

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABOU BEKR BELKAID  
FACULTÉ DE TECHNOLOGIE  
DÉPARTEMENT DE GÉNIE BIOMÉDICAL



## THÈSE

Pour obtenir le grade de  
**DOCTORAT 3 ÈME CYCLE EN GÉNIE BIOMÉDICAL**  
Spécialité : **Télémédecine et Informatique Biomédicale**

Présenté et soutenu publiquement  
par

**LAHSAINI Ilyas**

le 11 Novembre 2023

Titre:

# **Vers Un Traitement Intelligent Et Transparent Des Données Médicales**

## Jury

Président du jury. Pr. HADJ SLIMANE Zine-eddine,	Pr Université de Tlemcen
Examineurs. Pr. ATMANI Baghdad,	Pr Université de Mostaganem
Examineurs. Pr. BOUKLI HACENE Ismail,	Pr Université de Tlemcen
Examineurs. Dr. BENCHAIIB Yasmine,	MCA Université de Tlemcen
Invités d'honneur. Dr. EL HABIB DAHO Mostafa,	MCA Université de Tlemcen
Invités d'honneur. Pr. BENABDELLAH Mohammed,	Professeur en retraite
Encadreur. Pr. CHIKH Mohamed El Amine,	Professeur Université Tlemcen



# Remerciements

L'expérience de mon doctorat s'est étalée sur cinq années de recherche scientifique et de développement personnel, m'enseignant l'importance d'une attitude toujours positive, de la patience, de la persévérance et de l'effort assidu pour atteindre mes objectifs.

Je tiens à exprimer ma profonde gratitude envers les examinateurs de ma thèse. Leur expertise, leur soutien, et leurs commentaires constructifs ont été d'une valeur inestimable pour l'avancement de mes recherches. Leur engagement envers l'excellence académique et leur dévouement à évaluer mon travail avec rigueur et objectivité ont été une source d'inspiration tout au long de ce processus. Je suis reconnaissant du temps qu'ils ont consacré à l'évaluation de ma thèse, ainsi que de leurs suggestions et recommandations qui ont contribué à améliorer sa qualité et à développer mes compétences en recherche.

Ce travail n'aurait pu se réaliser dans les meilleures conditions sans l'aide précieuse de mes encadreurs, le Professeur Mohamed Amine CHIKH et le Docteur Mostafa EL HABIB DAHO. Je tiens à les remercier pour leur guidance tout au long de la recherche, depuis la sélection du titre jusqu'à l'obtention des résultats. Leurs immenses connaissances, leur motivation et leur patience ont renforcé ma détermination à exceller dans la rédaction de recherches. Mener une étude académique sur un sujet aussi complexe n'aurait pas été aussi accessible sans leur soutien.

En dehors de mes superviseurs, je souhaite exprimer ma gratitude envers le reste de l'équipe, le Docteur Nesma SETTOUTI et le Docteur Mohammed Lamine BENOMAR, pour leurs encouragements et leurs suggestions judicieuses. Tous ont joué un rôle majeur dans l'amélioration de mes compétences en rédaction de recherche.

Je tiens également à rendre hommage au Professeur Morad HADJAJ, ancien chef du service de pneumologie, dont l'aide généreuse et humble a été indispen-

sable pour la réalisation de ce travail. Je suis également reconnaissant envers le directeur général de l'hôpital de Tlemcen, monsieur Nasreddine MAZOUNI, pour avoir facilité mon accès au service de Covid19, ce qui a permis de collecter les données essentielles des cas suspects de Covid19. Cette recherche n'aurait pas été possible sans son précieux soutien.

Je garderai toujours en mémoire mes collègues de laboratoire, Hasnaoui Lyna henaar, Fatiha YOUNI et Rima GUILLAL, Amel BENABDALLAH. Depuis que je me suis joint à cette équipe, j'ai toujours pu compter sur votre soutien. Votre patience, vos conseils et votre motivation m'ont été d'un grand secours à chaque étape. Merci du fond du cœur.

J'adresse également mes remerciements à mes amis de l'Université de Tlemcen, en particulier le Dr. Ismail HADJ AHMED, dont le soutien indéfectible m'a aidé à surmonter de nombreux défis dans le processus d'acceptation de mon article.

Enfin, Pour conclure, je tiens à exprimer ma plus profonde gratitude envers mes parents, qui m'ont toujours accordé leur confiance inébranlable, un soutien sans faille et des encouragements constants. Je suis également reconnaissant envers mes frères et sœurs, mes amis et mes connaissances qui ont prié pour mon succès ultime, et je tiens à adresser une mention spéciale à la prochaine docteur, ma chère épouse, pour son aide et son appui indéfectible, Je me considère comme rien sans eux. Leur soutien moral, leurs encouragements et leur motivation ont été des éléments essentiels pour atteindre mes objectifs personnels.

Je dédie cette thèse à ma chère petite fille, Shahd Ines, car sa présence a été une source d'inspiration et de motivation inestimable pour moi, me poussant à mener ce projet à son terme.

# Abstract

**Problem Statement :** This thesis investigates the effectiveness of Convolutional Neural Networks (CNN) in classifying pneumonia cases, a critical task for accurate medical diagnosis. It specifically addresses the challenges posed by the COVID-19 pandemic, where rapid and reliable diagnosis is crucial for patient management and treatment.

**Context :** In a medical environment where traditional diagnostic methods face limitations in accuracy and speed, the urgency of the COVID-19 crisis has highlighted the need for advanced diagnostic tools. This research aims to leverage the capabilities of artificial intelligence to enhance the accuracy of diagnosing pulmonary diseases.

**Results :** The developed CNN model for pneumonia demonstrated remarkable accuracy of 93% and an F1 score of 92.96%, surpassing the outcomes of previous studies that used transfer learning methods or more complex architectures. For COVID-19 detection, a model based on DenseNet201, combined with the Grad-CAM algorithm, produced exceptional results, achieving an accuracy of 98.8%, precision of 99.5%, recall of 98.54%, and specificity of 99.22%. These findings underscore the significant potential of CNNs in improving medical diagnosis, especially during health crises like the COVID-19 pandemic.

**Keywords :** Pneumonia, CNN, COVID-19, artificial intelligence, transfer learning, DenseNet201, Grad-CAM

## ملخص

الكشف عن التهاب الرئة من خلال صور الأشعة السينية للرئتين هو مجال بحث متزايد في المجال الطبي. تُستخدم صور الأشعة السينية للرئتين بشكل شائع لتشخيص التهاب الرئة، ولكن تفسير هذه الصور يمكن أن يكون معقداً وموضوعياً. تم تصميم الشبكات العصبية المتسلسلة لمعالجة البيانات المهيكلة مثل الصور ويمكن استخدامها للكشف عن التهاب الرئة. لتدريب شبكة عصبية متسلسلة على اكتشاف التهاب الرئة، يتطلب ذلك مجموعة من البيانات المتعلقة بـ صور الأشعة السينية للرئتين المحددة. بمجرد تدريب الشبكة العصبية المتسلسلة، يمكن استخدامها لاكتشاف التهاب الرئة من صور الأشعة السينية الجديدة للرئتين. يمكن أن يحسن استخدام الشبكات العصبية المتسلسلة للكشف عن التهاب الرئة الدقة والكفاءة التشخيصية، مما يمكن أن يؤثر بشكل كبير على رعاية المرضى. تهدف هذه الأطروحة إلى تقييم الجهود البحثية الأخيرة في مجال التعلم الآلي، وخاصة في اكتشاف الأمراض المرتبطة بفيروس كورونا، من أجل مكافحة الفيروس. كما تدرس كيف يمكن استخدام التعلم العميق في مجال التشخيص باستخدام صور المسح الطبي للصدر، كما هو مقترح في الأدب المتاح، بالإضافة إلى التحديات التي يواجهها الباحثون. تهدف هذه البحث إلى توفير حلول قادرة على اكتشاف وتحديد الفيروس بشكل أسرع، من خلال تقليل التحيز وتوفير الوقت للمهنيين في مجال الرعاية الصحية.

الكلمات المفتاحية: أشعة الصدر، الالتهاب الرئوي، الشبكة العصبية التلافيفية، فيروس كورونا، التعلم العميق

## Résumé

**Problématique :** La thèse examine l'efficacité des réseaux de neurones convolutifs (CNN) dans la classification des cas de pneumonie, une tâche essentielle pour le diagnostic médical précis. Elle aborde spécifiquement les défis posés par la pandémie de COVID-19, où un diagnostic rapide et fiable est crucial pour la gestion et le traitement des patients.

**Contexte :** Dans un environnement médical où les méthodes de diagnostic traditionnelles rencontrent des limites en termes de précision et de rapidité, l'urgence de la crise COVID-19 a mis en évidence la nécessité d'outils de diagnostic avancés. La recherche vise à exploiter les capacités de l'intelligence artificielle pour améliorer la précision du diagnostic des maladies pulmonaires.

**Résultats :** Le modèle CNN développé pour la pneumonie a démontré une précision remarquable de 93% et un score F1 de 92.96%, surpassant les résultats des études antérieures qui utilisaient des méthodes d'apprentissage par transfert ou des architectures plus complexes. Pour la détection du COVID-19, un modèle basé sur DenseNet201, combiné avec l'algorithme Grad-CAM, a produit des résultats exceptionnels, atteignant une accuracy de 98.8%, une précision de 99.5%, un rappel de 98.54% et une spécificité de 99.22%. Ces découvertes soulignent le potentiel significatif des CNN dans l'amélioration du diagnostic médical, surtout en période de crise sanitaire comme la pandémie de COVID-19.

**Mots clés :** Pneumonie, CNN, COVID-19, intelligence artificielle, apprentissage par transfert, DenseNet201, Grad-CAM

# Table des matières

Remerciements . . . . .	0
Table des matières . . . . .	3
Table des figures . . . . .	5
Liste des tableaux . . . . .	7
Glossaire . . . . .	8
<b>Introduction Générale</b>	<b>1</b>
<b>1 L'apprentissage en profondeur</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.2 Apprentissage Automatique . . . . .	7
1.2.1 Les types d'apprentissage automatique . . . . .	7
1.3 Les réseaux de neurones artificiels (ANN) . . . . .	8
1.3.1 Réseaux de neurones biologiques (BNN) . . . . .	9
1.3.2 Principes des réseaux de neurones artificiels . . . . .	9
1.3.3 Fonctions d'activation . . . . .	11
1.3.4 Rétropropagation et réseaux multicouches . . . . .	13
1.4 Apprentissage en profondeur pour la classification des images . .	18
1.4.1 Réseaux de neurones convolutifs (CNN) . . . . .	18
1.4.2 Architecture des CNNs . . . . .	19
1.4.3 Algorithme d'optimisation . . . . .	23
1.4.4 Régularisation . . . . .	26
1.4.5 Différents types d'architectures CNN . . . . .	27
1.5 Explication des décisions des modèles d'apprentissage en pro- fondeur . . . . .	34
1.5.1 Les méthodes de visualisation . . . . .	34
1.6 Conclusion . . . . .	37
<b>2 Classification des images radiologiques par l'apprentissage en profon- deur</b>	<b>39</b>
2.1 Introduction . . . . .	39
2.2 Images radiologiques . . . . .	40

2.3	Images radiologiques du poumon . . . . .	41
2.4	Détection de la pneumonie dans les images radiographiques pulmonaires à l'aide d'un ensemble de modèles d'apprentissage en profondeur . . . . .	41
2.4.1	Etat de l'art sur la classification des images radiologiques du poumon . . . . .	42
2.4.2	Approche proposée . . . . .	45
2.4.3	Environnement de développement . . . . .	45
2.4.4	Description de la base de données . . . . .	50
2.4.5	Méthodes . . . . .	51
2.4.6	Résultat et discussion . . . . .	55
2.5	Conclusion . . . . .	59
<b>3</b>	<b>Modèle explicable pour l'aide au diagnostic de la COVID-19</b>	<b>60</b>
3.1	Introduction . . . . .	60
3.2	Etat de l'art sur la détection de la COVID-19 . . . . .	61
3.3	Base de données collectée . . . . .	64
3.4	Modèle proposé . . . . .	64
3.5	Résultats et discussion . . . . .	69
3.6	Conclusion . . . . .	75
	<b>Conclusion générale</b>	<b>77</b>
	<b>Bibliographie</b>	<b>79</b>
	<b>Bibliographie</b>	<b>88</b>

# Table des figures

0.1	figure1 . . . . .	3
1.1	Représentation graphique de L'intelligence artificielle et ses sous-domaines [1]. . . . .	6
1.2	Neurone biologique. [2] . . . . .	9
1.3	Principes de base d'un neurone artificiel et les fonctions d'activation. . . . .	10
1.4	Structure générale du réseau de neurones artificiels (ANN) [3] . .	11
1.5	Schéma de la rétropropagation des erreurs [4] . . . . .	17
1.6	La partie gauche est un exemple de convolution 3x3 avec un seul filtre. La dimension supplémentaire sur la convolution est la profondeur de l'entrée. Le côté droit est une convolution 3x3 avec 8 couches de filtres. Notez que chaque filtre a des poids de noyau individuels entraîna- bles [5] . . . . .	18
1.7	Comparaison entre l'apprentissage automatique et l'apprentissage en profondeur . . . . .	19
1.8	Architecture d'un réseau neuronal convolutif (CNN) typique [6].	20
1.9	Processus de convolution. Le filtre ici est de taille 3x3, Le filtre glisse sur l'entrée et exécute sa sortie sur la nouvelle couche. [7]. .	21
1.10	Principe de fonctionnement de la fonction d'activation ReLU. . .	22
1.11	Exemple de pooling avec les différentes stratégies (maximum, moyen). La taille de filtre ainsi que le pas utilisé dans cet exemple est de 2x2 [8]. . . . .	22
1.12	Le principe de fonctionnement de l'algorithme de descente de gradient pour trouver la valeur minimale de la fonction f par la dérivée [9]. . . . .	25
1.13	Architecture du réseau LeNet5 [10]. . . . .	28
1.14	Architecture du réseau de neurone convolutif AlexNet [11]. . . .	29
1.15	Module inception [12]. Il se compose de quatre chemins convolutifs avec des noyaux de tailles différentes. . . . .	30
1.16	Architecture complète du réseau GoogLeNet [12]. . . . .	31

1.17	Architecture VGG16 [13]. . . . .	32
1.18	Le principe de connexion résiduelle [14]. . . . .	33
1.19	Architecture Xception standard [15]. . . . .	34
1.20	Le processus de La carte d'activation de classe (Cam) [16]. . . . .	35
1.21	Un aperçu des deux méthodes, Grad-CAM et GradCAM++. La nuance de gris dans les grilles montre les attributions calculées des pixels correspondants. En haut à droite : Grad-CAM ne parvient pas à mettre correctement en évidence tous les pixels importants. En bas à droite : Grad-CAM++ attribue correctement la même importance aux pixels de toutes les cartes d'activation. [17].	37
2.1	Exemples de deux plaques radiographiques qui affichent (a) un poumon sain et (b) un poumon pneumonique. [18]. . . . .	43
2.2	Le datagramme de l'ensemble de données utilisé pour la phase d'apprentissage qui montre le déséquilibre des données. [19]. . .	51
2.3	La radiographie pulmonaire normale (panneau de gauche) montre des poumons clairs sans aucune zone d'opacification anormale sur l'image. La pneumonie bactérienne (au milieu) présente généralement une consolidation lobaire focale, dans ce cas dans le lobe supérieur droit (flèches blanches), tandis que la pneumonie virale (à droite) se manifeste par un schéma interstitiel plus diffus dans les deux poumons. [18]. . . . .	52
2.4	L'architecture proposée. . . . .	54
2.5	Performance du modèle de classification sur des dimensions d'images différentes. . . . .	56
3.1	Échantillons de l'ensemble des données collectées. . . . .	65
3.2	Schéma fonctionnel de ce travail. . . . .	66
3.3	Architecture du DenseNet201 [20]. . . . .	68
3.4	Organigramme du modèle proposé [20]. . . . .	69
3.5	Analyse d'Accuracy de l'entraînement et du test et analyse de la perte de l'entraînement et du test sur 200 époques. . . . .	72
3.6	Explication avec la carte thermique Grad-cam sur des cas positifs de COVID-19 . . . . .	74
3.7	Aire sous la courbe (AUC) du modèle proposé. . . . .	75

# Liste des tableaux

2.1	Paramètres pour l'augmentation de l'image. . . . .	52
2.2	Le résultat de l'architecture de réseau proposée. . . . .	55
2.3	Performance du modèle de classification sur différentes tailles de données . . . . .	56
2.4	Comparaison des résultats . . . . .	58
3.1	Les paramètres utilisés pour la méthode d'augmentation de donnée. . . . .	66
3.2	Comparaisons des résultats entre les différentes architectures appliquées à notre jeu de données. . . . .	71
3.3	TP, TN, FP et FN obtenu à partir de l'ensemble de données de test. . . . .	72
3.4	Comparaisons des résultats entre les différentes architectures appliquées à l'ensemble de données de [21]. . . . .	73

# Glossaire

COVID-19 : Coronavirus Disease 2019.  
OMS : Organisation mondiale de la Santé.  
SARS-CoV-2 : Coronavirus 2 du syndrome respiratoire aigu sévère.  
RT-PCR : Reverse Transcriptase Polymerase Chain Reaction.  
AI : Artificial intelligence.  
DL : Deep Learning.  
ML : Machine Learning.  
CNN : Convolutional Neural Network .  
ANN : artificial neural network.  
BNN : Biological Neural Network.  
ReLU : Rectified Linear Unit.  
ILSVRC : ImageNet Large Scale Visual Recognition Challenge.  
BN : Batch Normalization.  
NAG : Nesterov Accelerated Gradient.  
RMSProp : Root mean square propagation.  
Adam : Adaptive Moment Estimation.  
GAP : Global Average Pooling.  
CAM : Class activation map.  
Grad-CAM : Gradient-weighted Class Activation Mapping.  
TDM : Tomodensitométrie.  
IRM : Imagerie par résonance magnétique.  
SciPy : Scientific Python.  
GPU : Graphics Processing Unit.  
CPU : Central Processing Unit.  
TPU : Tensor Processing Unit.  
IRM : Imagerie par résonance magnétique  
MLP : Perceptron multicouche (multilayer perceptron en anglais)  
SMO : Optimisation Minimal Séquentielle  
AUC : Aire sous la courbe (Area Under the Curve)  
OCT : Tomographie à cohérence optique(Optical Coherence Tomography)  
Numpy : Numerical Python

SciPy : Scientific Python

CT : Computed tomography

DNN : Deep Neural Networks

SVM : Support Vector Machine ou Machine à vecteurs de support

ROC : Receiver operating characteristic

SGD : Descente stochastique de gradient

DTL : Deep Transfer Learning

TP : Vrai positif ( True positive)

VN : Vrai négatif (True negative)

FP : Faux positif (False positive)

FN : Faux négatif (False negative)

# Introduction Générale

## Contexte et motivations

L'imagerie médicale est utilisée depuis longtemps pour diagnostiquer les maladies, et l'une des premières formes de cette technologie est l'imagerie par rayons X. En 1900, environ 5 ans après son invention, l'appareil à rayons X était déjà considéré comme primordial dans le diagnostic des fractures et des corps étrangers [22]. Actuellement, les examens radiologiques sont une méthode fréquemment utilisée pour identifier les anomalies dans la poitrine, y compris celles causées par une maladie respiratoire. Une radiographie pulmonaire montrera les poumons du patient, à partir desquels des conditions telles que l'œdème et l'inflammation peuvent être observées par des radiologues qualifiés. La prolifération de l'imagerie médicale a créé une demande de techniques avancées pour analyser les données produites. Cela est dû en partie à la grande quantité d'images créées, ainsi qu'au coût et aux erreurs associés à l'analyse par des professionnels. Parallèlement à cette augmentation de la demande, il y a eu un raffinement et une propagation accrue des techniques d'apprentissage automatique pour les données d'image. Les algorithmes d'apprentissage en profondeur sont devenus essentiels dans de nombreuses tâches de vision par ordinateur. En particulier, le réseau neuronal convolutif (CNN) est devenu un outil couramment utilisé pour la classification des images, qui peut souvent produire des niveaux de précision proches de l'homme.

Dans le contexte de la détection de la pneumonie, les CNNs peuvent être utilisés pour analyser les images radiographiques pulmonaires et identifier les signes caractéristiques de la maladie. En entraînant un CNN sur un grand ensemble de données d'images radiographiques pulmonaires annotées, le modèle peut apprendre à reconnaître les motifs associés à la présence de pneumonie.

Une fois entraîné, le CNN peut être utilisé pour classer de nouvelles images radiographiques pulmonaires et déterminer si elles présentent des signes de pneumonie. Cette approche permet une détection rapide et précise de la maladie, ce qui peut être d'une grande aide pour les médecins dans leur processus de diagnostic. En outre, une application pertinente de la classification des images

est en réponse à la pandémie du Corona virus (COVID-19).

Le 30 janvier 2020, le comité d'urgence du Règlement sanitaire international de l'organisation mondiale de la santé (OMS) a déclaré l'épidémie de la COVID-19, comme une « urgence de santé publique de portée internationale ». La maladie du COVID-19, causée par le coronavirus 2 du syndrome respiratoire aigu sévère (SRAS-CoV-2), s'est propagée à tous les continents habités, devenant une crise de santé publique sans précédent. En octobre 2020, l'Université de médecine Johns Hopkins [23] a signalé un total de plus d'un million de décès alors que les infections au COVID-19 dans le monde dépassaient les 40 millions. Une propagation aussi rapide est due au fait que ce virus se transmet très facilement de personne à personne par la toux, les éternuements et les gouttelettes respiratoires [24]. Elle se manifeste généralement par des symptômes de fièvre, de toux, d'essoufflement et peut avoir des conséquences graves telles qu'une pneumonie, une défaillance multi-viscérale et la mort.

Aujourd'hui, il n'existe pas de traitement pour la COVID-19. La grande majorité des mesures prises au niveau des pays visent à empêcher la transmission de ce virus à un plus grand nombre de personnes. En raison de l'incertitude dans la dynamique de transmission du COVID-19 et de la certitude élevée de sa virulence, il est compréhensible que les premières réponses aient reposé sur des interventions très strictes, telles que des interdictions de déplacements et des fermetures, pour sauver des vies. Compte tenu de la charge de travail croissante des médecins spécialistes, il est urgent d'augmenter les compétences médicales et économiques pour faire face à cette maladie grave. Par conséquent, le défi scientifique consiste maintenant à identifier, par inférence et simulation, des mesures qui pourraient fournir une protection aussi bonne ou meilleure avec un coût social moindre [25]. L'importance croissante accordée aux techniques d'apprentissage automatique dans les domaines médicaux peut fournir un environnement propice au changement et à l'amélioration. Pour faire face à cette nouvelle pandémie mondiale, l'OMS, les scientifiques et les cliniciens des industries médicales sont à la recherche de nouvelles technologies pour dépister les patients infectés à différents stades, trouver les meilleurs essais cliniques, contrôler la propagation de ce virus, développer un vaccin et développer un traitement pour guérir les patients infectés [24]. Le rôle de la science des données dans ce scénario consiste à aider l'accélération de processus.

