaportail.com/definition-5597-levure.html>, consulté le : 15/03/2021.
- [5] <https://www.docteurcliv.com/maladie/parasites.aspx>, consulté le :15/03/2021.
- [6] <https://www.doctissimo.fr/sante/maladie-pulmonaire/pneumonie>, consulté le :15/03/2021.
- [7] N.Zhu , D.Zhang , W.Wang ,X. Li , B.Yang , J.Song , A novel Coronavirus from patients with pneumonia in China, 2019.
- [8] <https://www.apsam.com/theme/risques-biologiques/covid-19-coronavirus/transmission-de-la-covid-19?>, consulté le :14/04/2021.
- [9] [https://ed414-openlab.unistra.fr/les-tp/adn-et-genetique-2009-2012/pour-preparer-le-tp/la-pcr-quest-ce-que-cest/?fbclid=IwAR1bDPAFuAfuuZWpGEnuODhuq6XdsFXqF2fM7ZN9rEH9A2sL25fLu\\_YgR3k](https://ed414-openlab.unistra.fr/les-tp/adn-et-genetique-2009-2012/pour-preparer-le-tp/la-pcr-quest-ce-que-cest/?fbclid=IwAR1bDPAFuAfuuZWpGEnuODhuq6XdsFXqF2fM7ZN9rEH9A2sL25fLu_YgR3k), consulté le :14/04/2021.
- [10] <https://www.passeportsante.net/fr/specialites-medicales/Fiche.aspx?doc=serologie>, consulté le :14/04/2021.

- 
- [11] ZEHOR OUKSILI, Compensation du mouvement respiratoire dans les images TEP/TDM ,Thèse du doctorat en informatique, L'UNIVERSIT DE TOULOUSE,2010.
- [12] J.Christophe, Imagerie médicale : maîtriser les expositions aux rayonnements ionisants,Revue de l'ASN,91, 192,p8,2011.
- [13] V.Guillaume, Approche diagnostique des pneumonies aiguës communautaires, Séminaire DES Socle Pneumologie, Médecine Intensive Réanimation, Hôpital Tenon, Sorbonne Université, Paris, 2020.
- [14] F.Chollet, DEEP LEARNING WITH PYTHON,Manning Publication, 2017, page 4.
- [15] <https://www.demos.fr/sites/france.dev/files/resources/support-webinar-ia.pdf>, consulter le 27/février/2021
- [16] Y.LeCun , Y.Bengio,G. Hinton, Deep learning, Nature Deeplearning, 2015 May 28.
- [17] H. Wang, B. Raj, P. Xing, On the origin of deeplearning, arXiv preprint arXiv :1702.07800, 2017.
- [18] Khoulood DAHMANE, Analyse d'images par méthode de Deep Learning appliquée au contexte routier en conditions météorologiques dégradées, Hal, France,2020.
- [19] Karim Aderghal. Classification of multimodal MRI images using Deep Learning : Application to the diagnosis of Alzheimer's disease, Hal, France,2021.
- [20] DIALLO, Nene Adama Dian. La reconnadissance des expressions faciales. Master en Informatique. Université de 8 Mai 1945 – Guelma Juillet 2019.
- [21] Y. Bengio, A. Courville, and P. Vincent, Representation learning : A review and new perspectives,IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI),38 :1798–1828,2013.
- [22] <https://datascientest.com/convolutional-neural-network>, consulté le 2/mars/2021.
- [23] P. Buysens, A. Elmoataz, Réseaux de neurones convolutionnels multi-échelle pour la classification cellulaire,Hal, France, 2016.
- [24] <https://fr.mathworks.com/discovery/convolutional-neural-network-matlab.html>, consulté le 2/mars/2021.
- [25] <https://towardsdatascience.com/building-a-convolutional-neural-network-male-vs-female-50347e2fa88b>, consulter le 3/mars/2021

- [26] <https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/>, consulté le 2/mars/2021.
- [27] <https://www.datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs>, consulté le 3/mars/2021.
- [28] NacerFoued. Reconnaissance d'expression faciale à partir d'un visage reel. Master en Informatique. Université de 8 Mai 1945 – Guelma - Juillet 2019.
- [29] Dr. Merzougui. Deeplearning. Réseau de neurones convolutionnel CNN, pages 4, Université batan2.
- [30] <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/>, consulté le 8/mars/2021.
- [31] <https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63>, consulté le 8/mars/2021.
- [32] <https://developersbreach.com/convolution-neural-network-deep-learning/>, consulté le 8/mars/2021.
- [33] Mouale kDjelloul Youcef, Deep Learning pour la classification des images, Master en informatique, Université Abou Bakr Belkaid Tlemcen,2016.
- [34] <https://larevueia.fr/7-methodes-pour-eviter-loverfitting>,consulté le 20/avril/2021.
- [35] <https://allcloud.io/blog/how-to-solve-underfitting-and-overfitting-data-models/>,consulté le 20/avril/2021.
- [36] <https://www.kdnuggets.com/2019/12/fighting-overfitting-deep-learning.html>, consulté le 20/avril/2021.
- [37] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M.P. Lungren, Chexnet : radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv preprint arXiv :1711.05225. 25 Dec 2017.
- [38] H. Wang, Y. Xia. Chestnet : A deep neural network for classification of thoracic diseases on chest radiography. arXiv preprint arXiv :1807.03058. 9 Jul 2018
- [39] L. Wang, A. Wong, COVID-Net : A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images, arXiv preprint arXiv :2003.09871, 11 May 2020.