Le 16 mars 2020, la Maison Blanche, en collaboration avec des instituts de recherche et des entreprises technologiques, a lancé un appel à l'action pour les chercheurs mondiaux en intelligence artificielle pour le développement de nouvelles techniques d'extraction de texte et de données pour aider la recherche liée au COVID-19 [26]. "Allen Institute for AI", en partenariat avec des groupes de recherche de premier plan, ont publié un ensemble de données COVID-19 de re-

cherche disponible publiquement et mis à jour chaque semaine, qui documente en continu les articles liés à la COVID-19 pour accélérer de nouveaux projets de recherche nécessitant de toute urgence des données en temps réel. Des centaines d'équipes de recherche conjuguent leurs efforts pour collecter des données et développer des solutions chaque jour. L'apprentissage automatique s'est avéré efficace pour prédire les risques dans de nombreux domaines et depuis le début de la propagation du virus, son application nous aide à lutter contre la pandémie virale. Comme jamais auparavant, les professionnels du monde entier collectent et partagent ce qu'ils apprennent sur le virus. À partir de là, l'objectif principal de ce travail est de contribuer à ces travaux en collectant une nouvelle base de données des images de scanner thoracique des cas suspects de Covid-19 et confirmer avec l'atteinte (positive ou négative) par la RT-PCR (Reverse Transcriptase Polymerase Chain Reaction) au niveau du centre hospitalo-Universitaire de Tlemcen, en appliquant et en soulignant l'importance du rôle de l'apprentissage automatique pour lutter contre le SARS-CoV-2 (Figure 0.1)

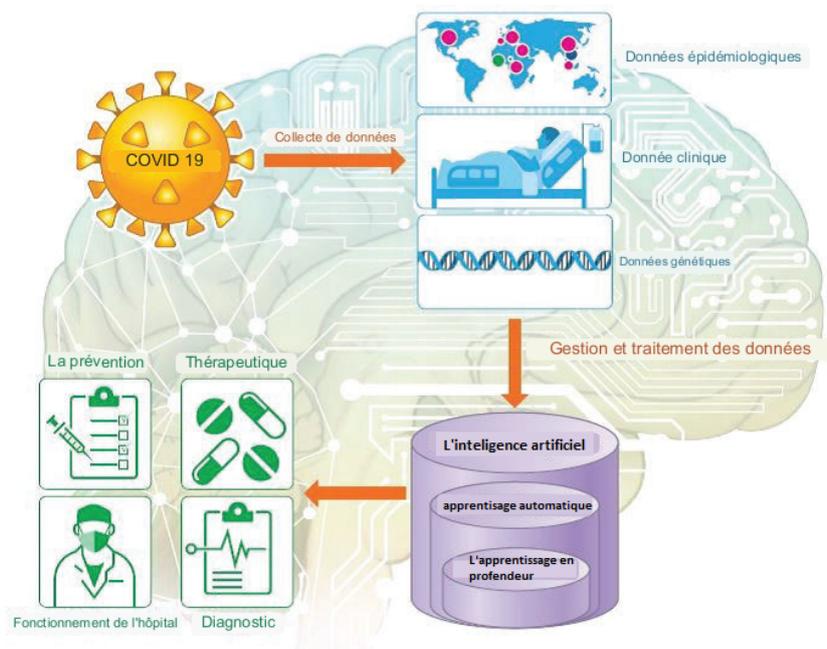


FIGURE 0.1 – Rôle de l'IA dans la lutte contre le COVID-19. [27]

Dans ce contexte, les objectifs de cette thèse sont :

- explorer le panorama général pour présenter les avancées récentes des technologies d'apprentissage automatique et leurs applications dans la lutte contre la COVID-19;

- identifier les réalisations et les défis à relever pour faire progresser les systèmes d'apprentissage automatique en médecine afin de lutter contre le virus ; approfondir la manière dont les applications d'apprentissage automatique dans différents domaines ont affecté la pandémie ;
- Faciliter la tâche de détection de la Covid-19 aux médecins, car le nombre de radiologues reste très faible par rapport à le nombre des patients infecté par la covid19. Par conséquent, l'utilisation des techniques d'apprentissage en profondeur (DL) est utile pour la détection automatique du COVID-19. Ces techniques permettent de traiter d'énormes ensembles de données et d'images avec un haut degré de précision et à moindre coût.

## Contributions

Étant donné que COVID-19 est une maladie aiguë, sur un temps qui peut aller de quelques jours à quelques semaines, les changements physiopathologiques affectant la zone pulmonaire normale peuvent être facilement reconnus par les des différents approches. Généralement, ces techniques existaient dans l'apprentissage en profondeur, il est donc passionnant pour les chercheurs de travailler sur COVID-19 et sa distinction par rapport à différentes images de patients. En résumé, les contributions de ce travail sont principalement axées sur les techniques, l'ensemble de données et l'analyse suivants :

- 1) Dans un premier temps, nous avons collecté de différents cas de patients COVID-19 suspectés et infectés pour former une base de données. Il construit un nouvel ensemble de données COVID-19 à grande échelle qui contient des images en série étiquetées de patients non infectés et infectés et confirmer avec la RT-PCR.
- 2) nous avons proposé un système de diagnostic de la COVID-19 pour effectuer la détection et la classification montrant une nette supériorité.
- 3) Sur l'ensemble de données COVID-19 collecté, Le système proposé démontre d'excellentes performances en termes de sensibilité, de spécificité et d'autres métriques de classification pour la détection du COVID-19 en utilisant un modèle d'apprentissage par transfert.

## Organisation de la thèse

Les travaux de cette thèse sont structurés en trois chapitres :

- 1) **Le chapitre 1** détaille les bases théoriques et les concepts généraux sur l'apprentissage en profondeur, ainsi que une description plus détaillée sur les réseaux de neurones convolutifs (CNNs) qui est la méthode choisie dans notre projet. par la suite, nous étudierons l'architecture des réseaux de neurones convolutifs et toutes leurs propriétés telles que les couches, les fonctions, les optimiseurs et tous les paramètres qui contrôlent l'apprentissage du réseau. Ensuite nous allons expliquer les méthodes d'explication des décisions des modèles d'apprentissage en profondeur.
- 2) **Le chapitre 2** introduit brièvement les images radiologiques en général et les images radiologiques du poumon en particulier. Dans ce chapitre, nous détaillons le système de classification utilisé pour détecter et classier la pneumonie dans les images radiographiques pulmonaires à l'aide d'un ensemble de modèles d'apprentissage en profondeur.
- 3) **Le chapitre 3** Dans cette partie de thèse, nous avons proposé une approche systématique pour la détection de COVID-19, et la classification de la gravité des patients à partir des images de tomodensitométrie et en se basant sur un système d'explication Grad-cam. Pour trouver les modèles d'apprentissage profond les plus performants, nous avons étudié plusieurs réseaux en s'appuyant sur un apprentissage par transfert.
- 4) Enfin, une conclusion générale synthétise le contenu de cette thèse.

# Chapitre 1

## L'apprentissage en profondeur

### 1.1 Introduction

L'apprentissage en profondeur ou apprentissage profond est un sous-domaine de l'apprentissage automatique, qui est à son tour un sous-domaine de l'intelligence artificielle, Figure 1.1.

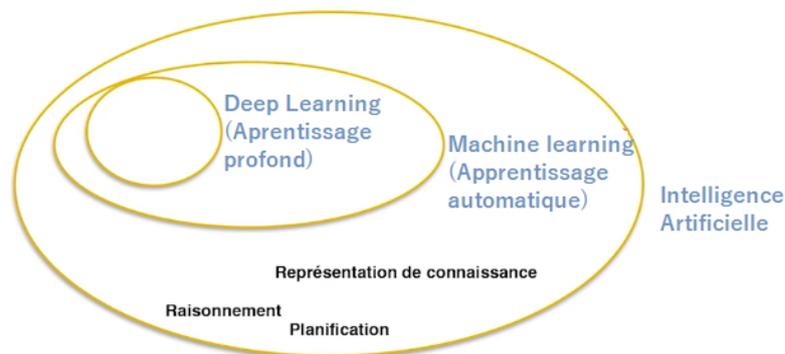


FIGURE 1.1 – Représentation graphique de L'intelligence artificielle et ses sous-domaines [1].

L'apprentissage en profondeur est une sous-catégorie des méthodes d'apprentissage automatique. Ces méthodes utilisent des architectures complexes pour modéliser des données à travers une multitude de transformations non linéaires, avec les réseaux de neurones comme élément fondamental. Parmi les avancées majeures qu'a connu l'apprentissage en profondeur, nous retrouvons des domaines comme le traitement du son et de l'image, la reconnaissance fa-

ciale, la reconnaissance de la parole, la vision par ordinateur, le traitement automatisé du langage naturel et la classification de texte.

Il existe diverses architectures de réseaux de neurones :

- Les perceptrons multicouches, qui peuvent être considérés comme l'architecture de base.
- Les réseaux de neurones convolutifs (CNN) principalement conçus pour le traitement d'images.
- Les réseaux de neurones récurrents pour le traitement de données séquentielles comme les textes ou les séries temporelles.

La profondeur des couches, les algorithmes d'optimisation, et la structure de ces réseaux sont méticuleusement élaborés. Cependant, bien que les méthodes d'apprentissage en profondeur donnent des résultats impressionnants, elles souffrent encore d'un manque de fondements théoriques solides. Dans ce chapitre, nous étudierons les bases de l'apprentissage profond et ses relations avec d'autres méthodes d'apprentissage automatique - supervisé, semi-supervisé, et non supervisé. Nous examinerons également des structures de base comme les perceptrons et les réseaux de neurones artificiels.

## 1.2 Apprentissage Automatique

L'apprentissage automatique est un champ de l'informatique et de la statistique qui vise à créer des algorithmes capables de faire des prédictions ou des décisions sans être explicitement programmés pour le faire. Ces algorithmes sont conçus pour comprendre et extraire de l'information à partir de grands ensembles de données, rivalisant parfois avec la capacité humaine d'extraction d'information. La modélisation statistique est devenue un élément central de l'apprentissage automatique avec la croissance exponentielle des volumes de données et les avancées dans la puissance de calcul. En essence, ces algorithmes "apprennent" à partir de l'information pour s'améliorer avec le temps.

### 1.2.1 Les types d'apprentissage automatique

#### L'apprentissage supervisé

L'apprentissage supervisé est une méthode d'apprentissage automatique où l'algorithme est entraîné sur un ensemble de données comportant des entrées et des sorties (ou étiquettes) correspondantes. L'objectif principal est de permettre à l'algorithme de générer des prédictions ou des décisions basées sur de

nouvelles données d'entrée. Les tâches d'apprentissage supervisé peuvent être classées en deux principales catégories :

- **Classification** : Dans les problèmes de classification, l'objectif est de prédire une étiquette catégorielle pour une donnée d'entrée. Par exemple, déterminer si un e-mail est un spam ou non.
- **Régression** : Dans les problèmes de régression, l'objectif est de prédire une valeur continue. Par exemple, prévoir le prix d'une maison en fonction de ses caractéristiques.

### L'apprentissage non-supervisé

L'apprentissage non-supervisé se réfère aux méthodes d'apprentissage automatique où l'algorithme est formé sur un ensemble de données sans étiquettes de sortie. Au lieu de fournir des réponses prédictives, l'apprentissage non-supervisé vise à identifier des structures ou des motifs cachés dans les données. Les principales techniques associées à cette approche comprennent :

- **Clustering** : Il s'agit de grouper des données similaires ensemble. Par exemple, segmenter les clients en différents groupes basés sur leurs comportements d'achat.
- **Réduction de dimensionnalité** : Cette technique cherche à réduire le nombre de variables dans un jeu de données tout en préservant l'essentiel de l'information. L'Analyse en Composantes Principales (ACP) est un exemple courant de cette technique.

### L'apprentissage semi-supervisé

L'apprentissage semi-supervisé est un compromis entre les méthodes supervisées et non supervisées. Dans cette approche, l'algorithme est formé sur un mélange de données étiquetées et non étiquetées. L'idée est d'utiliser les données non étiquetées pour renforcer l'apprentissage à partir des données étiquetées, souvent en exploitant la structure ou la distribution sous-jacente des données pour améliorer les performances du modèle.

## 1.3 Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels sont souvent utilisés en apprentissage profond. Inspirés du fonctionnement des neurones dans le cerveau humain, ils cherchent à établir des relations entre les données d'entrée et de sortie pour

établir des modèles prédictifs. Nous discuterons du modèle de perceptron, de l'algorithme de rétropropagation et de leur utilisation dans les réseaux multicouches.

### 1.3.1 Réseaux de neurones biologiques (BNN)

Les réseaux de neurones biologiques ont été la principale source d'inspiration pour le développement des réseaux de neurones artificiels, avec pour objectif d'imiter leur fonctionnement afin que les machines puissent réaliser des tâches complexes à la manière des humains. L'unité de base d'un réseau de neurones est le neurone. Il est constitué d'un corps cellulaire, de dendrites et d'un axone (voir figure 1.2). Les connexions entre neurones se font via des fibres, et les dendrites se ramifient dans plusieurs directions de manière complexe. Ces dendrites reçoivent des signaux provenant de neurones voisins, tandis que les axones servent à transférer les signaux vers d'autres neurones. La transmission de signaux d'un neurone à un autre s'effectue au niveau de la synapse située à l'extrémité des dendrites. À la manière d'un effet domino, les impulsions sont relayées de neurone en neurone.

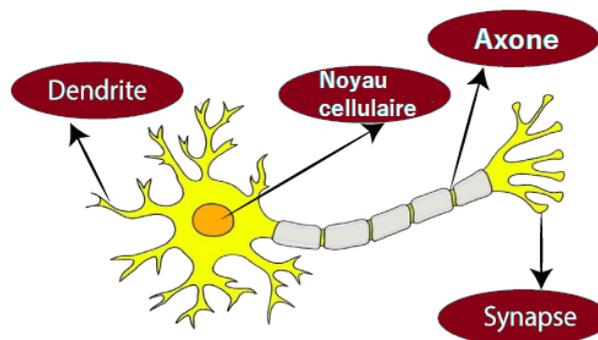


FIGURE 1.2 – Neurone biologique. [2]

### 1.3.2 Principes des réseaux de neurones artificiels

L'unité fondamentale du neurone artificiel est conçue en s'inspirant du modèle biologique, bien qu'elle soit nettement plus simple. La figure 1.3 illustre la structure basique d'un neurone artificiel.

Les ANN (Artificial Neural Networks) sont des architectures informatiques inspirées du cerveau humain. Ils sont formés d'une succession de « neurones »

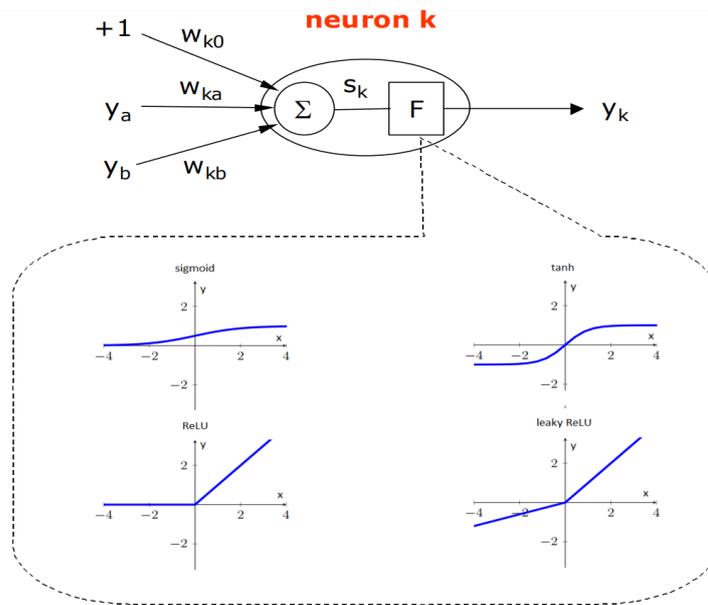


FIGURE 1.3 – Modèle d’un neurone artificiel et ses fonctions d’activation.

(ou « nœuds ») organisés en couches [28]. Le comportement global de ces neurones est déterminé par les connexions établies entre les divers éléments de traitement et les paramètres associés au sein de l’architecture du réseau neuronal. Chaque connexion reliant les neurones de deux couches consécutives possède une pondération. Le poids  $w_{ij}$  symbolise l’intensité de la connexion entre le  $i$ -ème neurone d’une couche et le  $j$ -ème neurone de la couche suivante. Un réseau neuronal typique comprend une couche d’entrée, une ou plusieurs couches cachées, et une couche de sortie. Le nombre de couches cachées et le nombre de neurones au sein de ces couches dépendent de la complexité du problème traité. La figure 1.4 présente une architecture ANN classique comprenant deux couches cachées. Dans un tel réseau, les données sont reçues par les neurones de la couche d’entrée, puis transmises aux neurones de la première couche cachée via des connexions pondérées. Ensuite, les données de chaque couche sont traitées mathématiquement et le résultat est envoyé à la couche suivante. Les étapes ci-après détaillent comment les données entrantes ( $x_i$ ) sont traitées par le  $j$ -ème nœud de la couche suivante, un processus représenté par la figure 1.4.

- 1) Tout d’abord, une somme pondérée est calculée, puis le terme de biais ( $\theta_j$ ) est ajouté à cette somme selon l’équation ci-dessous :

$$net_j = \sum_{i=1}^m x_i * w_{ij} + \theta_j (j = 1, 2, \dots, n) \quad (1.1)$$

- 2)  $net_j$  est transformé à l'aide d'une "fonction d'activation" mathématique. Cette fonction est utilisée pour normaliser toutes les entrées et sorties du réseau sur une plage finie de valeurs. Cela permet au réseau de neurones de mieux identifier les modèles et les tendances dans les données que d'avoir une plage de valeurs entre 50 et 500, par exemple. De nombreuses fonctions d'activation peuvent être utilisées tel que, la fonction sigmoïde.

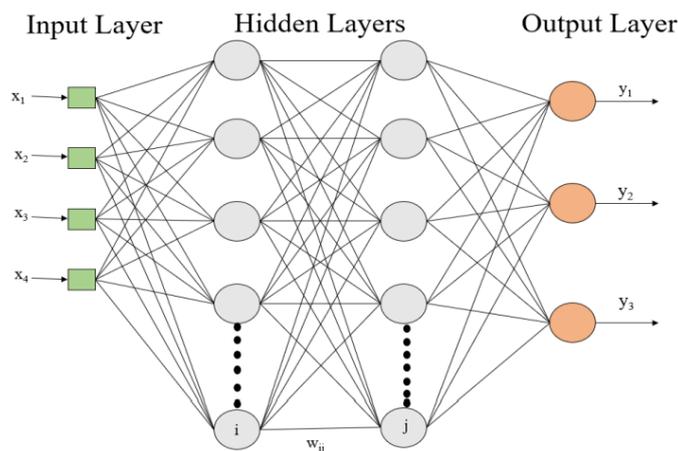


FIGURE 1.4 – Structure générale du réseau de neurones artificiels [3]

L'algorithme d'apprentissage peut être résumé en quatre points :

- 1) initialiser les poids par des nombres aléatoires (ou à 0).
- 2) Calculer la sortie  $y_j$  de chaque échantillon d'entrée  $x_i$ .
- 3) Pour chaque échantillon, mettre à jour simultanément les poids.
- 4) Répétez les étapes 2-3 jusqu'à ce qu'il n'y ait plus d'erreurs.

### 1.3.3 Fonctions d'activation

Le concept de fonctions d'activation s'inspire du fait que les signaux électriques des neurones biologiques atteignent des seuils d'activation et déclenchent des informations neuronales pour se propager le long des axones. Les fonctions d'activation utilisées dans les réseaux de neurones artificiels et les réseaux de neurones profonds sont des fonctions non linéaires qui permettent la classification de données non linéairement séparables. Les fonctions d'activations les plus utilisées sont la fonction sigmoïde, la fonction tanh, la fonction ReLU et la

fonction Leaky Relu. Différentes fonctions d'activations peuvent être utilisées selon les applications; dans ce travail, la fonction ReLU a été utilisée. La figure 1.3, illustre quelques type de fonctions d'activation.

- 1) *Fonction sigmoid* : En utilisant la fonction d'activation sigmoïde, le neurone artificiel ressemblera davantage au neurone naturel. L'avantage de cette fonction est qu'elle effectue une transition très douce (voir figure 1.3). Cela donne des sorties dans la plage  $[0, 1]$  :

$$y = \frac{1}{1 + e^{-x}} \quad (1.2)$$

- 2) *Fonction tanh* : Une autre fonction d'activation utilisée dans réseaux de neurones artificiels est la fonction hyperbolique tangente représentée par  $\tanh$ . elle est calculée comme suit (équation 1.3) :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{2 - (1 + e^{-2x})}{1 + e^{-2x}} = \frac{2}{1 + e^{-2x}} - 1 = 2\text{sigmoid}(2x) - 1 \quad (1.3)$$

Nous observons que la fonction  $\tanh$  est une autre version modifiée de la fonction sigmoïde (voir figure 1.3).

- 3) *Fonction ReLU* : La fonction d'activation rectifiée linéaire (ReLU) est une fonction linéaire par morceaux qui renvoie directement l'entrée si elle est positive, sinon elle renvoie zéro. Cette fonction est devenue la fonction d'activation par défaut pour de nombreux types de réseaux de neurones en raison de ses avantages en termes d'apprentissage et de performances.

La fonction ReLU présente plusieurs avantages. Tout d'abord, elle est simple et efficace à calculer, ce qui en fait un choix courant dans les réseaux de neurones. De plus, la fonction ReLU ne souffre pas du problème de disparition du gradient, qui peut se produire avec d'autres fonctions d'activation telles que la fonction sigmoïde. Cela signifie que les réseaux de neurones utilisant la fonction ReLU peuvent être plus faciles à entraîner et convergent très rapidement.

En outre, la fonction ReLU permet aux réseaux de neurones d'apprendre des représentations plus dispersées et plus discriminantes des données. En renvoyant zéro pour les valeurs négatives, la fonction ReLU supprime les activations négatives et favorise la sélectivité des neurones. Cela peut conduire à une meilleure capacité de généralisation et à une réduction du

surapprentissage.

La fonction ReLU est définie mathématiquement comme suit : (équation 1.4) :

$$ReLU(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (1.4)$$

En conclusion, la fonction d'activation rectifiée linéaire (ReLU) est une fonction couramment utilisée dans les réseaux de neurones en raison de ses avantages cités précédemment.

- 4) *Fonction leaky ReLU* : La fonction leaky ReLU est une amélioration de la fonction d'activation ReLU. Contrairement à la fonction ReLU, qui désactive les neurones en attribuant un gradient de 0 à toutes les valeurs d'entrée inférieures à zéro, le leaky ReLU résout ce problème. Au lieu de définir la fonction d'activation ReLU comme étant égale à 0 pour les valeurs négatives des entrées ( $x$ ), le leaky ReLU les définit comme une composante linéaire extrêmement petite de  $x$ . ci-dessous la formule de cette fonction d'activation équation (1.5) :

$$ReLU_{leaky}(x) = \max(0.01 * x, x). \quad (1.5)$$

### 1.3.4 Rétropropagation et réseaux multicouches

Les principales caractéristiques de performance des réseaux de neurones comprennent la capacité d'apprentissage et la mémoire distribuée, ce qui conduit finalement à une tolérance aux erreurs. En raison de ces avantages, la rétropropagation est un représentant emblématique des algorithmes de descente de gradient, technique couramment utilisée pour l'apprentissage des réseaux de neurones dans divers domaines d'application. La rétropropagation dans les réseaux de neurones multicouches utilise une technique d'apprentissage supervisé qui cartographie la relation complexe et non linéaire entre les entrées (variables indépendantes) et les sorties (variables dépendantes) au moyen du processus d'apprentissage avec des ensembles de données d'entraînement [29]. Le vecteur d'entrée est présenté au réseau, et la phase de propagation avant est tout d'abord exécutée. Ainsi, une erreur entre la sortie réelle du réseau et la sortie désirée, qui est fournie par l'ensemble de données d'entraînement, est calculée. Ces erreurs sont ensuite propagées en sens inverse (de la sortie vers l'entrée) pour déterminer les ajustements nécessaires aux poids de connexion dans le réseau de neurones. L'apprentissage dans le réseau de rétropropagation se compose de trois phases principales : l'introduction des données d'entrée, la

rétropropagation des erreurs et la mise à jour des poids de connexion. Ces trois phases principales sont détaillées dans la section suivante.

### Processus de propagation en avant :

Dans un réseau de neurones multicouches utilisant la rétropropagation, chaque neurone de la première couche reçoit une valeur d'entrée issue de l'ensemble de données d'apprentissage. Chacune de ces valeurs d'entrée est ensuite multipliée par les poids de connexion correspondants. Les résultats sont sommés, et une fonction d'activation est appliquée à cette somme, transformant ainsi la sortie en une valeur appartenant à une plage fixe, telle que  $[0, 1]$  ou  $[-1, 1]$ . La valeur résultante est ensuite transmise à tous les neurones de la couche suivante, qu'il s'agisse d'une couche cachée ou de la couche de sortie [30]. Les étapes de ce processus de propagation direct sont décrites ci-après [30] :

- 1) Étape 1 : Chaque neurone d'entrée  $x_i$  ( $i = 1, 2, \dots, n$ ), reçoit sa valeur d'entrée du modèle d'apprentissage.
- 2) Étape 2 : Chaque neurone caché  $h_j$  ( $j = 1, 2, \dots, p$ ), reçoit sa valeur d'entrée des neurones d'entrée :

- Additionnez l'entrée pondérée  $net_{h_j}$  comme suit :

$$net_{h_j} = (v_{0j} + \sum_{i=1}^n x_i v_{ij}) \quad (1.6)$$

Où  $v_{0j}$  est le biais de la couche cachée  $j$  ( $j = 1, \dots, p$ ),  $X$  est le vecteur d'apprentissage d'entrée :  $X = x_i, i = 1, \dots, n$ , et  $V_i$  est le vecteur de poids entrée-caché :  $V_i = v_{ij}, j = 1, \dots, p$ .

- calculez les valeurs de sortie en utilisant la fonction d'activation :

$$h_j = f(net_{h_j}) \quad (1.7)$$

Où  $f$  est la fonction d'activation sigmoïde binaire.

- 3) Étape 3 : Chaque neurone de sortie  $y_k$  a reçu son entrée du neurone caché :

- Additionnez l'entrée pondérée comme suit :

$$net_{y_k} = (w_{ok} + \sum_{j=1}^p h_j w_{jk}) \quad (1.8)$$

Où  $w_{0k}$  est le biais de la couche de sortie  $k(k = 1, \dots, m)$ ,  $h_j$  est la sortie du neurone caché, et  $W_j$  est le vecteur de poids caché-à-sortie :  $W_j = w_{jk}, k = 1, \dots, m$ .

- la fonction d'activation a été utilisé pour calculer les valeurs de sortie  $y_k$  comme suit :

$$y_k = \tilde{f}(net_{yk}) \quad (1.9)$$

Où  $\tilde{f}(x)$  est la dérivée de la fonction d'activation sigmoïde :  $\tilde{f}(x) = f(x)[1 - f(x)]$ .

### Processus de rétropropagation d'erreur :

L'erreur de prédiction est la différence entre la sortie réelle du réseau et la sortie désirée correspondante. Au cours du processus de rétropropagation des erreurs, le taux d'apprentissage ( $\delta$ ) détermine la quantité de modification de poids de connexion. Les étapes de la rétropropagation des erreurs peuvent être exprimées comme suit [30] :

1) Étape 1 : Pour chaque neurone de sortie  $y_k(k = 1, \dots, m)$  :

- Calculez l'erreur  $\delta_k$  comme suit :

$$\delta_k = (target_k - y_k)\tilde{f}(net_{yk}) \quad (1.10)$$

Où Target est le vecteur cible de sortie :  $Target = target_k, k = 1, \dots, m$  et  $y_k$  est la sortie réelle et  $\tilde{f}(x)$  est la dérivée de la fonction d'activation sigmoïde.

- Calculez le terme de correction de poids  $\Delta w_{jk}(j = 1, \dots, p)$  et  $\Delta w_{0k}$  comme suit :

$$\Delta w_{jk} = \partial \delta_k h_j, \Delta w_{0k} = \partial \delta_k \quad (1.11)$$

Où  $\delta_k$  est le terme d'erreur au niveau de la couche de sortie  $y_k$  et utilisé pour modifier le poids  $w_{jk}$ , et  $\partial$  est le taux d'apprentissage qui définit la quantité de poids modifiés pendant le cycle d'apprentissage.

2) Étape 2 : Pour chaque neurone caché  $h_j(j = 1, \dots, p)$  :

- Calculez le terme d'erreur  $\delta_j$  comme suit :

$$\delta_j = \left( \sum_{k=1}^m \delta_k w_{jk} \right) \tilde{f}'(net_{hj}) \quad (1.12)$$

Où  $\delta_j$  est le terme d'erreur à la couche cachée  $h_j$  et utilisé pour modifier le poids  $v_{ij}$ , et  $\tilde{f}'(x)$  est la dérivée de la fonction d'activation sigmoïde :  $\tilde{f}'(x) = f'(x)[1 - f(x)]$ .

- Calculez le terme de correction de poids  $v_{ij}$  et  $v_{0j}$  comme suit :

$$\Delta v_{ij} = \partial \delta_{jxi}, \Delta v_{0j} = \partial \delta \quad (1.13)$$

Où  $V_i$  est le vecteur des poids d'entrée à la couche cachée :  $V_i = v_{ij}, j = 1, \dots, p$ ,  $\delta_j$  est le terme d'erreur à la couche cachée  $h_j$ , et  $v_{0j}$  est le biais sur la couche cachée  $j(j = 1, \dots, p)$ .

### Processus de mise à jour des poids de connexion :

Lors de l'apprentissage les poids du réseau sont mis à jour par la règle du delta généralisé. Tous les poids des connexions sont modifiés dans chaque couche. Les étapes de la mise à jour des poids peuvent être décrites comme suit [30] :

1) Étape 1 : Chaque neurone de sortie  $y_k(k = 1, \dots, m)$  :

- Modifie son biais et ses pondérations comme suit :

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (1.14)$$

Où  $W_j(j = 1, 2, \dots, p)$  est le vecteur de poids caché-à-sortie :  $W_j = \{w_{jk}, k = 1, \dots, m\}$ .

1) Étape 2 : Chaque neurone caché  $h_j(j = 1, \dots, p)$  :

Modifie son biais et ses pondérations comme suit :

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (1.15)$$

Où  $V_i(i = 1, 2, \dots, n)$  est le vecteur de poids d'entrée-à-caché :  $V_i = \{v_{ij}, j = 1, \dots, p\}$ .

L'apprentissage d'un réseau de neurones artificiel est un processus d'ajustement des poids de connexion. La figure 1.5 montre le diagramme schématique de la rétropropagation des erreurs. L'erreur quadratique pour chaque neurone est la différence au carré entre la sortie désirée et la sortie réelle.

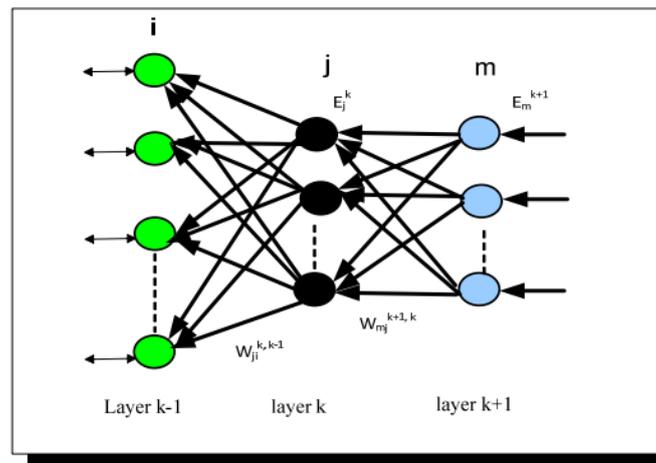


FIGURE 1.5 – Schéma de la rétropropagation des erreurs [4]

### L'algorithme d'apprentissage par rétropropagation :

Les étapes d'apprentissage de l'algorithme de rétropropagation utilisés sont resumés comme suit :

- 1) Les poids sont initialisés : Les poids du réseau sont initialement définis de manière aléatoire.
- 2) Propagation avant : Les données d'entrée sont transmises à travers le réseau pour calculer les sorties prévues.
- 3) Calcul de l'erreur : La différence entre les sorties prévues et les sorties réelles est calculée à l'aide d'une fonction de perte.
- 4) Propagation arrière : L'erreur est propagée de la couche de sortie vers les couches précédentes pour calculer les gradients par rapport aux poids.
- 5) Mise à jour des poids : Les poids sont ajustés en utilisant les gradients calculés, généralement avec un taux d'apprentissage pour contrôler la taille des mises à jour.
- 6) Répétition : Les étapes 2 à 5 sont répétées pour de nombreuses itérations (époques) jusqu'à ce que l'erreur soit minimisée ou converge.
- 7) Arrêt : L'algorithme s'arrête lorsque l'erreur atteint un seuil acceptable ou lorsque le nombre d'époques prédéfini est atteint.

## 1.4 Apprentissage en profondeur pour la classification des images

### 1.4.1 Réseaux de neurones convolutifs (CNN)

Les CNNs ont été introduit pour la première fois en 1988 [31]. Les CNN convoluent une matrice de noyau pondérée sur l'entrée, comme le montre la figure 1.6, au lieu de connecter entièrement tous les neurones. Cette pratique a permis au réseau de bien travailler avec les images et de réaliser des tâches de reconnaissance des formes. En raison de la technologie de cette époque, le calcul était difficile pour les réseaux complexes et les CNN étaient principalement utilisés pour des problèmes de jouets. Une avancée significative a eu lieu en 1998 avec l'introduction de la descente de gradient pour l'apprentissage des réseaux par [32]. Cela a fourni la base pour mettre à jour le réseau et les poids de biais d'une manière légère et efficace sur le plan informatique. Bien que les réseaux CNN soient restés lourds en termes de calcul à l'époque, il s'agissait d'une avancée significative dans la formation des réseaux CNN modernes. L'attention consacrée aux CNN a augmenté de façon spectaculaire lorsque [33] a surpassé les techniques traditionnelles de traitement d'images sur le jeu de données ILSVRC-2012. Depuis lors, de nombreux développements supplémentaires ont eu lieu pour optimiser ces réseaux.

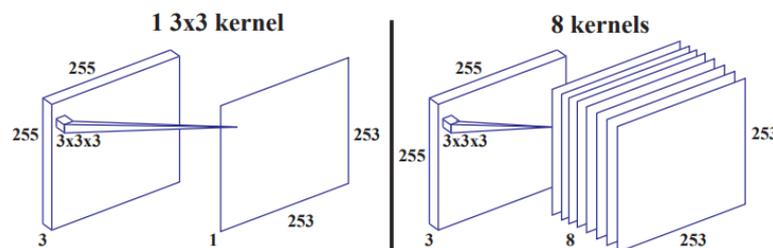


FIGURE 1.6 – La partie gauche est un exemple de convolution 3x3 avec un seul filtre. La dimension supplémentaire sur la convolution est la profondeur de l'entrée. Le côté droit est une convolution 3x3 avec 8 couches de filtres. Notez que chaque filtre a des poids de noyau individuels entraînaables [5].

Les réseaux de neurones convolutifs sont actuellement l'un des algorithmes les plus importants pour l'apprentissage en profondeur avec des données d'image. Alors que pour l'apprentissage automatique traditionnel, les fonctionnalités pertinentes doivent être extraites manuellement, l'apprentissage en profondeur uti-

lise des images brutes comme entrée pour extraire certaines caractéristiques. Les CNN se composent d'une couche d'entrée et de sortie et de plusieurs couches cachées entre l'entrée et la sortie. Des exemples de couches intermédiaires sont les couches convolutives, les couches max-pooling et les couches entièrement connectées. La figure 1.7 montre la différence entre les réseaux de neurones convolutifs et les réseaux de neurone traditionnel.

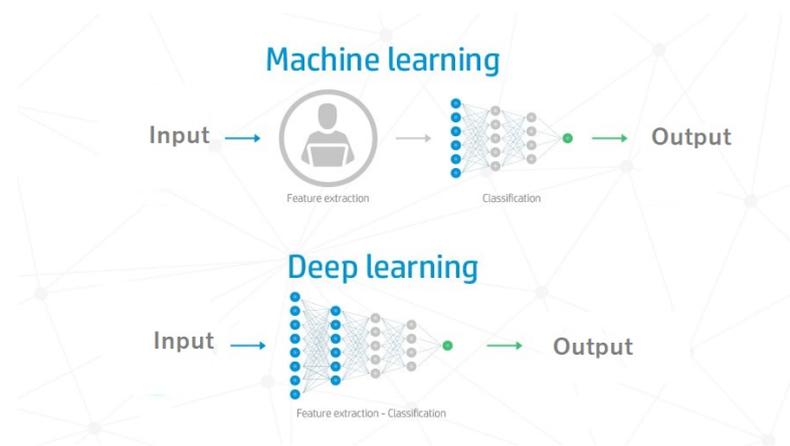


FIGURE 1.7 – Comparaison entre l'apprentissage automatique et l'apprentissage en profondeur

## 1.4.2 Architecture des CNNs

Les CNN sont un type d'algorithme d'apprentissage en profondeur utilisé pour traiter les données avec une topologie en forme de grille. Les CNN sont un type d'algorithme d'apprentissage en profondeur utilisé pour traiter des données ayant une relation spatiale ou temporelle. Les CNN sont similaires aux autres réseaux de neurones, mais ils ont une couche supplémentaire de complexité en raison du fait qu'ils utilisent une série de couches convolutionnelles. Les couches convolutives sont un composant essentiel des réseaux de neurones convolutifs (CNN). L'image ci-dessous représente une architecture CNN typique.

Dans cette partie, nous décrivons les différentes couches utilisées pour créer des réseaux de neurones convolutifs.

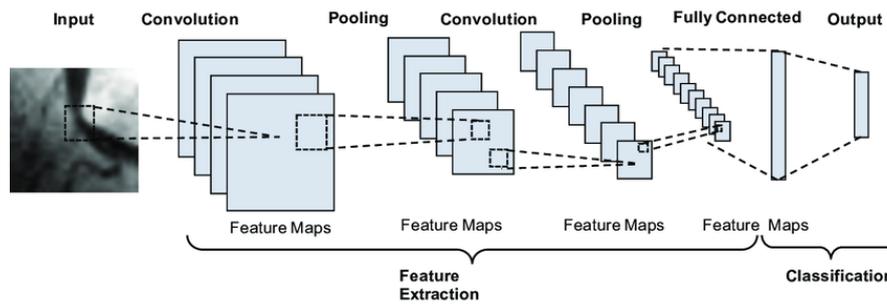


FIGURE 1.8 – Architecture d'un réseau neuronal convolutif (CNN) typique [6].

### Couche de convolution

Les couches de convolutions sont des éléments essentiels dans les réseaux de neurones convolutifs (CNN). Elles sont cohérentes dans l'application d'un filtre sur une information donnée. L'utilisation répétée d'un filtre similaire sur cette information produit une carte d'activations appelée carte de caractéristiques, qui met en évidence les zones et la qualité des caractéristiques distinctes présentes dans cette information, comme une image par exemple.

Dans la figure 1.9, une multiplication par produit scalaire est effectuée entre une matrice de filtre de taille  $3 \times 3$  et une zone de taille  $3 \times 3$  de la matrice de l'image d'entrée. Les éléments de la matrice résultante sont ensuite ajoutés pour obtenir la valeur de sortie correspondant à la carte des caractéristiques. Le filtre est ensuite déplacé sur la matrice d'entrée, répétant ainsi la multiplication du produit scalaire avec chaque combinaison restante de zones de taille  $3 \times 3$ , afin de compléter la carte de caractéristiques. Plusieurs filtres sont utilisés pour une entrée, et les cartes de caractéristiques résultantes sont combinées pour obtenir la sortie finale d'une couche convolutive.

Il existe deux autres concepts importants dans les couches convolutionnelles [7] : le pas (stride) et le rembourrage (padding). Le pas correspond au nombre de pixels que le noyau ou le filtre se déplace sur la matrice d'entrée. Le rembourrage, quant à lui, consiste à ajouter des zéros aux bords du volume d'entrée. La taille de ce rembourrage est un hyperparamètre supplémentaire. Le rembourrage permet de contrôler la dimension spatiale du volume de sortie, et il peut être souhaitable de conserver la même surface que celle du volume d'entrée dans certains cas.

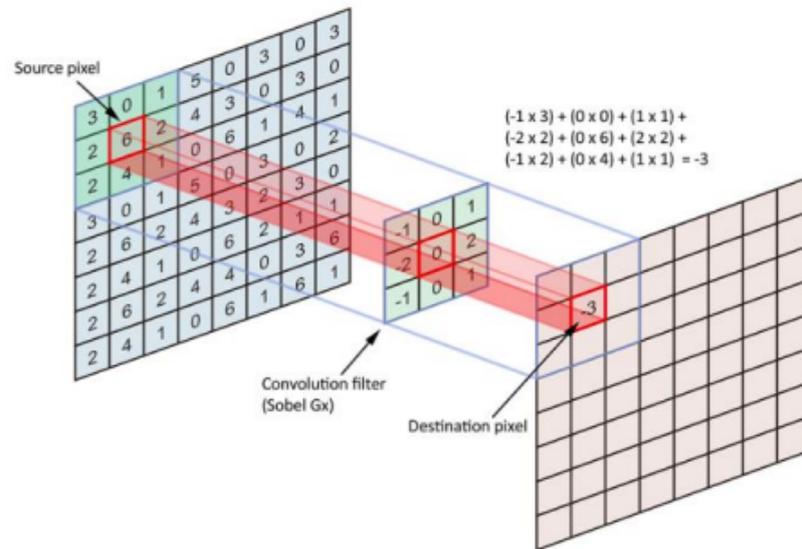


FIGURE 1.9 – Processus de convolution. Le filtre ici est de taille 3x3, Le filtre glisse sur l’entrée et exécute sa sortie sur la nouvelle couche. [7].

### Couche d’activation Relu

Chaque couche de convolution d’un CNN est généralement suivie d’une couche d’activation qui applique une fonction non linéaire de manière élémentaire sur les cartes de caractéristiques. Les premiers CNN étaient entraînés avec les fonctions TanH ou Sigmoides, mais les modèles récents utilisent la fonction ReLU qui garantit des temps d’apprentissage plus rapides et une moindre complexité de calcul, comme le souligne [33]. L’objectif de la fonction ReLU est d’apporter la non-linéarité dans les CNNs en appliquant une fonction d’activation sur la carte de caractéristique de sortie en conservant la même dimension. La figure 1.10, explique le fonctionnement principal de la fonction d’activation ReLU en mettant les valeurs négatives à 0 et en conservant les valeurs positives.

### Couche de Pooling

Les couches de pooling sont chargées de réduire la dimensionnalité des cartes de caractéristiques ("*feature maps*"), en particulier la hauteur et la largeur, en préservant la profondeur [34]. Cela est bénéfique car cela diminue la puis-

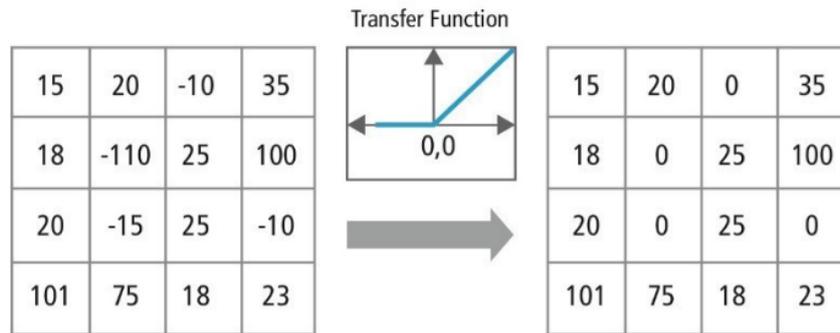


FIGURE 1.10 – Principe de fonctionnement de la fonction d’activation ReLU.

sance de calcul requise et contrôle le sur-apprentissage (*overfitting en anglais*) pour traiter les données, tout en extrayant les caractéristiques dominantes dans les cartes de caractéristiques [8].

La figure 1.11 montre les deux types de couche de pooling les plus fréquents.

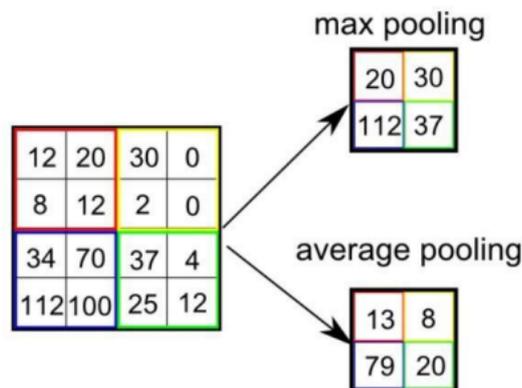


FIGURE 1.11 – Exemple de pooling avec les différentes stratégies (maximum, moyen). La taille de filtre ainsi que le pas utilisé dans cet exemple est de 2x2 [8].

### Couche entièrement connectée

Les couches entièrement connectées sont ceux où la classification se produit réellement. La matrice d’entrée est transformée en un vecteur colonne et est ensuite utilisée comme entrée pour un ensemble de couches entièrement connectées. Ces canapés sont identiques à l’architecture d’un réseau de neurones artificiels entièrement connectés présentés dans la section 1.3.2. Chaque

couche entièrement connectée (appelée couche dense) passe par une fonction d'activation (par exemple tanh ou ReLu), mais la couche dense de sortie passe par la fonction « sigmoïde » ou « softmax » pour la génération des prédictions des classes. Dans la classification multiclasse Softmax, la fonction de perte utilisée est l'entropie croisée (categorical\_crossentropy dans Keras) [34]. La sortie de la fonction Softmax est un vecteur à  $N$  dimensions, où  $N$  est le nombre de classes parmi lesquelles le CNN doit choisir.

### Couche de normalisation

La formation de réseaux de neurones profonds est une tâche complexe qui implique la résolution de plusieurs problèmes. Malgré leur énorme potentiel, ces réseaux peuvent être prêtés et sujets au sur-apprentissage. Par conséquent, la recherche en apprentissage approfondi se concentre constamment sur l'étude de méthodes visant à résoudre ces problèmes. La normalisation par lots (*Batch Normalization (BN)*) est l'une de ces méthodes. Actuellement, c'est une technique largement utilisée dans le domaine de l'apprentissage en profondeur. Il améliore la vitesse d'apprentissage des réseaux de neurones et assure la régularisation, en évitant le sur-apprentissage.

La distribution de l'entrée change pendant la phase d'apprentissage en raison des mises à jour des paramètres qui entraînent un problème de décalage de covariable interne qui ralentit le processus de convergence. Pour s'assurer que le modèle peut continuer à s'entraîner sur les mêmes distributions d'entrée (c'est-à-dire qu'elles ont une moyenne nulle et une variance unitaire) et pour accélérer la convergence du réseau, la normalisation devient alors une étape cruciale. Dans [35], ils ont montré que cette idée peut être simplement réalisée en effectuant une standardisation de chaque mini-lot d'entrée. Cela a pour effet de stabiliser le processus d'apprentissage et de réduire considérablement le nombre d'époques d'entraînement nécessaires pour former des réseaux de neurones profonds.

### 1.4.3 Algorithme d'optimisation

La section précédente a décrit les fondements de l'apprentissage automatique et la manière de trouver une correspondance entre les données d'entrée et les étiquettes de sortie. Cette correspondance est réalisée à l'aide d'une fonction qui utilise une matrice de poids  $W$  et un vecteur de biais  $b$  pour renvoyer une prédiction pour chaque entrée. Nous avons également exploré l'architecture des CNN et les principaux composants qui constituent un réseau de neu-

rones convolutif. Forts de ces connaissances, nous pouvons à présent aborder les techniques d'optimisation permettant au modèle d'apprendre à partir des données.

L'objectif de l'apprentissage automatique est de trouver un ensemble de paramètres, en l'occurrence les poids  $w$  et les biais  $b$ , qui permettent de classer correctement les données d'entrée. Toutefois, la question de la meilleure stratégie à adopter pour atteindre une combinaison optimale de  $W$  et  $b$ , assurant ainsi une précision de classification élevée.

Dans la suite de ce texte, nous présenterons l'algorithme le plus couramment utilisé en apprentissage profond : la méthode de descente de gradient, ainsi que ses variantes basées sur le même principe. Nous explorerons ensuite les améliorations possibles pouvant être apportées aux modèles profonds, aussi connues sous le nom de techniques de régularisation."

## Descente de gradient

L'algorithme de descente du gradient est un algorithme itératif qui permet à la fonction de coût  $C(w, b)$  d'être continuellement minimisée en modifiant les paramètres du modèle  $w$  et  $b$ . Le but est de trouver la combinaison la plus proche du minimum, le gradient  $\Delta(w; b) C(w; b)$  est calculé et Mettez à jour les poids  $w$  et le biais  $b$  dans le sens de la pente de la fonction  $C$ . La figure 1.12 montre une stratégie d'utilisation de cette dérivée pour minimiser la fonction. Nous devons d'abord déterminer les hyperparamètres appelés pas d'apprentissage et définir la longueur de l'étape pour aller dans la direction du gradient. A chaque répétition de l'algorithme d'optimisation, les paramètres changent en fonction de l'importance du neurone. Les paramètres des neurones impliqués dans la reproduction de la grande erreur finale ont changé de manière significative par rapport aux autres neurones qui ne contribuent pas explicitement à l'erreur finale. Par conséquent, pour mettre à jour le modèle, vous devez augmenter ou diminuer les paramètres du modèle.

## Variantes de l'algorithme de descente de gradient

Il existe trois variantes connues de l'algorithme de descente de gradient :

*La descente de gradient par batch* : Consiste à calculer la fonction de coût et son gradient sur la base d'apprentissage. Leur avantage est que le gradient calculé indique la direction correcte de la pente pour minimiser l'erreur globale. Cependant, pour chaque étape, chaque base d'apprentissage doit être évaluée

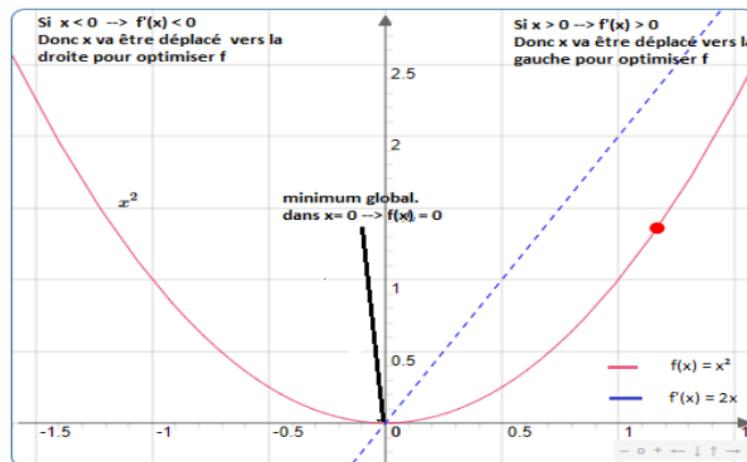


FIGURE 1.12 – Le principe de fonctionnement de l’algorithme de descente de gradient pour trouver la valeur minimale de la fonction  $f$  par la dérivée [9].

pour calculer l’erreur dans chaque instance, ce qui peut être très difficile à exécuter sur une grande base de données.