- [40] E.E. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net : a framework of deep learning classifiers to diagnose covid-19 in X-ray images, arXiv preprint arXiv :2003.11055, 24 Mar 2020.
- [41] H. S. Maghdid, A. T. Asaad, K. Z. Ghafour, A. S. Sadiq, and M. K. Khan, Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms, arXiv preprint arXiv :2004.00038, 2020.
- [42] P. K. Sethy, S. K. Behera, P. K. Ratha, and P. Biswas, Detection of coronavirus disease (covid-19) based on deep features and support vector machine, 2020.
- [43] A. I. Khan, J. L. Shah, M. M. Bhat, Coronet : A deep neural network for detection and diagnosis of covid-19 from chest x-ray images, Computer Methods and Programs in Biomedicine, p. 105581, 2020.
- [44] M. Rahimzadeh , A. Attar, A modified deep convolutional neural network for detecting covid-19 and pneumonia from chest x-ray images based on the concatenation of xception and resnet50v2, Informatics in Medicine Unlocked, p. 100360, 2020.
- [45] T.Mahmud ,MA. Rahman , SA.Fattah,CovXNet : A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization,Comput Biol Med,2020 Jul.
- [46] M.Mamalakis, A. J. Swift, B.Vorselaars, S.Ray, S.Weeks, W.Ding, R.H. Clayton, L.S. Mackenzie, A.Banerjee , DenResCov-19 : A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays ,arXiv :2104.04006,2021.
- [47] E.Hussain ,M. Hasan M, MA.Rahman ,I. Lee ,T. Tamanna , MZ.Parvez, CoroDet : A deep learning based classification for COVID-19 detection using chest X-ray images, Chaos Solitons Fractals, 2021 Jan.
- [48] AH.Panahi ,A. Rafiei , A.Rezaee , FCOD : Fast COVID-19 Detector based on deep learning techniques, Inform Med Unlocked, 2021.
- [49] AS.Elkorany , ZF.Elsharkawy , COVIDetection-Net : A tailored COVID-19 detection from chest radiography images using deep learning, Optik (Stuttg), 2021.
- [50] A.Ibrahim, M.Ozsoz, S.Serte,al, Pneumonia Classification Using Deep Learning from Chest X-ray Images During COVID-19,Cogn Comput ,2021.

- 
- [51] N.Kumar , M. Gupta, D.Gupta , S.Tiwari ,Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images. J Ambient Intell Humaniz Comput. 2021.
- [52] J.P. Cohen, P. Morrison and L. Dao, COVID-19 image data collection, arXiv :2003.11597, 2020 <https://github.com/ieee8023/covid-chestxray-dataset>.
- [53] Chest X-ray images (pneumonia). <https://www.kaggle.com/>
- [54] <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning>, consulté le 30/06/2021.
- [55] G.Huang, Z.Liu, L. Maaten, K. Q. Weinberger, Densely Connected Convolutional Networks, arXiv preprint arXiv :1608.06993.2016
- [56] <https://amaarora.github.io/2020/08/02/densenets.html>, consulté le 02/08/2020.
- [57] C.Szegedy, S.Ioffe, V.Vanhoucke, A.Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv preprint arXiv :1602.07261,2016
- [58] K.Simonyan, A.Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv :1409.1556 ,2015.
- [59] D.P, Kingma and J. Ba, Adam : A method for stochastic optimization, arXiv :1412.6980, 2014.
- [60] [https://Keras.io/api/callbacks/early/\\_stopping/](https://Keras.io/api/callbacks/early/_stopping/), consulté le 25/08/2021.
- [61] [https://Keras.io/api/callbacks/reduce\\_lr\\_on\\_plateau/](https://Keras.io/api/callbacks/reduce_lr_on_plateau/), consulté le 25/08/2021.
- [62] <https://docs.python.org/fr/3/tutorial/>, consulté le 20/08/2021.
- [63] <https://ledatascientist.com/google-colab-le-guide-ultime/>, consulté le 20/08/2021.
- [64] <https://jupyter.org/>, consulté 28/08/2021.
- [65] <https://www.tensorflow.org/>, consulté le 20/08/2021.
- [66] <https://keras.io/>, consulté le 20/08/2021.
- [67] <https://matplotlib.org/>, consulté le 20/08/2021.
- [68] <https://numpy.pydata.org/>, consulté le 20/08/2021.
- [69] <https://pandas.pydata.org/>, consulté le 20/08/2021.
- [70] <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/>, consulté 28/08/2021.

- [71] S.Kikkisetti, J.Zhu , B.Shen , H.Li and T.Q. Duong ,Deep-learning convolutional neural networks with transfer learning accurately classify COVID-19 lung infection on portable chest radiographs, PeerJ.2020.
- [72] I. Razzak, S. Naz, A. Rehman, A. Khan, and A. Zaib, Improving coronavirus (covid-19) diagnosis in gdeep transfer learning,” medRxiv, 2020
- [73] M. Loey, F. Smarandache, and N.E.M.Khalifa, Within the lack of chest covid-19 x-ray dataset : A noveldetection model based on gan and deeptransferlearning, Symmetry, vol. 12, no.4, p.651,2020.Pro

## Fragments de code

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range=50,
                                   featurewise_std_normalization = True,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.25,
                                   zoom_range=0.1,
                                   channel_shift_range = 20,
                                   horizontal_flip = True ,
                                   vertical_flip = True ,
                                   validation_split = 0.2,
                                   fill_mode='constant')
```

FIGURE A.1 – L'augmentation de données.

```
feature = Flatten(name='flatten')(bch)
pred = Dense(128, activation='relu')(feature)
dp = Dropout(0.5)(pred) #add dropout
pre = Dense(64, activation='relu')(dp)
preds = Dense(4, activation='softmax')(pre)
```

FIGURE A.2 – Les couches de la partie de la classification.