*descente de gradient stochastique* : Cette méthode calcule les erreurs dans les échantillons les uns des autres par sélection aléatoire, puis met à jour les paramètres après chaque évaluation. Cette stratégie permet un ajustement plus rapide des paramètres du réseau et donc une convergence plus rapide. Cependant, la modification des paramètres de chaque instance peut entraîner un minimum local. De plus, si la base d’apprentissage comporte des exemples compliqués ou des exemples incorrects (par exemple, une annotation incorrecte), le sens du gradient peut être fait dans le mauvais sens, mais il peut être corrigé par les exemples corrects qui suivent. Ce problème est résolu par la troisième variante de descente de gradient appelée gradient par mini-batch.

*gradient par mini-batch* : Il s’agit de calculer les erreurs par petits lots d’une taille prédéterminée par l’utilisateur, puis de calculer l’erreur moyenne, et enfin de réinitialiser les paramètres. L’évaluation par mini-lot permet de trouver un gradient plus précis sans avoir à parcourir l’ensemble des données pour ajuster les paramètres.

Il existe d’autres méthodes dans l’algorithme de descente de gradient qui permettent une convergence plus rapide à la fonction de coût minimum local. Ci-dessous on peut citer Momentum [36], NAG (Nestrov Accelerated Gra-

dient) [37], Adadelta [38], RMSprop [39] et Adam (estimation adaptative du moment) [40].

#### 1.4.4 Régularisation

Comme nous l'avons évoqué dans les sections précédentes, l'optimisation vise à minimiser l'erreur générée durant la phase d'apprentissage (erreur d'apprentissage) en ajustant les paramètres du modèle, jusqu'à ce qu'une combinaison optimale soit trouvée. Dans l'apprentissage automatique, l'objectif principal est de concevoir un modèle capable de bien généraliser, c'est-à-dire de bien performer non seulement sur les données d'entraînement, mais aussi sur de nouvelles données inédites. Si l'erreur sur le jeu de test est supérieure à l'erreur d'apprentissage, cela peut indiquer un surapprentissage. À l'inverse, si l'erreur de test est également élevée, le modèle peut souffrir de sous-apprentissage. Plusieurs méthodes de régularisation existent pour contrôler l'apprentissage des modèles dans le but d'éviter le surapprentissage et de minimiser l'erreur sur le jeu de test. Parmi les stratégies pour régulariser les réseaux de neurones profonds, on peut citer la modification de l'architecture, par exemple en ajoutant des couches spécifiques telles que Dropout. Ces couches sont conçues pour réduire la complexité du modèle et améliorer sa capacité de généralisation. Le Dropout fonctionne en désactivant aléatoirement certains neurones et leurs connexions, empêchant ainsi le réseau de dépendre excessivement de neurones spécifiques et forçant tous les neurones à participer à l'apprentissage. D'autres techniques de régularisation, telles que l'augmentation des données et l'arrêt précoce (Early Stopping), sont également utilisées durant l'apprentissage. Il existe enfin des méthodes de régularisation agissant directement sur la fonction de coût, en y ajoutant un terme de pénalisation qui contraint l'entraînement du modèle sur les données d'apprentissage. Ce terme de régularisation pénalise les paramètres qui attribuent une importance excessive à des détails spécifiques des données d'entrée. Les méthodes L2 et L1 sont deux exemples couramment utilisés de ce type de régularisation.

##### *La régularisation L2 et L1 :*

En raison de l'ajout de ce terme de régularisation, les valeurs des matrices de poids diminuent car cela suppose qu'un réseau de neurones avec des matrices de poids plus petites conduit à des modèles plus simples. Par conséquent, cela réduira également le sur-apprentissage dans une certaine mesure. Cependant, ce terme de régularisation diffère en L1 et L2.

En L2, nous avons :

$$Cost\ function = Loss + \frac{\lambda}{2m} * \sum \| w \|^2 \quad (1.16)$$

Ici,  $\lambda$  est le paramètre de régularisation. C'est l'hyperparamètre dont la valeur est optimisée pour de meilleurs résultats. La régularisation L2 est également connue sous le nom de décroissance des poids car elle force les poids à décroître vers zéro (mais pas exactement zéro).

En L1, nous avons :

$$Cost\ function = Loss + \frac{\lambda}{2m} * \sum \| w \| \quad (1.17)$$

Dans ce cas, nous pénalisons la valeur absolue des poids. Contrairement à L2, les poids peuvent être réduits à zéro ici. Par conséquent, il est très utile lorsque nous essayons de compresser notre modèle. Sinon, nous préférons généralement la L2 à celle-ci.

### 1.4.5 Différents types d'architectures CNN

Dans les sections précédentes, nous avons présenté les éléments de base des réseaux de neurones convolutifs, leur apprentissage, ainsi que des stratégies pour améliorer leurs performances. Par la suite, nous passerons en revue les modèles de pointe les plus populaires qui ont largement contribué au développement du domaine de la reconnaissance d'images. Ces modèles apportent leurs succès à un concours annuel appelé ILSVRC (ImageNet Large Scale Visual Recognition Challenge) [9], qui utilise la base de données ImageNet [41] pour évaluer différentes architectures. La base de données ImageNet contient plus de 14 millions d'images annotées avec 1000 catégories d'objets tirées de la base lexicale WordNet.

#### LeNet :

LeNet est la première architecture CNN. Il a été développé en 1998 par Yann LeCun, Corinna Cortes et Christopher Burges [10] pour des problèmes de reconnaissance de chiffres manuscrits. LeNet a été l'un des premiers CNN réussis et est souvent considéré comme le "Hello World" de l'apprentissage profond. C'est l'une des premières architectures CNN et l'une des plus utilisées. Elle a été appliquée avec succès à des tâches telles que la reconnaissance de chiffres manuscrits. L'architecture LeNet se compose de plusieurs couches de convolution et de mise en commun, suivies d'une couche entièrement connectée. Le

modèle comporte cinq couches de convolution suivies de deux couches entièrement connectées. LeNet a été le début des CNN dans l'apprentissage profond pour les problèmes de vision par ordinateur. Cependant, LeNet ne pouvait pas s'entraîner correctement en raison du problème du gradient de disparition. Pour résoudre ce problème, une couche de connexion raccourcie connue sous le nom de max-pooling est utilisée entre les couches convolutionnelles pour réduire la taille spatiale des images, ce qui permet d'éviter le sur-apprentissage et permet aux CNN de s'entraîner plus efficacement. Le diagramme ci-dessous représente l'architecture de LeNet-5 (figure 1.13). Le CNN LeNet est un modèle simple mais puissant qui a été utilisé pour diverses tâches telles que la reconnaissance de chiffres manuscrits, la reconnaissance de panneaux de signalisation et la détection de visages. Bien que LeNet ait été développé il y a plus de 20 ans, son architecture est toujours pertinente aujourd'hui et continue d'être utilisée.

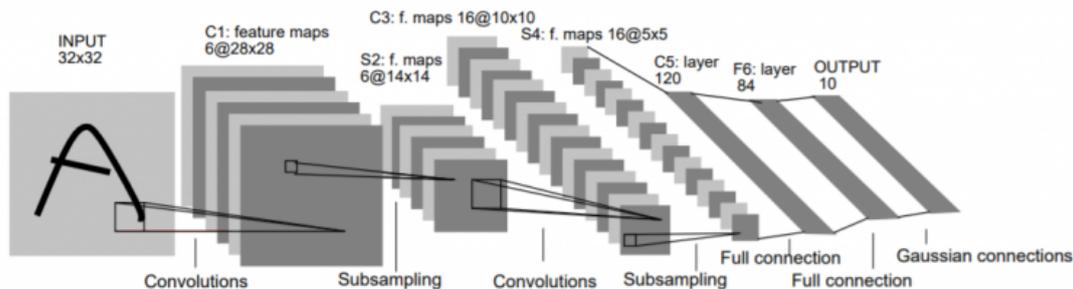


FIGURE 1.13 – Architecture du réseau LeNet5 [10].

### AlexNet :

AlexNet est l'architecture d'apprentissage profond qui a popularisé le CNN. Elle a été développée par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton [33]. Le réseau AlexNet avait une architecture très similaire à celle de LeNet, mais était plus profond, plus grand et comportait des couches convolutives empilées les unes sur les autres. AlexNet a été le premier réseau CNN à grande échelle et a été utilisé pour remporter le concours ImageNet en 2012. Comme montré dans la figure 1.14, AlexNet se compose de 5 couches convolutionnelles avec une combinaison de couches de clustering maximum, de 3 couches entièrement connectées et de 2 couches de dropout. La fonction d'activation utilisée dans toutes les couches est Relu. La fonction d'activation utilisée dans la couche de

sortie est Softmax. Le nombre total de paramètres dans cette architecture est d'environ 60 millions.

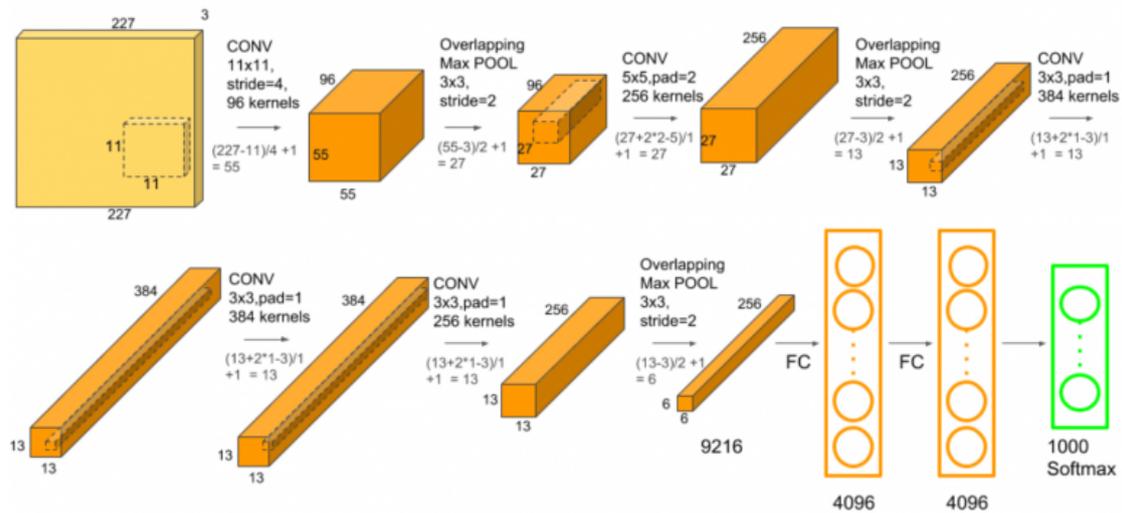


FIGURE 1.14 – Architecture du réseau de neurone convolutif AlexNet [11].

### ZF Net :

ZFnet est l'architecture CNN qui utilise une combinaison de couches entièrement connectées et de CNN. ZF Net a été développé par Matthew Zeiler et Rob Fergus [42]. Il a été le gagnant de l'ILSVRC 2013. Le réseau a relativement moins de paramètres qu'AlexNet, mais il est toujours plus performant dans la tâche de classification ILSVRC 2012 en atteignant la meilleure précision avec seulement 1000 images par classe. Ce réseau a été amélioré par rapport à AlexNet en modifiant les hyperparamètres de l'architecture, notamment en augmentant la taille des couches convolutionnelles intermédiaires et en réduisant la taille du pas et du filtre de la première couche. L'architecture ZF Net CNN se compose de cinq couches convolutives partagées, ainsi que de couches de Pooling-max, de couches de Dropout et de trois couches entièrement connectées. Dans la première couche, ils ont utilisé un filtre de taille 77 et une valeur de pas inférieure. La couche softmax est la dernière couche du ZFNet. Ce modèle CNN est plus efficace en termes de calcul que AlexNet en introduisant une étape d'inférence approximative par le biais de couches de déconvolution au milieu des CNN.

## GoogLeNet :

GoogLeNet (ou Inception) a été proposé par la recherche de Google (avec la collaboration de diverses universités) en 2014 dans le document de recherche intitulé "Going Deeper with Convolutions" [12]. Cette architecture a remporté le concours de classification d'images ILSVRC 2014. Il a fourni une diminution significative du taux d'erreur par rapport aux gagnants précédents Alex-Net (vainqueur de l'ILSVRC 2012) et ZF-Net (vainqueur de l'ILSVRC 2013) et un taux d'erreur nettement inférieur à celui de VGG (finaliste de 2014). Cette architecture utilise des techniques telles que les convolutions  $1 \times 1$  au milieu de l'architecture et la mise en commun de la moyenne globale. GoogLeNet est un réseau de neurones conçu par Google qui possède un module spécialisé appelé 'Inception Modul'. Ce module aide le réseau à apprendre plus efficacement en limitant la quantité de paramètres dans le réseau à 4 millions au lieu de 60 millions (comme dans AlexNet). Ce module est composé de quatre chemins parallèles, comme le montre la figure 1.15, puis les résultats sont concaténés et utilisés par le module suivant. La figure 1.16 montre l'architecture complète du réseau GoogLeNet. La dernière version de GoogLeNet est inception-V4.

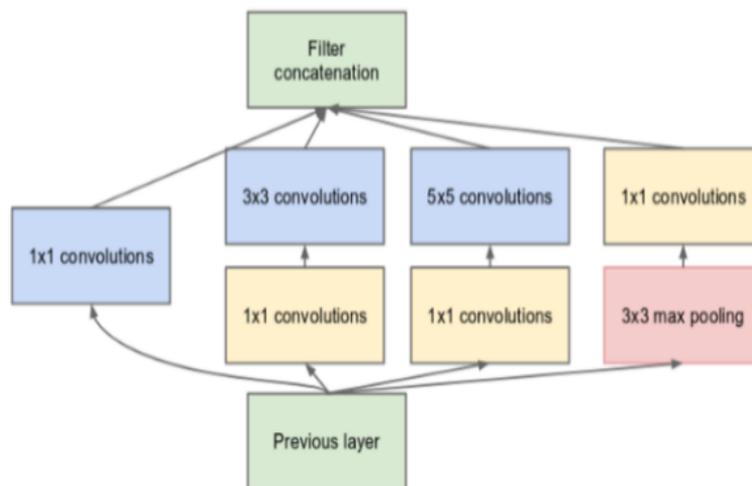


FIGURE 1.15 – Module inception [12]. Il se compose de quatre chemins convolutifs avec des noyaux de tailles différentes.

## VGGNet :

VGGNet est l'architecture CNN développée par Karen Simonyan, Andrew Zisserman et al [43]. à l'Université d'Oxford. VGGNet est un CNN à 16 couches

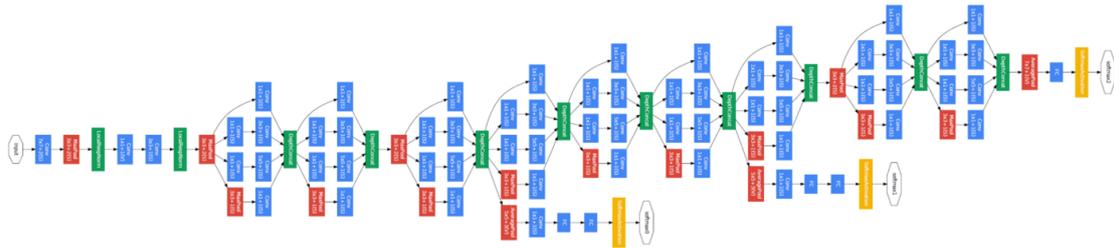


FIGURE 1.16 – Architecture GoogLeNet avec toutes les couches et les caractéristiques [12].

avec 95 millions de paramètres et formé sur plus d'un milliard d'images (1000 classes). Il peut prendre de grandes images d'entrée de 224 x 224 pixels pour lesquelles il dispose de 4096 caractéristiques convolutives. Les CNN avec des filtres aussi grands sont coûteux à former et nécessitent beaucoup de données, ce qui est la principale raison pour laquelle les architectures CNN comme GoogLeNet (architecture AlexNet) fonctionnent mieux que VGGNet pour la plupart des tâches de classification d'images où les images d'entrée ont une taille comprise entre 100 x 100- pixels et 350 x 350 pixels. Le modèle VGG est efficace sur le plan informatique et sert de base solide pour de nombreuses applications de vision par ordinateur en raison de son applicabilité à de nombreuses tâches, y compris la détection d'objets. Ses représentations de caractéristiques profondes sont utilisées sur plusieurs architectures de réseaux neuronaux comme YOLO, SSD, etc. Le diagramme ci-dessous représente le diagramme d'architecture de réseau VGG16 standard :

**ResNet :**

Residual Network ou ResNet, est une forme du réseau de neurones développé par Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun dans leur article "Deep Residual Learning for Image Recognition" publié en 2015 [14]. Les modèles ResNet ont connu un succès incroyable, comme en témoignent les éléments suivants :

- 1) Avec un taux d'erreur parmi les 5 premiers de 3,57%, a remporté la première place du concours de classification ILSVRC 2015 (un modèle d'ensemble)
- 2) A remporté la première place dans ImageNet Détection, ImageNet Localization, Coco Détection et Coco Segmentation aux concours ILSVRC et COCO 2015.

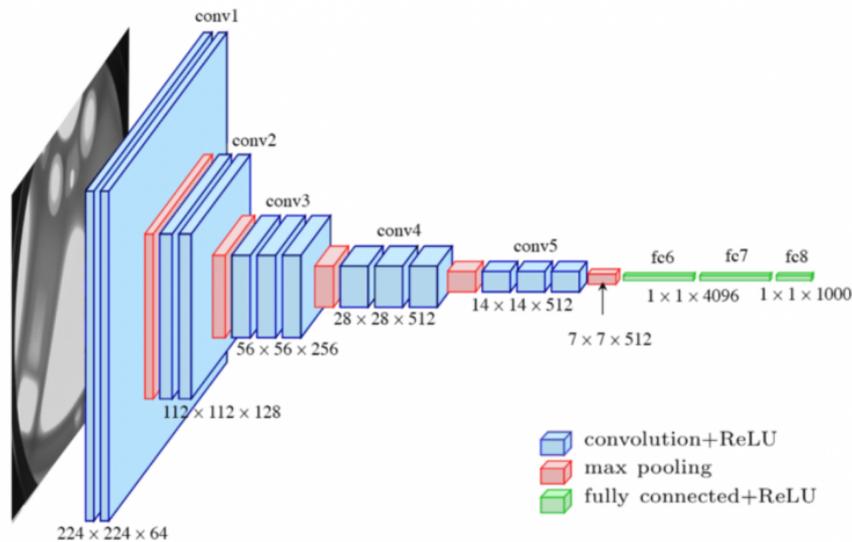


FIGURE 1.17 – Architecture VGG16 [13].

- 3) ResNet-101 est utilisé pour remplacer les couches VGG-16 dans Faster R-CNN. Ils ont constaté une amélioration de 28 % en termes relatifs.

Les chercheurs utilisent des connexions résiduelles dans leur architecture, notamment en ajoutant la sortie d'une couche à son entrée (voir Figure 1.18). Ces couches résiduelles peuvent éviter le problème de disparition de gradient lié à la profondeur du réseau, c'est-à-dire que lorsque la profondeur dans le réseau résiduel augmente, l'efficacité est supérieure qu'un simple réseau de neurones convolutifs qui n'utilise pas de connexions résiduelles.

### InceptionResNetV2

InceptionResNetV2 est une architecture de modèle d'apprentissage en profondeur qui combine les concepts des modèles Inception et ResNet. Il a été développé par Google [44] et est largement utilisé pour diverses tâches de vision par ordinateur, telles que la classification d'images et la détection d'objets. L'architecture InceptionResNetV2 s'appuie sur le modèle Inception, connu pour son utilisation efficace des ressources informatiques en utilisant plusieurs couches de convolution parallèles de différentes tailles. Cela permet au modèle de capturer à la fois des caractéristiques locales et globales dans une image. De plus, InceptionResNetV2 intègre des connexions résiduelles du modèle ResNet. Les connexions résiduelles aident à résoudre le problème des gradients qui dispa-

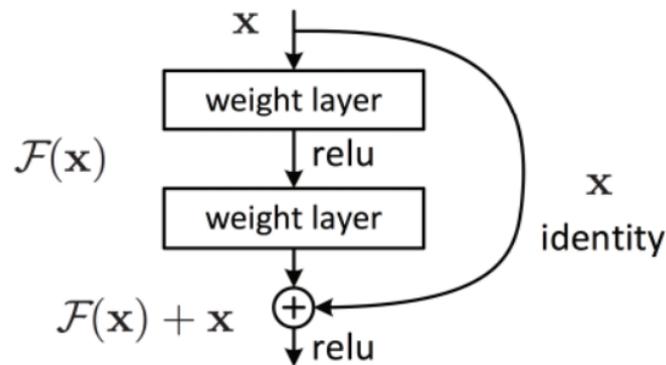


FIGURE 1.18 – Le principe de connexion résiduelle [14].

raissent dans les réseaux neuronaux profonds, permettant ainsi une formation plus facile des modèles plus profonds. En combinant les forces d’Inception et de ResNet, InceptionResNetV2 atteint des performances de pointe sur divers ensembles de données de référence dans le domaine de la vision par ordinateur. Il a été largement adopté dans la recherche et l’industrie pour des tâches telles que la reconnaissance d’images, la détection d’objets et la segmentation d’images. Dans l’ensemble, InceptionResNetV2 est une architecture puissante d’apprentissage en profondeur qui exploite les forces d’Inception et de ResNet pour atteindre une précision et une efficacité élevées dans les tâches de vision par ordinateur.

### Xception :

Le modèle Xception [45] a été proposé en tant qu’architecture CNN en 2017. Xception a 36 couches de convolutions. Il se compose de trois flux. Le premier flux est le flux d’entrée qui comporte des couches de convolution, de convolution séparable et de mise en commun. Le deuxième flux est le flux intermédiaire qui a des couches de convolution séparables. Et le flux du milieu se répète 8 fois. Le troisième flux est le flux de sortie. C’est le dernier flux et donne le résultat avec la couche dense. Le modèle Xception peut être vu sur la Fig. 1.19. La sélection de l’algorithme Xception est que l’algorithme Xception est très efficace et a donné de très bons résultats dans des recherches antérieures (Chen et al., [46] [47]; Shaheed et al, [48] ). Ainsi, il est utilisé pour le problème de détection COVID-19.

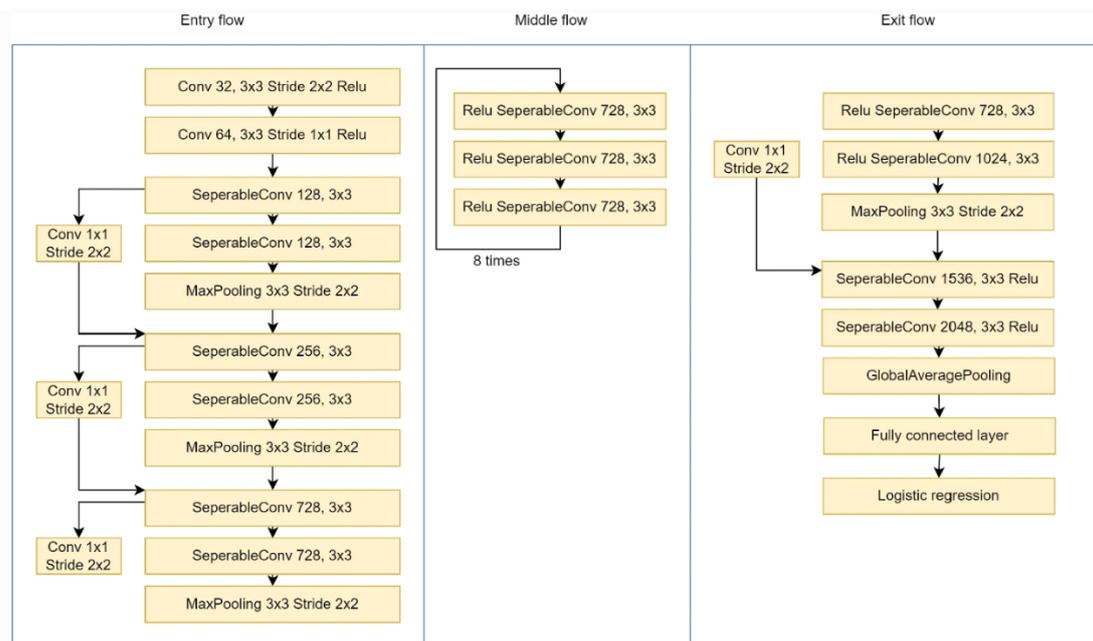


FIGURE 1.19 – Architecture Xception standard [15].

## 1.5 Explication des décisions des modèles d'apprentissage en profondeur

### 1.5.1 Les méthodes de visualisation

Afin de renforcer la confiance dans les modèles d'apprentissage automatique et de progresser vers leur intégration dans notre vie quotidienne, nous devons utiliser des modèles « transparents » qui pourraient expliquer pourquoi ils prédisent ce qu'ils prédisent. À cette fin, de nombreux chercheurs ont proposé plusieurs méthodes qui nous aident à mieux comprendre le fonctionnement de ces modèles de « boîte noire ». Dans cette partie, nous passerons en revue certaines des méthodes d'interprétation les plus fréquemment utilisées en apprentissage profond, l'un des domaines de recherche fructueux en apprentissage automatique. Comme il existe de nombreuses méthodes d'interprétation tel que Cam, Grad-Cam, Grad-Cam+.

#### Class activation map (CAM)

La carte d'activation de classe (CAM) est une méthode d'explication utilisée pour les CNN, qui a été proposée par [16]. Les auteurs de l'article ont évalué

des réseaux ayant une architecture similaire à celle du réseau "Network in Network". Dans ces réseaux, la pile de couches entièrement connectées à la fin du modèle a été remplacée par une couche appelée Global Average Pooling (GAP). Le GAP calcule simplement la moyenne des activations de chaque carte de caractéristiques, les concatène et les produits sous forme de vecteur. Ensuite, une somme pondérée de ce vecteur est transmise à la couche de perte softmax finale. En utilisant cette architecture, nous pouvons mettre en évidence les régions importantes de l'image en projetant les poids de la sortie sur les cartes de caractéristiques convolutionnelles. La figure suivante illustre ce processus :

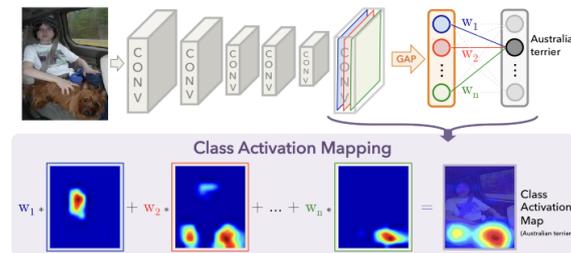


FIGURE 1.20 – Le processus de La carte d'activation de classe (Cam) [16].

### La carte d'activation de classe pondérée par le gradient (Grad-CAM)

Grad-CAM [49] est une version plus polyvalente de CAM qui peut produire des explications visuelles pour n'importe quel CNN arbitraire, même si le réseau contient également une pile de couches entièrement connectées (par exemple, les réseaux VGG). Grad-CAM utilise le gradient cible de la couche convolutive finale pour générer une carte thermique mettant en évidence les régions importantes de l'image.

La méthode Grad-CAM est une technique de visualisation populaire pour comprendre comment les réseaux de neurones convolutionnels sont amenés à prendre des décisions de classification. Il est propre à chaque classe, ce qui signifie qu'il peut générer des visualisations distinctes pour chaque classe présente dans l'image. En cas d'erreur de classification, cette approche est utile pour comprendre où se situe le problème dans le réseau convolutif. Cela permet également à l'algorithme d'être plus transparent.

Grad-CAM implique de découvrir quelles parties de l'image conduisent le réseau neuronal convolutif à prendre la décision finale. Le procédé comprend la génération d'une carte thermique représentant des classes d'activation sur une image reçue en entrée. Les classes d'activation sont associées à des classes de

sortie spécifiques. Ces classes peuvent indiquer l'importance de chaque pixel par rapport à la classe pertinente en augmentant ou en diminuant l'intensité du pixel. Par exemple, si des images sont utilisées dans un réseau convolutif de cas sains (négative) et pathologiques (positive), la visualisation de Grad-CAM peut générer des cartes thermiques pour la classe "negative", montrant à quel point différentes parties de l'image correspondent aux cas normal ou bonne santé, et pour la classe "positive", montrant la façon dont certaines parties correspondent aux cas pathologiques.

Dans cette thèse, nous proposons une technique visant à expliquer les réseaux de neurones convolutifs, qui sont des architectures de pointe, en particulier pour les tâches liées à l'image. La recherche dans le domaine de l'apprentissage automatique interprétable s'accélère et s'avère inestimable pour gagner la confiance des utilisateurs et contribuer à éclaircir les modèles.

### Grad-CAM++

Grad-CAM++ est une généralisation de la méthode Grad-CAM introduite par Chattopadhyay et al. Dans [50]. Ses auteurs soutiennent que Grad-CAM échoue souvent à mettre en évidence correctement l'objet entier ou tous les objets d'une image avec plusieurs occurrences de la même classe.

L'équation 1.18 montre que Grad-CAM calcule le poids  $w_k^c$  de la carte d'activation  $A^k$  pour la classe  $c$  comme une moyenne non pondérée de tous les gradients de pixels.

$$w_k^c \Rightarrow \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \quad (1.18)$$

Traiter chaque pixel de manière égale lors du calcul de l'importance d'une carte d'activation peut supprimer les cartes d'activation avec une empreinte spatiale relativement moindre, comme le montre la figure 2.1. Dans cet exemple hypothétique avec les cartes d'activation  $A^1$ ,  $A^2$ ,  $A^3$ , Grad-CAM attribue des poids inférieurs aux cartes d'activation  $A^2$ ,  $A^3$  en raison de leur empreinte spatiale plus petite, bien que leurs pixels aient la même importance que les pixels de  $A^1$ .

Grad-CAM++ résout ce problème en introduisant des poids pixel par pixel  $\alpha_{ij}^{kc}$  pour la carte d'activation  $k$ , la classe d'intérêt  $c$  et l'emplacement spatial  $(i, j)$ . Les poids  $w_k^c$  sont alors donnés par :

$$w_k^c \Rightarrow \sum_i \sum_j \alpha_{ij}^{kc} ReLU\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right). \quad (1.19)$$

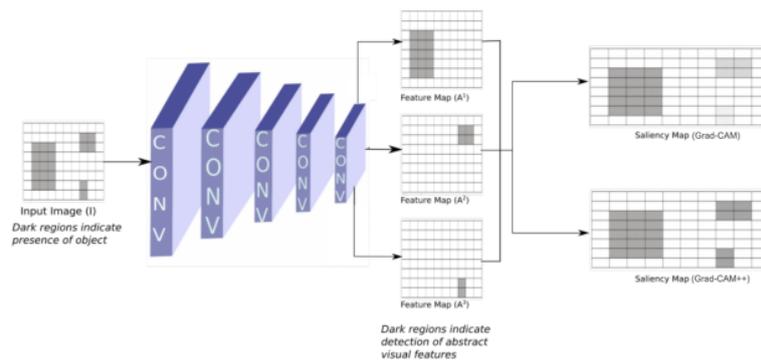


FIGURE 1.21 – Un aperçu des deux méthodes, Grad-CAM et GradCAM++. La nuance de gris dans les grilles montre les attributions calculées des pixels correspondants. En haut à droite : Grad-CAM ne parvient pas à mettre correctement en évidence tous les pixels importants. En bas à droite : Grad-CAM++ attribue correctement la même importance aux pixels de toutes les cartes d'activation. [17].

## 1.6 Conclusion

Dans ce chapitre, nous avons exposé les concepts fondamentaux des réseaux de neurones convolutifs (CNN). Tout d'abord, nous avons présenté les trois principaux types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé. Dans le cadre de cette thèse, nous avons principalement utilisé l'apprentissage supervisé en utilisant des étiquettes de classe.

Ensuite, nous avons abordé les principes des réseaux de neurones artificiels et mis en lumière leurs liens avec les réseaux biologiques. Dans cette section, nous avons également introduit les divers types de fonctions d'activation. Pour résoudre des problèmes non linéaires, il est essentiel d'utiliser à la fois des fonctions d'activation non linéaires et des réseaux de neurones multicouches.

Après avoir posé les bases de l'apprentissage automatique et des réseaux de neurones artificiels, ainsi que de leurs mécanismes d'apprentissage, nous avons ensuite exploré les concepts fondamentaux des réseaux de neurones convolutifs dans la section suivante. Nous avons commencé par présenter les différents éléments nécessaires à la construction d'un réseau de neurones convolutifs, ainsi que les algorithmes utilisés pour ajuster les paramètres d'apprentissage.

Par la suite, nous avons abordé la question de la régularisation, définie comme une technique visant à améliorer la précision des tests. Nous avons mis l'accent

sur quelques architectures CNN populaires, pouvant être exploitées pour élaborer des modèles spécialisés adaptés à notre tâche spécifique.

Enfin, ce chapitre s'est conclu par une revue des techniques de visualisation. Ces méthodes nous permettent non seulement de comprendre ce que le modèle a appris à identifier dans les images, mais aussi de déterminer les domaines où ses performances peuvent être améliorées. Ces informations sont cruciales pour évaluer la capacité des modèles à être interprétés, ce qui revêt une grande importance dans des contextes d'utilisation réelle.

# Chapitre 2

## Classification des images radiologiques par l'apprentissage en profondeur

### 2.1 Introduction

La pneumonie est une infection qui affecte un ou les deux poumons, généralement causée par des bactéries, des virus, voire des champignons. Cette maladie peut présenter une variété d'infections potentiellement mortelles. Les symptômes courants de la pneumonie comprennent des douleurs musculaires et des maux de tête. Le diagnostic précis de la pneumonie est une tâche complexe et exigeante, principalement réservée aux radiologues experts.

Pour améliorer la précision et l'efficacité de la détection de la pneumonie, les réseaux de neurones convolutifs, également connus sous l'acronyme CNN (Convolutional Neural Network), sont largement utilisés. Ces réseaux sont réputés pour leur capacité à traiter rapidement et avec une grande précision les images médicales, en particulier les images de grande taille. Ils sont spécialement conçus pour extraire automatiquement les caractéristiques cruciales des images d'entrée.

L'apprentissage profond basé sur les CNN est devenu la méthode privilégiée au cours de la dernière décennie pour la sélection et la reconnaissance des motifs dans les images médicales. C'est pourquoi nous avons opté pour cette architecture afin d'aider les radiologues à détecter automatiquement la pneumonie à partir des images de radiographie thoracique, améliorant ainsi le processus de diagnostic médical.

## 2.2 Images radiologiques

La radiographie du thorax, également connue sous le nom de Chest X-Ray (CXR), est une technique médicale qui permet d'obtenir des images des structures internes de la cage thoracique en utilisant des rayons X. Chaque organe à l'intérieur de la cavité thoracique retient différemment les radiations en fonction de son épaisseur, créant ainsi des ombres distinctes sur le film radiographique [51].

Les images radiographiques du thorax sont caractérisées par un fort contraste, où la luminosité et la netteté varient selon les structures observées. Par exemple, les os de la cage thoracique, tels que les côtes et les vertèbres, absorbent davantage de rayonnement et apparaissent donc plus blancs sur l'image. En revanche, le tissu pulmonaire, composé en grande partie d'air, permet le passage de la plupart des radiations, donnant ainsi une apparence plus sombre sur le film. Le cœur et l'aorte apparaissent également plus clairs, bien que généralement moins lumineux que les os, en raison de leur densité différente.

Les médecins prescrivent des radiographies thoraciques pour une variété de raisons médicales, car cet examen radiologique de base permet d'évaluer de nombreuses conditions cliniques. Parmi les affections couramment détectées sur une radiographie thoracique, on peut citer :

- La pneumonie,
- L'hypertrophie cardiaque,
- L'insuffisance cardiaque congestive,
- Les masses pulmonaires,
- Les fractures des côtes,
- L'accumulation de liquide autour du poumon (appelée épanchement pleural),
- L'accumulation d'air autour du poumon (pneumothorax).

Dans l'ensemble, l'examen radiographique thoracique est une procédure simple, rapide, relativement peu coûteuse et présentant un risque minime d'exposition aux radiations. De plus, il est largement accessible en pratique médicale.

## 2.3 Images radiologiques du poumon

La radiographie thoracique est une procédure qui utilise une faible dose de rayonnement ionisant pour obtenir des images de l'intérieur de la poitrine. Elle est couramment utilisée pour évaluer les poumons, le cœur et la paroi thoracique, et peut aider au diagnostic de symptômes tels que l'essoufflement, la toux persistante, la fièvre, les douleurs thoraciques ou les blessures. De plus, elle peut être utile pour diagnostiquer et surveiller le traitement de diverses affections pulmonaires telles que la pneumonie, l'emphysème et le cancer. En raison de sa rapidité et de sa simplicité, la radiographie thoracique est particulièrement utile dans les situations d'urgence. Il s'agit de l'examen radiographique diagnostique le plus fréquemment réalisé. Lors d'une radiographie pulmonaire, des images du cœur, des poumons, des voies respiratoires, des vaisseaux sanguins, de la colonne vertébrale et de la poitrine sont obtenues.

Les rayons X sont la forme d'imagerie médicale la plus ancienne et la plus utilisée. Ils permettent aux médecins de diagnostiquer et de traiter diverses affections médicales en exposant le patient à une faible dose de rayonnement ionisant pour obtenir des images de l'intérieur du corps. La radiographie thoracique est généralement le premier test d'imagerie utilisé pour aider au diagnostic de symptômes tels que :

- Des difficultés respiratoires
- Une toux persistante
- Des douleurs ou des blessures thoraciques, ou encore de la fièvre.

Les médecins utilisent cet examen pour diagnostiquer ou surveiller le traitement de conditions telles qu'une infection pulmonaire bactérienne, virale ou fongique, une inflammation des bronches, une insuffisance cardiaque, l'emphysème, le cancer du poumon, le positionnement des dispositifs médicaux, ou encore l'accumulation de liquide ou d'air autour des poumons.

## 2.4 Détection de la pneumonie dans les images radiographiques pulmonaires à l'aide d'un ensemble de modèles d'apprentissage en profondeur

La pneumonie est une infection pulmonaire aiguë pouvant être causée par des bactéries, des virus ou des champignons, et qui affecte les poumons, provoquant une inflammation des sacs aériens et un épanchement pleural. Ce dernier

est une condition où le poumon se remplit de liquide. Elle représente plus de 15 % des décès d'enfants de moins de cinq ans [52]. La pneumonie est plus fréquente dans les pays sous-développés et en développement, où la surpopulation, la pollution et les conditions environnementales insalubres aggravent la situation et où les ressources médicales sont rares. Par conséquent, un diagnostic et une prise en charge précoces peuvent jouer un rôle central dans la prévention de la mort de la maladie. L'examen radiologique des poumons par tomodensitométrie (TDM), imagerie par résonance magnétique (IRM) ou radiographie (rayons X) est fréquemment utilisé pour le diagnostic. L'imagerie par rayons X constitue un examen non invasif et relativement peu coûteux des poumons. Figure 2.1 montre un exemple de radiographie comparant une pneumonie à un poumon sain. Sur la radiographie pulmonaire, les taches blanches (indiquées par des flèches rouges) et appelées infiltrats, permettent de distinguer un état pulmonaire pathologique d'un état sain. Cependant, l'interprétation des examens radiographiques du thorax pour la détection de la pneumonie est sujette à une variabilité subjective [53], [54]. Il est donc nécessaire de disposer d'un système automatisé pour la détection de la pneumonie. Dans le cadre de cette étude, nous avons mis au point un système de diagnostic assisté par ordinateur s'appuyant sur divers modèles d'apprentissage profond pour assurer une classification précise des images radiographiques thoraciques. L'apprentissage profond, un outil essentiel de l'intelligence artificielle, joue un rôle déterminant dans la résolution de problématiques complexes en vision par ordinateur [55] [56]. Les modèles d'apprentissage en profondeur, en particulier les réseaux de neurones convolutionnels (CNN), sont largement utilisés pour divers problèmes de classification d'images. Cependant, pour fonctionner de manière optimale, ces modèles nécessitent une grande quantité de données. Dans le domaine de la classification d'images biomédicales, l'acquisition d'un volume conséquent de données étiquetées s'avère ardue, nécessitant l'intervention de médecins experts pour classer chaque image, une tâche à la fois coûteuse et chronophage.

### 2.4.1 Etat de l'art sur la classification des images radiologiques du poumon

La détection de la pneumonie à l'aide de radiographies pulmonaires est un problème ouvert depuis de nombreuses années [57, 58], la principale limitation étant la rareté des données accessibles au public. Les méthodes traditionnelles d'apprentissage automatique ont été largement explorées. Chandra et al. [59] ont segmenté les régions pulmonaires à partir d'images radiographiques thoraciques et extrait huit caractéristiques statistiques de ces régions, qu'ils ont

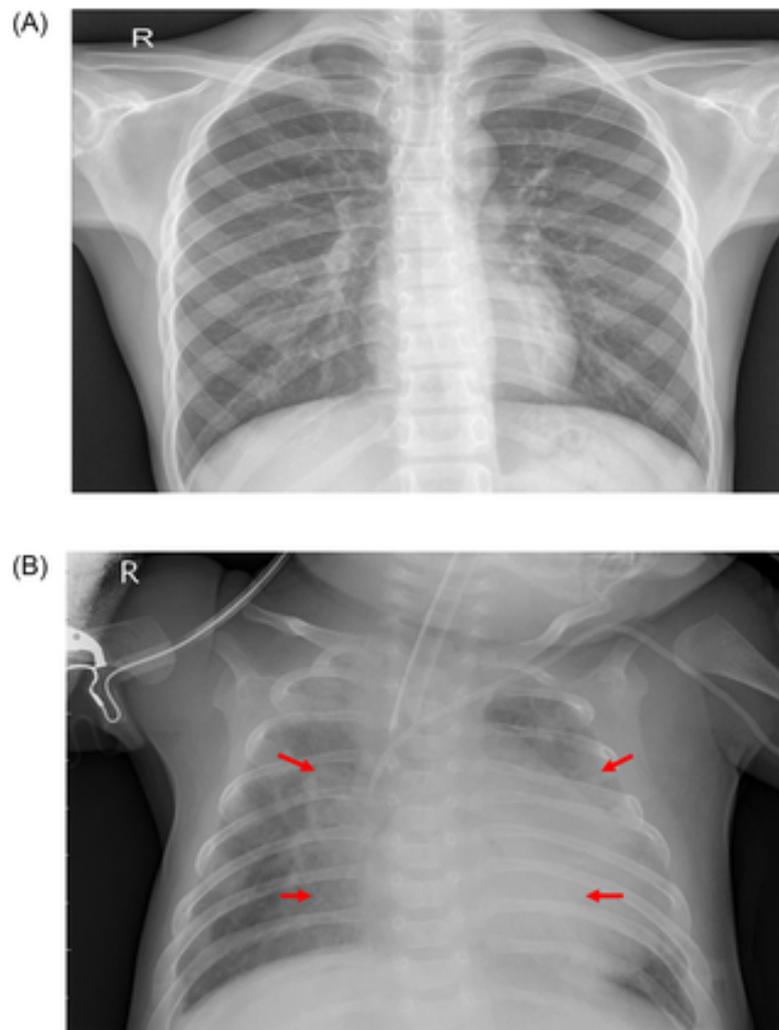


FIGURE 2.1 – Exemples de deux plaques radiographiques qui affichent (a) un poumon sain et (b) un poumon pneumonique. Les flèches rouges en (b) indiquent des infiltrats blancs, une caractéristique distinctive de la pneumonie. Les images ont été extraites du jeu de données Kermay [18].

utilisées pour les classer. Ils ont mis en œuvre cinq classificateurs traditionnels : le perceptron multicouche (MLP), la forêt aléatoire, l'optimisation minimale séquentielle (SMO), la classification par régression et la régression logistique. Ils ont évalué leur méthode sur 412 images et ont obtenu un taux d'accuracy de 95,39% en utilisant le classificateur MLP. Kuo et al. [60] ont utilisé 11 caractéristiques pour détecter la pneumonie chez 185 patients schizophrènes. Ils ont appliqué ces caractéristiques dans un grand nombre de modèles de ré-

gression et de classification, tels que les arbres de décision, les machines à vecteurs de support et la régression logistique, et ont comparé les résultats des modèles. Ils ont atteint le taux d'accuracy le plus élevé, 94,5%, en utilisant un classificateur d'arbre de décision; les autres modèles ont échoué par de grandes marges. De même, Yue et al. [61] ont utilisé 6 caractéristiques pour détecter la pneumonie dans les images de tomographie thoracique de 52 patients; la meilleure valeur AUC qu'ils ont obtenue était de 97%. Cependant, ces méthodes ne peuvent pas être généralisées et ont été évaluées sur de petits ensembles de données.

Contrairement aux algorithmes d'apprentissage automatique, pour lesquels les caractéristiques doivent être extraites et sélectionnées manuellement pour la classification ou la segmentation [62], les méthodes basées sur l'apprentissage en profondeur effectuent une classification de bout en bout [63–65], où les caractéristiques pertinentes et informatives sont automatiquement extraites des données d'entrée pour la classification. Les CNN sont préférés pour la classification des données d'image car ils extraient automatiquement les caractéristiques invariantes en translation grâce à la convolution de l'image d'entrée avec des filtres. Les CNNs fonctionnent mieux que l'apprentissage automatique ou les méthodes traditionnelles de traitement d'images dans les tâches de classification d'images et sont donc largement utilisés par les chercheurs.

Sharma et al. [66] et Stephen et al. [67] ont conçu des architectures CNN pour la classification des images radiographiques pulmonaires. Ils ont utilisé l'augmentation des données pour compenser la rareté des données. Sharma et al. et Stephen et al. ont atteint un taux de précision de 90,68% et 93,73% respectivement sur l'ensemble de données fourni par Kermay et al. [18]. Cela indique que leurs modèles ont bien fonctionné pour prédire avec précision les résultats sur l'ensemble de données donné. Rajpurkar et al. [68] ont utilisé le modèle CNN DenseNet-121 pour la classification de la pneumonie, mais n'ont obtenu qu'un score F1 de 76,8% pour la classification. Ils soupçonnaient que l'absence d'informations sur les antécédents du patient était une cause majeure de la performance inférieure à la fois de leur modèle d'apprentissage en profondeur et des radiologues avec lesquels ils comparaient les performances de leur méthode.

Dans l'étude de Kermay et al. [18], ils ont proposé une méthode utilisant les CNN basée sur l'apprentissage par transfert, une approche efficace dans les domaines où les données sont limitées. L'architecture principale a été proposée initialement pour le diagnostic du glaucome à partir d'images OCT de la rétine.

La même architecture a également été testée sur des images de radiographies thoraciques en utilisant l'apprentissage par transfert. Pour l'OCT rétinien, dans une comparaison multiclasse entre œdème maculaire diabétique, néovascularisation choroïdienne, drusen et normal, ils ont obtenu une accuracy de 96,6%, une sensibilité de 97,8% et une spécificité de 97,4%. Pour les radiographies pulmonaires, en comparant des cas anormaux et normaux, ils ont obtenu une accuracy de 92,8%, une sensibilité de 93,2% et une spécificité de 90,1%.

Les travaux présentés ci-dessus montrent l'applicabilité de certaines architectures CNN qui permettent la sélection et la détection de caractéristiques pour améliorer au maximum les performances d'un algorithme d'apprentissage profond. Dans cette étude, nous proposons une architecture CNN appliquée aux images radiographiques thoraciques contenant des manifestations de pneumonie.

## 2.4.2 Approche proposée

Dans ce travail de thèse, nous proposons une approche efficace pour la classification des images radiographiques pulmonaires. L'objectif principal est de composer un réseau neuronal profond sur notre ensemble de données, afin qu'il puisse distinguer entre les cas positifs et les cas négatifs. L'approche proposée comporte trois étapes fondamentales : la préparation des données, un prétraitement des données, la formation du modèle, et la phase de test. Les deux premières étapes contiennent des méthodes et des techniques appliquées à la base de données afin qu'elle puisse être utilisée par le processus d'apprentissage. Elles sont considérées comme des étapes très importantes qui affectent directement le résultat final. Dans la troisième étape, nous proposons une architecture basée sur les CNNs, où le modèle s'entraîne en ajustant les poids à l'aide d'un algorithme appelé la rétropropagation du gradient d'erreur. Lorsque le modèle a fini de mettre à jour ses poids, on évalue le modèle en présentant la base de données de validation et en calculant le taux de bonne classification, c'est ce qu'on appelle l'accuracy du modèle.

## 2.4.3 Environnement de développement

Lorsque nous voulons travailler avec des systèmes d'apprentissage en profondeur, nous devons d'abord commencer par préparer et comprendre les outils et bibliothèques appropriés. Avec différents systèmes d'exploitation et types de langages de programmation et de bibliothèques d'apprentissage en profondeur,

il n'est pas toujours facile de trouver une combinaison qui fonctionne sans erreur. La sortie de nouvelles bibliothèques et leurs mises à jour apportent de nouvelles fonctionnalités et améliorations. Cependant, sous toute forme qui montre la compatibilité devient plus compliquée et peut causer des problèmes d'incompatibilité pour ceux qui utilisent des versions plus anciennes. Ainsi, la conception de l'environnement de développement est souvent basée sur les écoles et les versions sélectionnées qui sont compatibles entre elles. Dans ce qui suit, nous présentons le langage de programmation utilisé, et les principales bibliothèques pour réaliser nos expérimentations.

## Python

Python est l'un des langages de programmation les plus largement utilisés par les développeurs à l'heure actuelle. Il fait partie des langages les plus populaires aux côtés de C++, Java et autres.

Python est un langage de programmation léger, adaptable et simple qui peut alimenter des scripts sophistiqués et des applications en ligne lorsqu'il est utilisé dans un cadre solide. En tant que langage de programmation à usage général, il a été développé en 1991. Les développeurs louent depuis longtemps sa simplicité et sa facilité d'apprentissage, et sa popularité ne connaît pas de limites. Il est polyvalent et prend en charge une variété de Frameworks et de bibliothèques. Python est l'un des langages de programmation les plus demandés pour l'apprentissage automatique, l'analyse de données et le développement Web, et parce qu'il est rapide à coder et simple à apprendre, les ingénieurs Python sont en demande. Tout le monde aime Python car il offre une grande flexibilité de codage. Il dispose d'un certain nombre de packages de visualisation et de bibliothèques de base importantes telles que sklearn, seaborn, etc., en raison de son évolutivité et de sa nature open source. Ces bibliothèques puissantes facilitent le codage et donnent aux ordinateurs la possibilité d'en savoir plus.

## Les bibliothèques Python

Un grand choix de bibliothèques est l'une des principales raisons pour lesquelles Python est le langage de programmation le plus populaire utilisé pour l'AI. Une bibliothèque est un module ou un groupe de modules publiés par différentes sources, comme PyPi, qui comprend un morceau de code pré-écrit permettant aux utilisateurs d'atteindre certaines fonctionnalités ou d'effectuer différentes actions. Les bibliothèques Python fournissent des éléments de base afin que les développeurs n'aient pas à les coder depuis le début à chaque fois.

Le ML nécessite un traitement continu des données, et les bibliothèques Python vous permettent d'accéder aux données, de les manipuler et de les trans-

former. Voici quelques-unes des bibliothèques les plus répandues en Python pour l'IA :

- 1) **NumPy** : NumPy est largement considéré comme la meilleure bibliothèque Python pour l'apprentissage automatique et l'IA. Il s'agit d'une bibliothèque numérique open source qui peut être utilisée pour effectuer diverses opérations mathématiques sur différentes matrices. NumPy est considérée comme l'une des bibliothèques scientifiques les plus utilisées, c'est pourquoi de nombreux data scientists s'y fient pour analyser les données. Les tableaux NumPy nécessitent beaucoup moins d'espace de stockage que les autres listes Python, et ils sont plus rapides et plus pratiques à utiliser. NumPy permet de manipuler les données matricielles, les transposer et les remodeler. Dans l'ensemble, NumPy est une excellente option pour augmenter les performances des modèles d'apprentissage automatique sans trop de travail complexe.

Voici quelques-unes des principales fonctionnalités de NumPy :

- Manipulation de forme.
- Nettoyage/manipulation des données.
- Opérations statistiques et algèbre linéaire.

- 2) **SciPy** : SciPy est une bibliothèque open source gratuite basée sur NumPy. Il est particulièrement utile pour les grands ensembles de données, étant capable d'effectuer des calculs scientifiques et techniques. SciPy est également livré avec des modules intégrés pour l'optimisation des tableaux et l'algèbre linéaire, tout comme NumPy.

Le langage de programmation inclut toutes les fonctions de NumPy, mais il les transforme en outils scientifiques conviviaux. Il est souvent utilisé pour la manipulation d'images et fournit des fonctionnalités de traitement de base pour les fonctions mathématiques de haut niveau non scientifiques.

SciPy est l'une des bibliothèques Python fondamentales grâce à son rôle dans l'analyse scientifique et l'ingénierie.

Voici quelques-unes des principales fonctionnalités de SciPy :

- Convivial.
- Visualisation et manipulation des données.
- Analyse scientifique et technique.
- Calcul de grands ensembles de données.

- 3) **Theano** : Bibliothèque Python de calcul numérique, Theano a été développée spécifiquement pour l'apprentissage automatique. Il permet l'optimisation, la définition et l'évaluation des expressions mathématiques et des calculs matriciels. Cela permet d'utiliser des tableaux dimensionnels pour construire des modèles d'apprentissage en profondeur.

Theano est une bibliothèque très spécifique, et elle est principalement utilisée par les développeurs et les programmeurs d'apprentissage automatique et d'apprentissage en profondeur. Il prend en charge l'intégration avec NumPy et peut être utilisé avec une unité de traitement graphique (GPU) au lieu d'une unité centrale de traitement (CPU), ce qui se traduit par des calculs intensifs de données 140 fois plus rapides.

Voici quelques-unes des principales caractéristiques de Theano :

- Outils intégrés de validation et de test unitaire.
  - Évaluations rapides et stables.
  - Calculs intensifs de données.
  - Calculs mathématiques performants.
- 4) **Pandas** : Une autre bibliothèque Python de premier plan sur le marché est Pandas, qui est souvent utilisée pour l'apprentissage automatique. Il agit comme une bibliothèque d'analyse de données qui analyse et manipule les données, et il permet aux développeurs de travailler facilement avec des données multidimensionnelles structurées et des concepts de séries chronologiques.

La bibliothèque Pandas offre un moyen rapide et efficace de gérer et d'explorer les données en fournissant des séries et des DataFrames, qui représentent efficacement les données tout en les manipulant de différentes manières.

Voici quelques-unes des principales fonctionnalités de Pandas :

- Indexation des données.
  - Alignement des données
  - Fusion/regroupement de jeux de données.
  - Manipulation et analyse des données.
- 5) **TensorFlow** : Autre bibliothèque Python gratuite et open source, TensorFlow est spécialisée dans la programmation différentiable. La bibliothèque se compose d'une collection d'outils et de ressources qui permettent aux

débutants et aux professionnels de construire des modèles DL et ML, ainsi que des réseaux de neurones.

TensorFlow se compose d'une architecture et d'un cadre flexibles, lui permettant de s'exécuter sur diverses plates-formes de calcul telles que CPU et GPU. Cela dit, il fonctionne mieux lorsqu'il est utilisé sur une unité de traitement de tenseur (TPU). La bibliothèque Python est souvent utilisée pour implémenter l'apprentissage par renforcement dans les modèles ML et DL, qui permet la visualisation directe des modèles d'apprentissage automatique. Voici quelques-unes des principales fonctionnalités de TensorFlow :

- Architecture et Framework flexibles.
- Fonctionne sur une variété de plates-formes de calcul.
- Capacités d'abstraction
- Gère les réseaux de neurones profonds.

- 6) **keras** : Keras est une bibliothèque Python open source destinée au développement et à l'évaluation de réseaux de neurones au sein de modèles d'apprentissage automatique et d'apprentissage en profondeur. Il est capable de fonctionner sur Theano et Tensorflow, ce qui signifie qu'il peut former des réseaux de neurones avec peu de code.

La bibliothèque Keras est souvent préférée car elle est modulaire, extensible et flexible. Cela en fait une option conviviale pour les débutants. Il peut également s'intégrer à des objectifs, des couches, des optimiseurs et des fonctions d'activation. Keras fonctionne dans divers environnements et peut fonctionner sur des CPU et des GPU. Il offre également l'une des plages les plus larges pour les types de données.

Voici quelques-unes des principales fonctionnalités de Keras :

- Mutualisation des données.
- Développer des couches neuronales.
- Crée des modèles d'apprentissage en profondeur et d'apprentissage automatique.
- Fonctions d'activation et de coût.

- 7) **Scikit-Learn** :

À l'origine une extension tierce de la bibliothèque SciPy, Scikit-learn est désormais une bibliothèque Python autonome sur Github. Il est utilisé par

de grandes entreprises comme Spotify, et son utilisation présente de nombreux avantages. D'une part, il est très utile pour les algorithmes d'apprentissage automatique classiques, tels que ceux pour la détection de spam, la reconnaissance d'images, la prédiction et la segmentation des clients.

Un autre des principaux arguments de vente de Scikit-learn est qu'il est facilement interopérable avec d'autres outils de pile SciPy. Scikit-learn a une interaction conviviale et cohérente qui permet de partager et d'utiliser facilement des données.

Voici quelques-unes des principales fonctionnalités de Scikit-learn :

- Classification et modélisation des données.
- Algorithmes d'apprentissage automatique de bout en bout.
- Pré-traitement des données.
- Sélection du modèle.

#### 8) **Matplotlib :**

Matplotlib est une unité de NumPy et SciPy, et il a été conçu pour remplacer la nécessité d'utiliser le langage statistique propriétaire MATLAB. La bibliothèque complète, gratuite et open source est utilisée pour créer des visualisations statiques, animées et interactives en Python.

La bibliothèque Python aide à comprendre les données avant de les déplacer vers le traitement des données et la formation pour les tâches d'apprentissage automatique. Il s'appuie sur les boîtes à outils Python GUI pour produire des tracés et des graphiques avec des API orientées objet. Il fournit également une interface similaire à MATLAB afin qu'un utilisateur puisse effectuer des tâches similaires à MATLAB.

Voici quelques-unes des principales fonctionnalités de Matplotlib :

- Créez des tracés de qualité de publication.
- Personnalisez le style visuel et la mise en page.
- Exportez vers différents formats de fichiers.
- Figures interactives qui peuvent zoomer, faire un panoramique et mettre à jour.

### 2.4.4 Description de la base de données

Le jeu de données de Kermany et al. [18], utilisé dans ce travail, contient 5856 images de radiographie thoracique au format JPEG, parmi lesquelles 5232

images ont été sélectionnées pour la phase d'entraînement. Parmi ces images, 3883 cas représentent des patients atteints de la pneumonie (2538 bactériennes et 1345 virales), tandis que 1349 sont des cas sains. La première remarque que l'on a pu constater est que l'ensemble de données utilisé pour cette étude est déséquilibré (voir Figure 2.2). Le modèle a ensuite été testé avec 234 images nor-

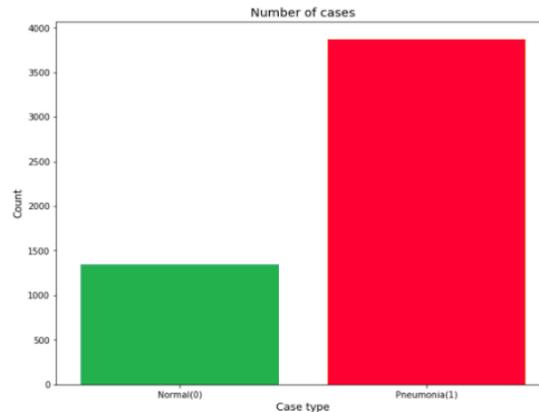


FIGURE 2.2 – Le datagramme de l'ensemble de données utilisé pour la phase d'apprentissage qui montre le déséquilibre des données. [19].

males et 390 images de pneumonie (242 bactériennes et 148 virales) provenant de 624 patients. Ces images radiographiques ont été choisies parmi des enfants âgés de 1 à 5 ans du Centre médical de Guangzhou. Toutes ces images ont été réalisées dans le cadre de soins cliniques et ont subi un prétraitement pour éliminer tous les scans de mauvaise qualité. La Figure 2.3 présente un échantillon de la base de données utilisée.

## 2.4.5 Méthodes

### Prétraitement et augmentation des données :

Pour résoudre le problème de sur-apprentissage et augmenter la capacité du modèle à généraliser pendant l'entraînement et pour augmenter la taille et la qualité de l'ensemble de données, nous avons utilisé des méthodes d'augmentation des données qui existent dans la littérature [69]. Les paramètres utilisés pour l'augmentation des données sont présentés ci-dessous dans le tableau 2.1. Le traitement commence par une remise à l'échelle de l'image (réduction ou agrandissement pendant le processus d'augmentation). Ensuite, nous effectuons une rotation des images qui seront tournées de manière aléatoire au cours



FIGURE 2.3 – La radiographie pulmonaire normale (panneau de gauche) montre des poumons clairs sans aucune zone d’opacification anormale sur l’image. La pneumonie bactérienne (au milieu) présente généralement une consolidation lobaire focale, dans ce cas dans le lobe supérieur droit (flèches blanches), tandis que la pneumonie virale (à droite) se manifeste par un schéma interstitiel plus diffus dans les deux poumons. [18].

de l’entraînement. Le décalage en largeur est le décalage des images horizontalement et le décalage en hauteur est le décalage des images verticalement ; dans notre cas, il est de 10 % pour les deux cas. La plage de zoom est de 20 % de manière aléatoire, et enfin, les images sont inversées horizontalement.

méthodes	paramètre
Plage de rotation	30
Décalage de largeur	0.1
Décalage de hauteur	0.1
Rééchelle	1/255
Plage de cisaillement	0.2
Plage de zoom	0.2
Retournement horizontal	true

TABLE 2.1 – Paramètres pour l’augmentation de l’image.

### Le modèle proposé

Dans cette étude, les images pulmonaires sont classées en deux catégories distinctes : normales et anormales, avec la détection de diverses manifestations de la pneumonie. Pour atteindre cet objectif, nous avons mis en œuvre une architecture de réseau de neurones convolutifs (CNN), comme représenté dans la figure 2.4. Cette architecture se compose de deux blocs principaux. Le premier bloc est dédié à l’extraction des caractéristiques, avec trois couches d’entrée de

32 neurones, suivies de trois couches de sortie de 64 neurones. Des couches d'activation RELU sont utilisées pour éliminer les valeurs négatives et les remplacer par des zéros. Les opérations de MaxPooling  $2 \times 2$  et de Batch Normalization sont également appliquées entre ces couches.

Ensuite, nous avons trois couches entièrement connectées pour la phase de classification, avec respectivement 512, 256 et 128 neurones, et une couche de sortie à deux neurones utilisant la fonction Sigmoid pour obtenir une probabilité. Chaque neurone d'une couche est connecté à tous les neurones du bloc suivant. Pour éviter le sur-apprentissage, nous avons ajouté une couche Dropout définie à 50%.

Nous avons utilisé un ensemble de données comprenant 5856 images de radiographies pulmonaires, avec des classes normales et anormales. Sur ces images, 5216 ont été utilisées pour la phase d'entraînement (sur 70 époques) et le reste pour la phase de test.

Les opérations de convolution, de max-pooling et de Batch Normalization produisent des plans 2D appelés "feature maps". Nous avons obtenu les tailles de feature maps de  $248 \times 248 \times 32$ ,  $122 \times 122 \times 32$ ,  $59 \times 59 \times 32$ ,  $27 \times 27 \times 64$ ,  $11 \times 11 \times 64$  et  $3 \times 3 \times 64$  pour les opérations de convolution, et de  $124 \times 124 \times 32$ ,  $61 \times 61 \times 32$ ,  $29 \times 29 \times 32$ ,  $13 \times 13 \times 64$ ,  $5 \times 5 \times 64$  et  $1 \times 1 \times 64$  pour les opérations de pooling (voir tableau 2.2).

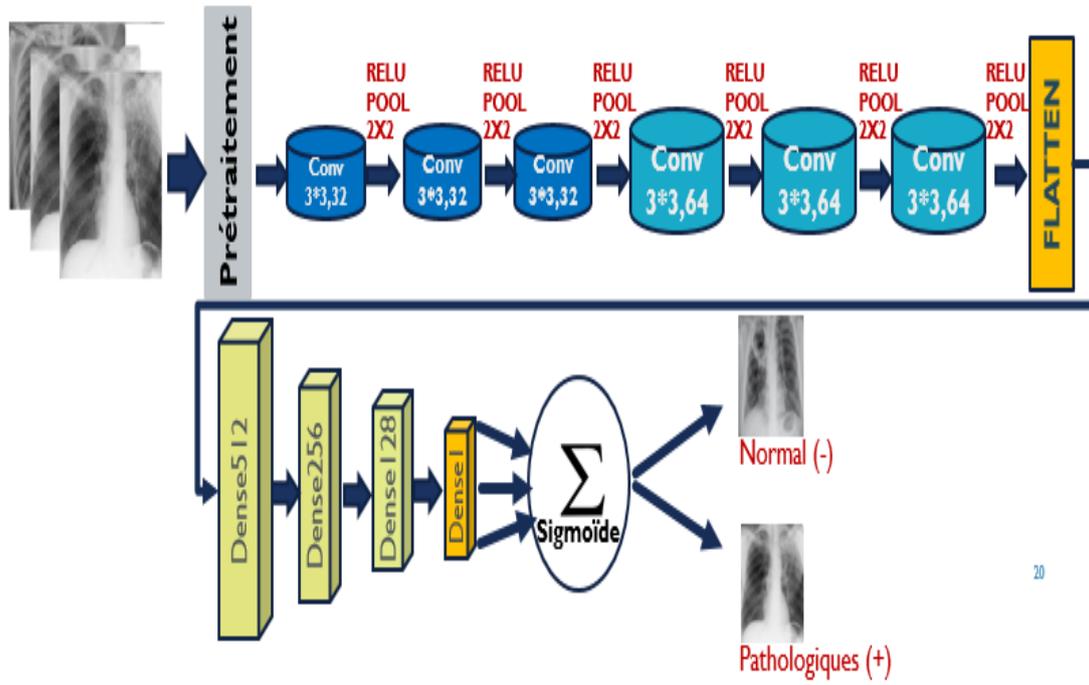


FIGURE 2.4 – L'architecture proposée.

Couche (type))	Forme de sortie	Quantité d'hyperparamètres
conv2d_1	<i>None, None, 248, 248, 32</i>	896
max_pooling2d_1	<i>None, 124, 124, 32</i>	0
batch_normalization_1	<i>None, 124, 124, 32</i>	128
conv2d_2	<i>None, 122, 122, 32</i>	9248
m_pooling2d_2	<i>None, 61, 61, 32)</i>	0
batch_normalization_2	<i>None, 61, 61, 32</i>	128
conv2d_3 (Conv2D)	<i>None, 59, 59, 32</i>	9248
max_pooling2d_3	<i>None, 29, 29, 32</i>	0
batch_normalization_3	<i>None, 29, 29, 32</i>	128
conv2d_4 (Conv2D)	<i>None, 27, 27, 64</i>	18496
max_pooling2d_4	<i>None, 13, 13, 64</i>	0
batch_normalization_4	<i>None, 13, 13, 64</i>	256
conv2d_5 (Conv2D)	<i>None, 11, 11, 64</i>	36928
max_pooling2d_5	<i>None, 5, 5, 64</i>	0
batch_normalization_5	<i>None, 5, 5, 64</i>	256
conv2d_6 (Conv2D)	<i>Lowest None, 3, 3, 64</i>	36928
max_pooling2d_6	<i>None, 1, 1, 64</i>	0
batch_normalization_6	<i>None, 1, 1, 64</i>	256
flatten_1	<i>None, 64</i>	0
dense_1	<i>None, 512</i>	33280
Dropout_1 (Dropout)	<i>None, 512</i>	0
dense_2	<i>None, 256</i>	131328
Dropout_2 (Dropout)	<i>None, 256</i>	0
dense_3 (Dense)	<i>None, 128</i>	32896
Dropout_2 (Dropout)	<i>None, 128</i>	0
dense_4 (Dense)	<i>None, 1</i>	129

TABLE 2.2 – Le résultat de l'architecture de réseau proposée.

## 2.4.6 Résultat et discussion

Comme mentionné ci-dessus, plusieurs méthodes telles que l'augmentation des données, la variation du taux d'apprentissage et le recuit ont été utilisées pour résoudre le problème de l'adaptation d'un petit ensemble de données dans l'architecture de réseau de neurones convolutifs profonds [67]. Les tailles d'images à traiter sont fixes, donc pour évaluer les performances de validation de notre modèle, nous avons redimensionné les images radiographiques à 200x200x3, 250x250x3 et 300x300x3, respectivement, et nous les avons entraînées pendant environ quatre heures chacune. Les résultats obtenus ont montré que la taille d'image 250x250x3 offrait un meilleur résultat que les autres avec

une accuracy de validation de 93,24% et une perte d'entraînement de 11,81% contre une accuracy de validation de 90,71% et une perte d'entraînement de 12,74% pour des images de taille 200x200x3, ainsi qu'une accuracy de validation de 89,9% et une perte d'entraînement de 28,68% pour des images de taille 300x300x3, comme indiqué dans le Tableau 2.3 et la Figure 2.5 .

Les di- mensions	L'accuracy d'entraînement	L'accuracy de validation
200	95.72%	90.71%
<b>250</b>	<b>95.81%</b>	<b>93.24</b>
300	94.6%	89.9%

TABLE 2.3 – Performance du modèle de classification sur différentes tailles de données

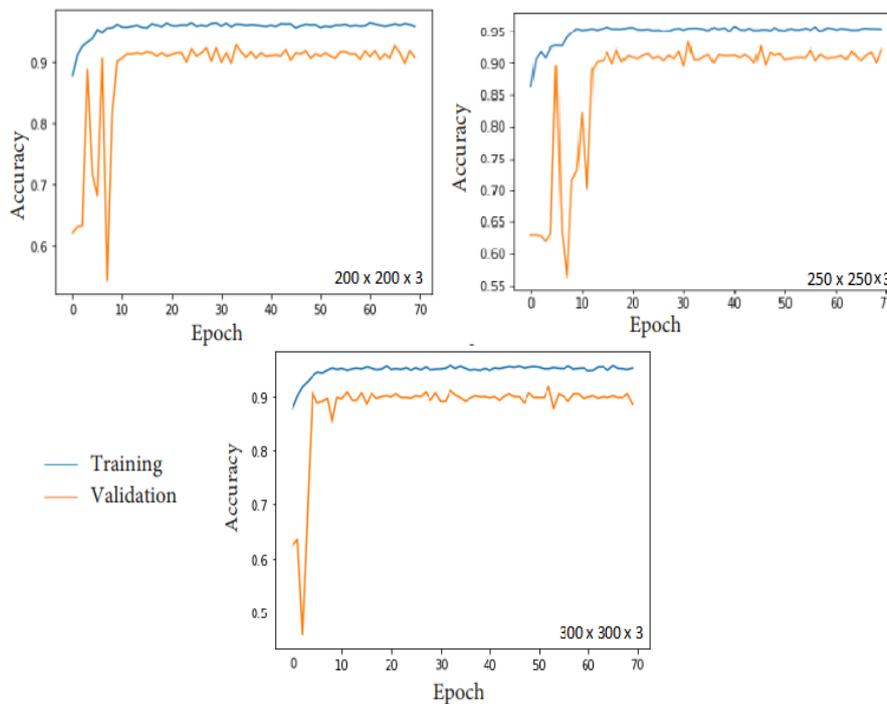


FIGURE 2.5 – Performance du modèle de classification sur des dimensions d'images différentes.

Ensuite, pour analyser plus en détail les performances de notre modèle sur des images de taille 250x250x3, quatre métriques importantes ont été considérées, qui sont l'accuracy 3.3, la sensibilité 3.5, la spécificité 3.6 et la F1-score 3.7 .

Notre modèle présente de nombreux avantages significatifs qui jouent un rôle crucial dans ses performances exceptionnelles en matière de classification des images de pneumonie. Tout d'abord, il affiche une accuracy remarquable de 93,24 % , témoignant de sa capacité à classer de manière extrêmement précis les images de pneumonie. Cette précision est essentielle pour garantir que les résultats du modèle sont fiables et cohérents.

De plus, notre modèle affiche à la fois une sensibilité et une précision élevées. Une sensibilité élevée signifie que le modèle est capable de détecter la grande majorité des cas positifs de pneumonie, réduisant ainsi le nombre de faux négatifs, ce qui est particulièrement important pour éviter de manquer des cas de pneumonie. D'autre part, la précision indique que le modèle minimise les faux positifs, c'est-à-dire qu'il ne classe pas à tort des images saines comme étant des cas de pneumonie. Cela confirme la fiabilité du modèle dans la distinction entre les cas positifs et négatifs de pneumonie.

Ces performances remarquables ont été obtenues en utilisant seulement 0,31 million de paramètres comme le montre le tableau 2.4, ce qui réduit considérablement le temps d'apprentissage par rapport à toutes les approches existantes dans la littérature. Cela démontre l'efficacité et l'efficacité de notre modèle dans la détection de la pneumonie sur des images médicales, ce qui le positionne comme une solution prometteuse pour l'amélioration du diagnostic précoce et précis de cette maladie.

Ce travail représente une contribution significative dans le domaine, car nous avons développé un modèle pour la classification des images radiographiques thoraciques avec une accuracy de validation prometteuse. Nous avons prouvé que pour obtenir d'excellentes performances sur cet ensemble de données, il n'est pas nécessaire de recourir à des architectures CNN complexes telles que VGG16, ResNet ou xception. De telles architectures peuvent conduire à un sur-apprentissage et consommer un temps d'entraînement important.

Authors	ACC	SN	SP	F1-score	Quantité d'hyperparamètres	
Kermany et al ,2018 [18]	92.8%	93.20%	90.1%	-	23 millions	
Gaobo Liang et al, 2019 [70]	90.5%	<b>96.7%</b>	-	92.7%	-	
Enes AYAN et al, 2019 [71]	Xception	82%	85%	76%	-	22 millions
	VGG16	87%	82%	91%	-	144 millions
Harsh Sharma et al, 2020 [66]	90.68%	-	-	-	3,47 millions	
<b>Le modèle proposé</b>	<b>93.24%</b>	95%	<b>91.4%</b>	<b>92.96%</b>	<b>0.31 million</b>	

TABLE 2.4 – Comparaison des résultats

Dans ce travail de thèse, nous avons présenté un modèle destiné à la classification d'images radiographiques thoraciques en vue frontale avec accuracy de validation élevée. Au début, l'algorithme minimisait la taille de ces images thoraciques, puis nous avons utilisé la méthode d'augmentation des données pour réduire le risque de surapprentissage. L'étape suivante consiste à appliquer ces données à un réseau de neurones convolutifs pour extraire les caractéristiques des images pour l'identification et la classification. En raison de la robustesse de CNN, l'accuracy de notre modèle était encourageante et prometteuse par rapport à d'autres approches, comme le montre le tableau 2.4 . À partir de ces résultats, nous avons démontré que pour obtenir de bonnes performances dans cet ensemble de données, nous n'avons pas besoin d'utiliser une architecture complexe de CNN telle que VGG16, ResNet ou AlexNet, car, compte tenu de la taille limitée de cette base de données, un apprentissage plus profond pourrait conduire à un sur-apprentissage, sans oublier le temps considérable requis pour la phase d'entraînement. La validation des performances du modèle entraîné sur différentes tailles d'images radiographiques est basée sur la variation des tailles de l'ensemble de données d'entraînement et de validation et nous avons obtenu des résultats relativement similaires. Cela pourrait grandement contribuer à améliorer la santé infantile par la prévention et la détection précoce de la pneumonie.

## 2.5 Conclusion

Le recours à la radiographie pour le diagnostic des maladies pulmonaires est une pratique établie et fréquemment utilisée dans les centres hospitaliers. Récemment, l'intégration de techniques d'apprentissage en profondeur, et plus précisément de réseaux de neurones convolutifs (CNN), s'est avérée être une méthode prometteuse pour accélérer et améliorer la précision du diagnostic de la pneumonie à travers l'analyse et la classification d'images médicales.

Dans ce contexte, le modèle que nous avons proposé se distingue par son approche innovante, prenant en considération à la fois les paramètres essentiels et le temps d'apprentissage. Cette démarche le différencie avantageusement des autres méthodologies qui s'appuient majoritairement sur le transfert d'apprentissage ou sur l'utilisation d'architectures plus complexes.

Un des défis majeurs de cette démarche réside dans la sélection d'une architecture de classification pertinente et efficace. C'est pourquoi, dans le cadre de ce travail, nous avons mis en œuvre une stratégie rigoureuse pour générer, tester, et évaluer divers modèles de CNN adaptés à différentes dimensions d'images.

Les résultats obtenus témoignent de la réussite de notre approche, avec des performances qui surpassent celles des modèles existants en termes d'accuracy, de sensibilité, et de spécificité. De plus, notre modèle se distingue par sa capacité à délivrer ces performances exceptionnelles tout en nécessitant une puissance de calcul moindre. Cette caractéristique le rend particulièrement adapté aux contextes où les ressources sont limitées, assurant ainsi une plus grande accessibilité et une utilisation optimale même dans des conditions suboptimales.

# Chapitre 3

## Modèle explicable pour l'aide au diagnostic de la COVID-19

### 3.1 Introduction

Le COVID-19, aussi connu sous le nom de syndrome respiratoire aigu, est un virus hautement contagieux et représente l'un des défis majeurs auxquels la société moderne est confrontée. Identifiée pour la première fois à Wuhan, en Chine, en décembre 2019, cette pandémie s'est rapidement propagée à l'échelle mondiale [72, 73]. De nombreux chercheurs et laboratoires du monde entier se sont mobilisés pour trouver un vaccin. Toutefois, en dépit de la mise au point de certains vaccins, le virus continue de se propager rapidement, en particulier avec l'émergence de nouvelles souches.

La méthode de dépistage la plus couramment utilisée est la réaction en chaîne par polymérase par transcription inverse (RT-PCR). Cependant, cette technique présente une sensibilité limitée et une pénurie mondiale de kits de test est survenue [74]. Par conséquent, beaucoup de malades ne sont pas diagnostiqués et confinés à temps, amplifiant ainsi la propagation du virus. Face à cette situation, les professionnels de la santé ont dû recourir à d'autres techniques de diagnostic pour endiguer la propagation [75, 76]. Parmi celles-ci, l'imagerie médicale, notamment la tomodensitométrie (CT) et la radiographie (rayons X), est recommandée comme complément à la RT-PCR [77]. Les scans thoraciques, rapides et facilement accessibles, ont montré une sensibilité bien supérieure à celle de la RT-PCR dans le diagnostic de la COVID-19 [78]. Néanmoins, étant donné le volume élevé de cas (dépassant les 160 millions à ce jour), il y a un manque criant de radiologues pour analyser ces images. Ainsi, les techniques d'apprentissage profond (Deep Learning, DL) se présentent comme une solution prometteuse pour la détection automatique de la COVID-19, grâce à

leur capacité à traiter d'importants volumes de données image avec précision.

Dans le chapitre précédent, nous avons exploré la classification des pneumonies à partir d'images radiographiques thoraciques (Chest X-Ray).

Suite à nos échanges avec les médecins du CHU, il ressort que la plupart s'appuient sur les images de tomodensitométrie pour diagnostiquer la COVID-19, ces dernières étant plus riches en informations que les radiographies simples. Ainsi, nous avons proposé une architecture explicative basée sur l'apprentissage profond pour classifier les images de tomodensitométrie thoracique.

Si certaines bases de données sont disponibles publiquement, elles ne fournissent souvent pas d'informations détaillées sur la manière dont les patients ont été diagnostiqués (par exemple, par RT-PCR). Pour pallier cette lacune, nous avons constitué un nouvel ensemble de données de scanners thoraciques pour la COVID-19 à partir de patients de l'hôpital de Tlemcen en Algérie, incluant des cas confirmés par RT-PCR et d'autres non.

Dans le cadre de ce travail de thèse, nous avons défini deux objectifs principaux :

- La collecte d'un nouvel ensemble de données de tomodensitométrie COVID-19 depuis des patients testés positifs ou négatifs par RT-PCR.
- Le développement d'une nouvelle architecture personnalisée basée sur le transfert d'apprentissage, utilisant le modèle Densenet201 combiné à l'algorithme GradCam.

## 3.2 Etat de l'art sur la détection de la COVID-19

Ces dernières années, l'apprentissage profond a connu un franc succès dans de nombreux domaines de l'imagerie médicale, améliorant considérablement la précision de détection et de classification des différentes sortes d'images. Plusieurs études ont recouru à des techniques d'apprentissage profond pour la détection automatique de maladies pulmonaires, telles que la pneumonie [19], [79], et la tuberculose [80], [81], [82]. Dans le contexte actuel, où le coronavirus se propage rapidement à l'échelle mondiale, de nombreuses recherches se sont concentrées sur le diagnostic de la COVID-19 en s'appuyant sur des tomodensitogrammes et des images radiographiques pulmonaires en utilisant des approches d'apprentissage profond.

Diverses études ont employé l'apprentissage profond dans l'interprétation des images radiologiques pour répondre aux défis posés par les techniques médicales COVID-19 basées sur des images. L'architecture CNN se démarque comme étant particulièrement efficace parmi les algorithmes d'apprentissage

profond les plus pertinents. Le traitement des données avec des algorithmes d'apprentissage profond, en particulier CNN, a suscité un vif intérêt.

Les algorithmes d'apprentissage profond permettent de distinguer, à partir de radiographies et de tomodensitométries, un patient sain d'un patient atteint de la COVID-19. Des modèles d'apprentissage profond sont mis à profit pour élaborer des systèmes de diagnostic de la COVID-19. Plusieurs architectures, telles que DenseNet121, VGG16, Xception, EfficientNet et NASNet, ont été utilisées pour une classification multiclasse, prenant en compte les personnes testées positives à la COVID-19, les patients non infectés, ainsi que d'autres patients présentant des maladies pulmonaires diverses [83]. Pour le diagnostic de la maladie, certains algorithmes reposent sur le DNN (Deep Neural Networks) en s'appuyant sur les caractéristiques d'imagerie des fractales, tandis que d'autres privilégient le CNN en utilisant directement l'imagerie pulmonaire. L'architecture CNN suggérée, avec une précision de 93,2 % et une sensibilité de 96,1 %, surpasse la technique DNN présentant une précision de 83,4 % et une sensibilité de 86 % [84].

Hemdan et al. ont utilisé des modèles d'apprentissage en profondeur pour déduire le statut positif ou négatif de COVID-19 et ont rapporté que le modèle VGG19 fonctionnait mieux avec une Accuracy de 90% sur 25 images infectées par COVID-19 et 25 images non COVID-19 [85]. Toğaçar et al ont incorporé 295 images COVID-19, 98 images de pneumonie et 65 images normales dans MobileNet et SqueezeNet [86]. Ils ont extrait les caractéristiques des modèles Net entraînés, puis ont utilisé l'algorithme SMO (Sequential Minimal Optimization) pour sélectionner les caractéristiques, avec une Accuracy globale de 99,27 % signalée pour le classificateur SVM. Zhang et al. ont étudié un modèle ResNet à 18 couches pour 100 images de COVID-19 et 1431 images de pneumonie et ont rapporté une Accuracy de 95,18 % [87]. Apostolopoulos et Mpesiana [88] ont utilisé une base de données composée de 1428 images radiographiques pulmonaires (224 patients positifs pour COVID-19, 700 cas de pneumonie, et le reste sont des patients sains) afin d'évaluer l'efficacité de l'avancement des architectures de réseaux neuronaux convolutifs (CNN) telles que l'apprentissage par transfert dans la classification COVID-19. Les chercheurs ont proposé un modèle VGG-19 pré-formé composé de dix-neuf couches convolutionnelles profondes, les résultats ont démontré une Accuracy de 98,75% .

Xu et al [89] ont proposé un modèle CNN afin de faire la distinction entre le COVID-19, la grippe A et la pneumonie virale; l'architecture proposée offrait une précision de 86,7 %. Wang et al [90] ont développé un modèle de prédiction basé sur des approches DL utilisant la technique modifiée d'Inception basée sur l'apprentissage par transfert. La précision obtenue par cette méthode est de 89,5% ce qui la rend meilleure par rapport aux résultats de [89]. Narine et

al [91] ont appliqué un autre modèle pré-formé plus complexe d'ImageNet avec apprentissage par transfert pour la prédiction de COVID-19 dans les images radiographiques thoraciques. Les chercheurs ont utilisé les modèles Inception-ResNetV2, InceptionV3 et ResNet50. Le modèle ResNet50 a atteint la meilleure précision avec une précision de 98%, ce qui est meilleur et plus significatif que Xu et al [89] et Wang et al [90].

Les auteurs de [92] ont évalué le DarkNet avec 17 couches convolutives en utilisant 127 COVID-19, 500 pneumonies et 500 images normales et ont rapporté une Accuracy de 98,08 %. Dans [93], les performances de CNN ont été améliorées via des algorithmes de prétraitement d'image, résultant en un modèle avec une Accuracy de 94,5%.

DenseNet, InceptionV3 et Inception-ResNetV4 ont été recommandés comme trois modèles différents. Dans l'étude, des radiographies du thorax ont été utilisées pour diagnostiquer des individus atteints de COVID-19 et de pneumonie. En utilisant la validation croisée 5 fois, ces trois modèles créent et évaluent les analyses des courbes ROC et les matrices d'incertitude. L'architecture DenseNet pré-entraînée a atteint une efficacité de classification optimale de 92 % dans les simulations, tandis que les deux autres modèles, Inception V3 et Inception-ResNetV4, ont atteint 83,47 % et 85,57 %, respectivement [94].

Les techniques d'imagerie radiologique, combinées à des méthodes avancées d'intelligence artificielle, peuvent contribuer à une détection précise des maladies, palliant ainsi le manque de médecins spécialisés dans certaines zones, notamment rurales. Cet article propose une nouvelle méthode d'identification automatisée de la COVID-19 à partir d'images radiologiques brutes du thorax [92].

Dans certaines études, des méthodes explicables ont été exploitées pour visualiser les prédictions du réseau. Ces techniques d'apprentissage profond sont essentielles pour un contrôle rapide et efficace de la propagation de la COVID-19. Plusieurs méthodes, fondées sur l'apprentissage profond, ont été suggérées pour le diagnostic de la COVID-19. Certaines se basent sur des images radiographiques du thorax à l'aide d'architectures CNN avancées, tandis que des études plus récentes ont utilisé des images CT pour identifier et classer les traits caractéristiques de la maladie.

Dans cette section, nous avons présenté une méthode d'apprentissage profond explicable, nous permettant d'extraire les caractéristiques de la COVID-19 à partir d'images CT, afin de réduire le taux de faux négatifs et de fournir une analyse clinique préliminaire ainsi qu'une justification visuelle des décisions.

### 3.3 Base de données collectée

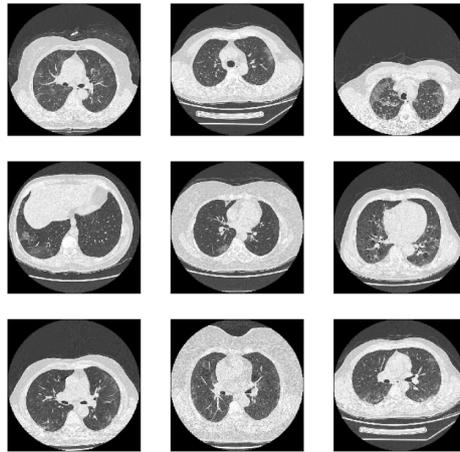
L'ensemble de données a été collecté entre juin 2020 et octobre 2020 à l'hôpital universitaire de Tlemcen, en Algérie. Il comprend 177 patients (69 infectés et 108 non infectés) et se compose de 4986 images CT. De ce total, 1868 images proviennent de patients infectés par la COVID-19 et confirmés par RT-PCR, et 3118 images de patients non infectés mais souffrant d'autres maladies pulmonaires. Toutes ces images ont été prétraitées pour éliminer tout contenu non pertinent. En effet, les images CT comprennent parfois des tranches noires ou des images sans sections pulmonaires. Après un nettoyage minutieux, nous avons obtenu un ensemble de données plus cohérent. La figure 3.1 illustre quelques exemples issus de notre collection.

### 3.4 Modèle proposé

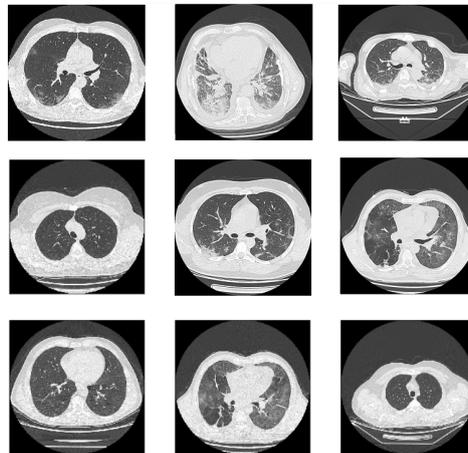
Cette étude vise à déterminer la meilleure architecture pour classer les patients suspectés de COVID-19 (positifs ou négatifs). À cet effet, nous avons sélectionné plusieurs architectures CNN ayant montré d'excellents résultats sur le jeu de données ImageNet. Nous avons employé la méthode d'augmentation des données pour enrichir automatiquement notre ensemble de données et prévenir le surapprentissage [95]. Cette méthode affecte de manière significative la taille et la qualité de l'ensemble de données ainsi que la capacité du modèle pendant l'entraînement. Dans ce cadre, l'outil d'augmentation a d'abord re-sizé les images (réduction ou agrandissement pendant le processus d'augmentation). Par la suite, une rotation aléatoire a été appliquée pendant l'entraînement. Nous avons également procédé à un décalage en hauteur et en largeur pour bouger les images horizontalement et verticalement. La plage de zoom a été fixée aléatoirement à 30%. Finalement, les images de l'ensemble de données ont été inversées horizontalement.

Le tableau 3.1 présente les paramètres utilisés pour l'augmentation des données. L'ensemble de données a été divisé aléatoirement, 80% étant dédiés à l'entraînement et le reste pour les tests. Nous avons ensuite expérimenté plusieurs architectures CNN, modifiant seulement les couches entièrement connectées pour adapter le modèle avec les sorties attendues, tout en conservant les couches de convolution et de pooling. Ces deux derniers types de couches sont mobilisés lors de l'extraction des caractéristiques. Dans notre étude, nous avons deux sorties, et la fonction Softmax est utilisée pour classer ces deux catégories : l'une pour les cas sains (négatifs) et l'autre pour les cas infectés (positifs).

Le schéma fonctionnel de notre approche est illustré à la figure 3.2. Pour tous les modèles proposés, les images ont été redimensionnées à 128 x 128 x 3.



(a) Les patients infectés par COVID-19 et confirmés avec la RT-PCR positif



(b) Les Patients dont les résultats de la RT-PCR sont négatifs mais qui présentent d'autres maladies pulmonaires

FIGURE 3.1 – Échantillons de l'ensemble des données collectées.

La taille du lot et le nombre d'époques ont été fixés respectivement à 64 et 200, en utilisant la descente stochastique du gradient (SGD) comme optimiseur. Pour une meilleure visualisation des résultats, nous avons fait appel à l'algorithme GradCam pour expliquer la prise de décision du modèle.

méthodes	paramètre
Plage de rotation	30
Décalage de largeur	0.2
Décalage en hauteur	0.2
Réechelle	1/255
Plage de cisaillement	0.2
Plage de zoom	0.2
Bascule horizontale	vrai
Bascule verticale	vrai

TABLE 3.1 – Les paramètres utilisés pour la méthode d’augmentation de donnée.

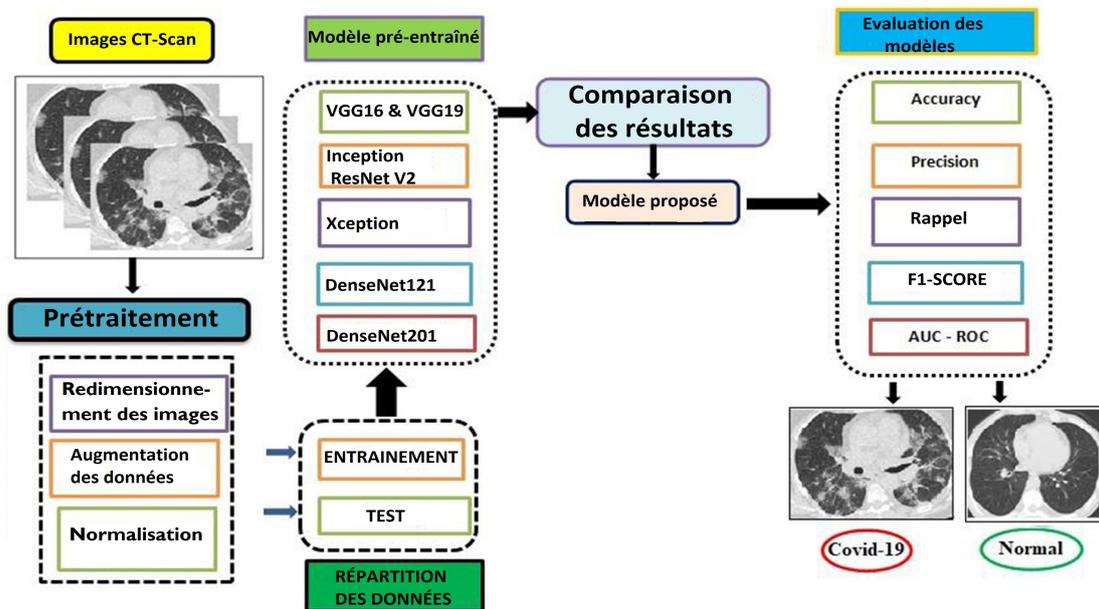


FIGURE 3.2 – Schéma fonctionnel de ce travail.

L’apprentissage par transfert a montré sa robustesse dans des problèmes limités de classification de données. Les résultats peuvent être améliorés en utilisant l’hyper-réglage de Deep Transfer Learning (DTL). Dans notre contribution, le modèle DenseNet201 par transfert d’apprentissage a été proposé pour extraire automatiquement les caractéristiques et utiliser leurs poids appris sur le jeu de données ImageNet, ce qui réduit l’effort de calcul. DenseNet a prouvé son efficacité sur différents ensembles de données tels que CIFAR-100 Deng et

al. (2009) et ImageNet. De plus, cette architecture fournit des modèles simples et faciles à construire. De plus, il est possible de réutiliser des fonctionnalités par différentes couches, ce qui rend les paramètres de cette architecture très efficaces et permet d'augmenter la variation dans les couches suivantes et d'améliorer les performances. Les caractéristiques des dernières couches profondes peuvent être projetées par celles de toutes les couches précédentes (c'est-à-dire que les couches du réseau profond peuvent réutiliser toutes les caractéristiques produites par les couches précédentes) sous la forme de :

$$X^l = H_l([X^0, X^1, \dots, X^{l-1}]). \quad (3.1)$$

Où  $H_l(\cdot)$  est une fonction composite de trois opérations, y compris la normalisation par lots (BN). Elle est suivie d'une fonction ReLu et d'une couche de convolution (3,3). Dans les architectures CNN classiques, les couches de convolution sont généralement suivies de couches de sous-échantillonnage afin de réduire de moitié la taille des cartes de caractéristiques. Par conséquent, la concaténation des cartes de caractéristiques autour des couches de sous-échantillonnage peut entraîner une multiplicité de tailles. Pour résoudre ce problème, des blocs denses ont été conçus avant les couches de sous-échantillonnage, tandis que les couches des blocs denses sont étroitement connectées, comme le montre la figure 3.3, ce qui rend la taille des cartes d'entités constante dans tous les blocs denses et réduite de moitié après le sous-échantillonnage. Ainsi, pour  $X$  couches d'un bloc dense, le nombre total de liens entre les couches est  $X(X+1)/2$ , contrairement à un réseau convolutionnel traditionnel qui est égal à  $X$ . Cependant, le calcul sera énorme si les couches sont profondes car le nombre de cartes de caractéristiques concaténées saisies dans les couches est élevé. Il est donc nécessaire de contrôler le nombre de cartes de caractéristiques nouvellement créées par un taux de croissance  $K$ . Le nombre total de cartes dans la dernière couche d'un bloc dense se présente sous la forme suivante :

$$K_0 + (X-1)K. \quad (3.2)$$

Où  $K_0$  est le nombre de canaux dans la première couche d'entrée. Pour résoudre l'effort de calcul, des couches  $1 \times 1$  ont été appliquées avant chaque couche de convolution  $3 \times 3$ , suivies par des couches de transition qui améliorent considérablement la compacité du réseau en contrôlant le nombre de cartes de caractéristiques de sortie. En outre, il se trouve généralement après le bloc dense à une certaine profondeur, ce qui explique pourquoi il n'y a pas de transition après le premier bloc dense, comme le montre 3.3. Dans ce travail, nous avons introduit un modèle sur mesure basé sur DenseNet201 pour le dépistage du COVID-19. Nous avons exploité le DenseNet201 pré-entraîné, conservant uniquement les couches d'extraction de caractéristiques. Puis, nous avons ajouté

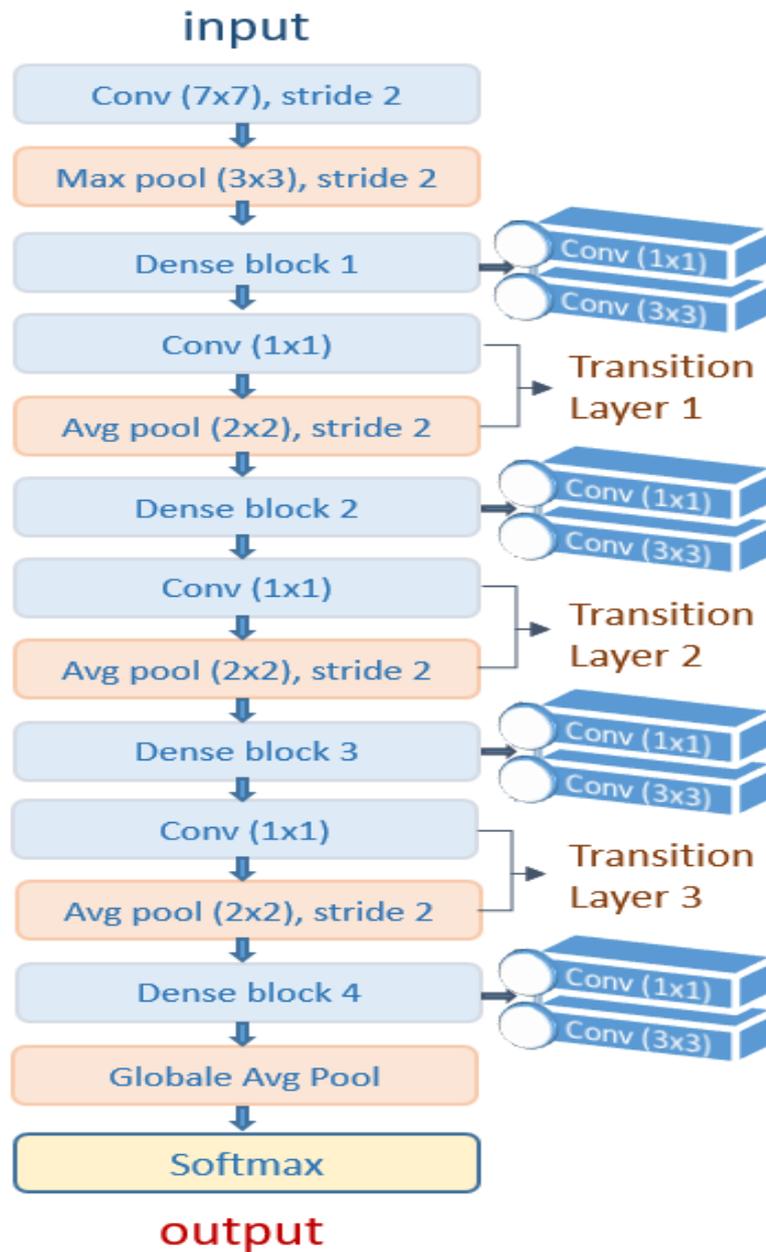


FIGURE 3.3 – Architecture du DenseNet201 [20].

une couche de convolution, suivi d'une couche GlobalAveragePooling, puis ajouté trois couches entièrement connectées de 256, 128, et 64 neurones. La dernière couche de classification contient deux neurones et utilise la fonction Softmax. Tous les neurones d'une couche sont connectés à ceux de la couche sui-

vante. Pour prévenir un surapprentissage de notre modèle, nous avons intégré des couches Dropout fixées à 20%, 30% et 30%, respectivement. Afin de fournir une preuve visuelle et d'améliorer l'explicabilité de notre architecture, mettant en exergue les régions pertinentes influençant la décision de notre modèle, nous avons fait appel à l'algorithme Grad-CAM. Sans annotations manuelles supplémentaires, cette carte thermique est générée entièrement par le modèle DL. Les prédictions du réseau sont traduites par des cartes thermiques illustrant les zones les plus représentatives de l'image, grâce à des cartes d'activation de classes pondérées par le gradient. La méthodologie suivie est synthétisée dans l'organigramme de la figure 3.4.

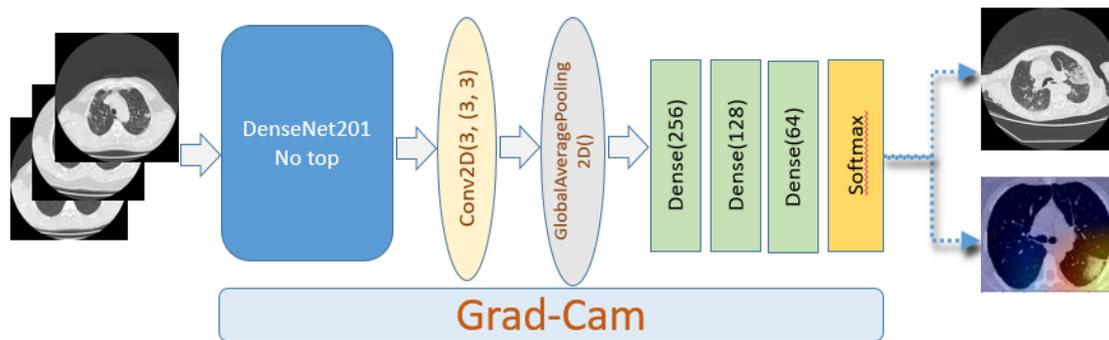


FIGURE 3.4 – Organigramme du modèle proposé [20].

### 3.5 Résultats et discussion

Dans ce travail, nous avons collecté un nouvel ensemble de données de l'hôpital de Tlemcen en Algérie. Chaque image a été étiquetée (positive ou négative) par un radiologue et cette étiquetage a été confirmé par un test RT-PCR. Les données ont ensuite été prétraitées. Nous avons redimensionné toutes les images à une taille de 128x128. Pour assurer une meilleure généralisation, nous avons recouru à des techniques d'augmentation de données, en appliquant des transformations aléatoires sur les images d'entrée. Afin de garantir la diversité des données, plusieurs transformations telles que la rotation et le zoom ont été effectuées, avec une probabilité variant de 0 à 1. Cette étude compare le modèle que nous proposons avec cinq autres architectures : VGG16, VGG19, Xception, Inception V2 Resnet, et DenseNet121. Tous les modèles ont été entraînés en utilisant des lots de taille 32 pendant 200 époques, en ayant recours à la méthode de descente stochastique du gradient (SGD) pour minimiser la fonction de perte.

Les résultats ont été comparés en se basant sur différentes métriques comme l'Accuracy 3.3, la précision 3.4, le rappel 3.5, et le F1-score 3.7.

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.4)$$

$$SN = \frac{TP}{TP + FN} \quad (3.5)$$

$$SP = 1 - \frac{FP}{FP + TN} = \frac{TN}{TN + FP} \quad (3.6)$$

$$F1 - score = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (3.7)$$

Où :

P : nombre total de patients COVID-19.

N : nombre total de patients non COVID-19.

TP (Vrai positif) : la prédiction indique COVID et le patient est effectivement infecté par le virus.

TN (Vrai négatif) : la prédiction indique Non-COVID et le patient est effectivement en bonne santé.

FP (Faux positif) : la prédiction indique COVID alors que le patient est en bonne santé.

FN (Faux négatif) : la prédiction indique Non-COVID alors que le patient est infecté par le virus.

Le Tableau 3.2 met en évidence les performances réalisées par les différents modèles examinés. L'approche proposé se démarque nettement des autres modèles en termes d'accuracy. Habituellement, l'accuracy est la métrique de référence lorsqu'il s'agit d'évaluer un modèle, car elle donne un aperçu global de sa performance. Néanmoins, elle peut s'avérer trompeuse, surtout quand il y a un déséquilibre des classes, comme c'est le cas dans notre dataset où les cas positifs sont inférieurs aux cas négatifs. C'est là que la précision et les autres métriques entrent en jeu.

La métrique de précision, représentant le ratio de vrais positifs par rapport à l'ensemble des prédictions positives, est d'une importance capitale dans le secteur médical. Un haut niveau de précision garantit que les diagnostics positifs correspondent réellement à des cas d'infection. De plus, notre modèle brille par son excellent taux de rappel, minimisant de fait le nombre de faux négatifs. Ce genre d'erreur, où un patient infecté est erronément classifié comme sain, peut avoir des répercussions dramatiques, tant pour l'individu que pour la société

en raison du risque de propagation du virus.

Le tableau 3.3 met également en lumière de bons chiffres en ce qui concerne les faux positifs. Bien que moins critiques, ces erreurs peuvent néanmoins provoquer du stress inutile chez le patient.

En synthèse, notre évaluation montre que le modèle Densenet201 est supérieur en termes de performances, avec un équilibre parfait entre précision et rappel, comme l'indique son F1-score élevé. Cette métrique, qui englobe à la fois la précision et le rappel, donne une vue d'ensemble de l'efficacité du modèle. Dans le contexte de la détection de la COVID-19, cela garantit un diagnostic à la fois fiable et précis. En conclusion, le modèle que nous proposons est non seulement précis, mais aussi hautement sensible à la détection de la COVID-19, tout en réduisant au maximum les faux positifs et négatifs. Cette prouesse est primordiale dans le combat contre la pandémie, car elle offre une détection rapide et fiable des cas, un atout majeur pour orienter les décisions en santé publique.

Modèle	Accuracy	Précision	Rappel	Spécificité	F1-score	AUC
VGG16	94.69%	96.21%	95.12%	94.00%	95.67%	94.56%
VGG19	94.38%	93.47%	97.72%	89.03%	95.55%	93.38%
Xception	96.59%	97.08%	97.4%	95.3%	97.24%	96.35%
Inception_V2_Resnet	96.49%	97.54%	96.75%	96.08%	97.14%	96.42%
DenseNet121	96.99%	98.99%	96.09%	98.43%	97.52%	97.27%
DenseNet201	98.1%	99.01%	97.89%	98.43%	98.45%	98.16%
<b>Le modèle proposée</b>	<b>98.8%</b>	<b>99.50%</b>	<b>98.54%</b>	<b>99.22%</b>	<b>99.02%</b>	<b>98.88%</b>

TABLE 3.2 – Comparaisons des résultats entre les différentes architectures appliquées à notre jeu de données.

La figure 3.5 montre les courbes de l'Accuracy d'entraînement et de validation du travail proposé. La première remarque est que notre méthode proposée Densenet201 a dépassé 90% juste après la 15ème époque et gagne plus d'Accuracy. Cela montre clairement que le modèle proposé peut être utilisé efficacement pour le diagnostic précoce de COVID-19. Nous pouvons également remarquer que la plupart des modèles ont bien appris de l'ensemble de données et ne souffrent ni de sur-apprentissage ni de sous-apprentissage. Cela peut

Modèle	TP	FN	FP	TN
VGG16	585	30	23	360
VGG19	601	14	42	341
Xception	599	16	18	365
Inception_				
V2_Resnet	595	20	15	368
DenseNet121	593	24	6	375
DenseNet201	602	13	6	377
<b>Le modèle proposé</b>	<b>606</b>	<b>9</b>	<b>3</b>	<b>380</b>

TABLE 3.3 – TP, TN, FP et FN obtenu à partir de l'ensemble de données de test.

s'expliquer par le fait que SGD contient l'erreur, reste stable et empêche le sur-apprentissage même sans régularisation.

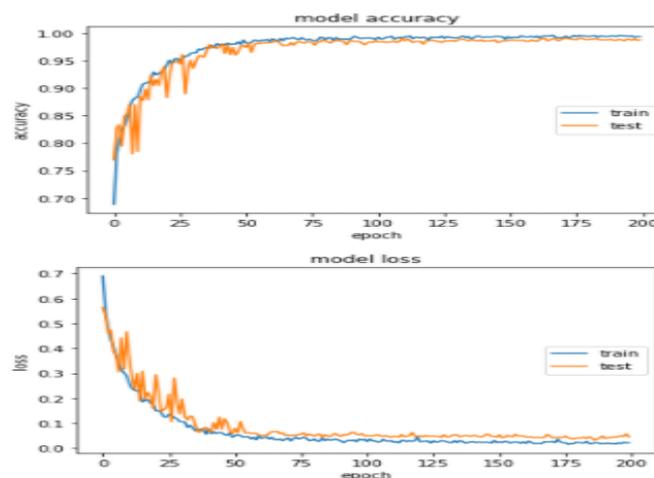


FIGURE 3.5 – Analyse d'Accuracy de l'entraînement et du test et analyse de la perte de l'entraînement et du test sur 200 époques.

Afin de prouver la robustesse du modèle proposé, nous avons utilisé une autre base de données publique [21]. Le tableau 3.4 présente les résultats des performances. La sensibilité obtenue est de 98,20 % et la spécificité de 98,17 % pour les cas COVID-19 positifs. Les résultats obtenus nous permettent de conclure que le modèle proposé peut détecter correctement les vrais positifs (c'est-à-dire les patients infectés par COVID-19) avec un Accuracy de 98,18 %. En outre, le modèle proposé a montré de bonnes performances en termes de

précision et de F1-score.

Modèle	Accuracy	Précision	Rappel	Spécificité	F1-score	AUC
VGG16	94.37%	91.49%	96.41%	92.70%	93.89%	94.96%
VGG19	96.37%	95.96%	95.96%	96.72%	95.96%	96.33%
Xception	91.95%	91.40%	90.58%	93.07%	91.00%	91.82%
Inception_V2_Resnet	94.57%	93.36%	94.62%	94.52%	93.98%	91.82%
DenseNet121	95.77%	91.73%	<b>99.55%</b>	92.73%	95.48%	96.12%
DenseNet201	97.38%	95.26%	99.10%	95.98%	97.14%	97.54%
<b>Le modèle proposée</b>	<b>98.18%</b>	<b>97.76%</b>	98.20%	<b>98.17%</b>	<b>97.98%</b>	<b>98.82%</b>

TABLE 3.4 – Comparaisons des résultats entre les différentes architectures appliquées à l'ensemble de données de [21].

Pour convaincre les médecins de la précision et de l'utilité de l'outil que nous proposons, et afin de visualiser les zones qui motivent la décision du modèle, nous avons appliqué l'algorithme Grad-CAM. La figure 3.6 montre les cartes thermiques sur les zones suspectes, qui prouvent que notre algorithme se concentre sur les zones infectées en négligeant les autres régions normales. Les prédictions du réseau sont interprétées sur la base des cartes thermiques pour visualiser les zones les plus significatives d'une image en appliquant des cartes d'activation de classe pondérées par le gradient. Pour mieux évaluer les performances de notre modèle, nous avons tracé la courbe ROC (Receiver Operating Characteristic) et calculé l'AUC (Area Under the Curve). La courbe ROC est un graphique qui permet d'affirmer l'efficacité de classification d'un modèle en fonction des vrais positifs et des faux positifs. Comme le montre la figure 3.7, le modèle proposé par DTL avec Densenet201 offre une AUC de 98,88%. Nous remarquons que ce modèle de classification fournit de meilleurs résultats que les modèles de classification existants, comme le montre le tableau 3.4.

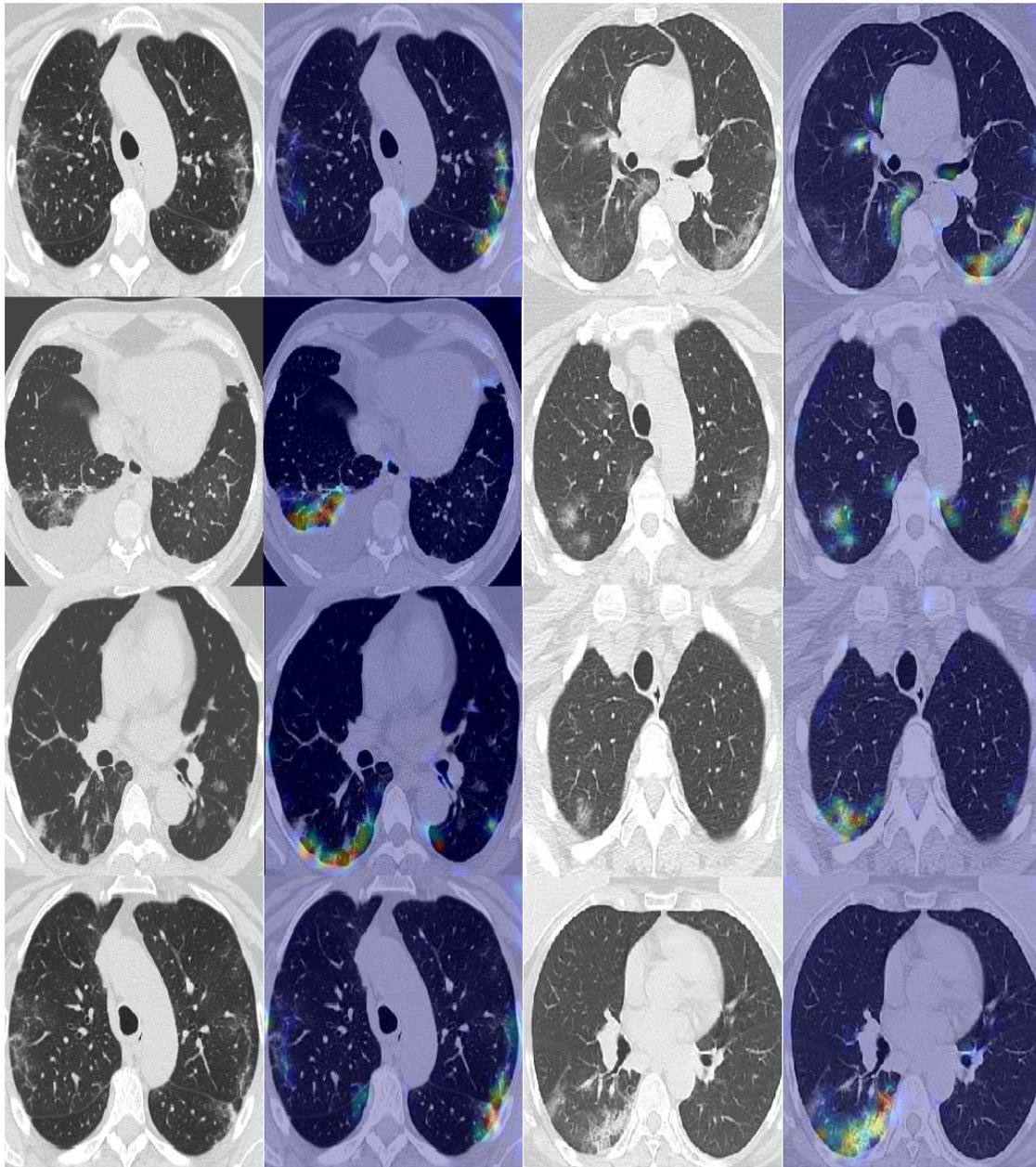


FIGURE 3.6 – Explication avec la carte thermique Grad-cam sur des cas positifs de COVID-19

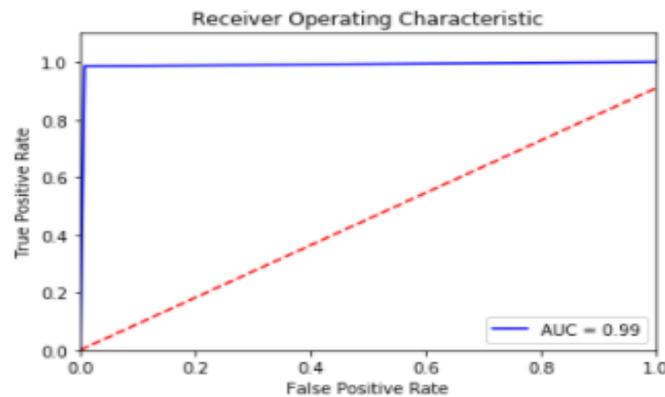


FIGURE 3.7 – Aire sous la courbe (AUC) du modèle proposé.

### 3.6 Conclusion

L'identification et l'isolement précoces des cas de COVID-19 hautement infectieux jouent un rôle essentiel dans la prévention de la propagation du virus. L'imagerie par rayons X est une méthode peu coûteuse, facilement accessible et rapide qui peut constituer une excellente alternative aux méthodes de diagnostic conventionnelles telles que la RT-PCR et la tomodensitométrie. De nombreuses études ont proposé des solutions basées sur l'IA pour la détection automatique et en temps réel du COVID-19, démontrant des performances remarquables pour le diagnostic précoce.

Cependant, ces méthodes ont montré des performances limitées dans la localisation de l'infection et le classement de la gravité de la pneumonie COVID-19. Dans ce chapitre, nous avons présenté une méthode intelligente basée sur l'apprentissage en profondeur. Nous avons réalisé une analyse comparative et une étude sur différents modèles pour classer les images de tomodensitométrie du COVID-19. Pour cela, nous avons collecté un nouvel ensemble de données de tomodensitométrie de l'hôpital de Tlemcen en Algérie.

En appliquant un apprentissage par transfert sur six architectures DL bien connues (VGG16, VGG19, Xception, Inception V2 Resnet, DenseNet121, DenseNet201), nous avons proposé un modèle basé sur DenseNet201 en appliquant une augmentation des données pour obtenir plus d'images. Pour comprendre comment notre modèle convolutif prend une décision de classification et fournir des preuves visuelles, nous avons utilisé l'algorithme GradCam.

Les expériences réalisées ont démontré la fiabilité de l'architecture proposée avec une précision de 98,8%. Elle fournit également une explication visuelle, prouvant que le modèle proposé peut être considéré comme une alternative pour le dépistage du COVID-19 et le suivi des patients. Ainsi, les méthodes DL peuvent contribuer à la lutte contre la propagation du virus.

Les travaux futurs pourraient également étudier l'architecture YOLO pour obtenir les meilleurs résultats. De plus, nous avons l'intention d'ajouter d'autres classes contenant d'autres maladies pulmonaires afin d'aider les experts en santé et les médecins généralistes.

# Conclusion générale

L'imagerie médicale, au cœur des avancées technologiques du XXI<sup>e</sup> siècle, est devenue une composante essentielle du diagnostic médical moderne. Pourtant, malgré ses avancées, le secteur médical est confronté à des défis considérables. Le manque de spécialistes pour analyser ces images, associé au temps nécessaire à leur interprétation, pose des contraintes majeures, notamment en période de pandémies comme celle du COVID-19. Dans un tel contexte, où chaque minute est précieuse, la nécessité d'une intervention technologique, en particulier celle de l'intelligence artificielle, devient non seulement bénéfique, mais essentielle.

Pour répondre à ces besoins, cette thèse s'est articulée autour de deux axes principaux : l'analyse d'images radiographiques X-ray pour la classification de la pneumonie et l'analyse d'images scanner CT-scans pour la classification du COVID-19. Ces deux axes reflètent l'importance grandissante de l'intelligence artificielle dans l'optimisation du diagnostic médical à l'aide d'images.

Concernant la pneumonie, notre recherche a mis en avant une approche innovante qui s'appuie sur les réseaux de neurones convolutifs (CNN) pour accélérer le diagnostic par radiographie, une pratique courante dans le milieu hospitalier. Notre modèle, publié dans la conférence sous le titre "Convolutional Neural Network for Chest X-ray Pneumonia Detection", se distingue par son temps d'apprentissage optimisé et sa capacité à tester diverses dimensions de CNN. Son efficacité, mesurée en termes de précision, de sensibilité et de spécificité, tout en requérant moins de ressources calculatoires, le rend particulièrement adapté aux environnements de diagnostic. Cet article a obtenu la distinction du "Best Paper Award" lors de la conférence ISPR'2020 en Tunisie (International Conference on Intelligent Systems and Pattern Recognition).

Dans le volet du COVID-19, notre travail a ciblé l'importance de la détection précoce des infections. Grâce à une méthode basée sur l'apprentissage en profondeur et en collectant un ensemble de données de l'hôpital de Tlemcen,

nous avons présenté un modèle performant pour classer les images de tomodensitométrie. Ce modèle, exposé dans l'article "Deep transfer learning based classification model for covid-19 using chest CT-scans", utilise une architecture basée sur DenseNet201 et intègre l'augmentation des données pour accroître sa robustesse. Pour garantir sa transparence, nous avons employé la méthode GradCAM afin d'offrir une interprétation visuelle des décisions du modèle. Ses performances, évaluées à 98,8% de précision, valident son efficacité et son utilité pour le dépistage du COVID-19. Ce travail a été publié dans le journal "Pattern Recognition Letters" (classe A, facteur d'impact 5.1).

Les travaux réalisés au cours de cette thèse témoignent de la pertinence de combiner la puissance de l'apprentissage en profondeur avec l'analyse d'images médicales. Ils soulignent également l'importance de l'interprétabilité des décisions des modèles IA, afin de gagner la confiance des professionnels de santé.

Comme perspectives à notre travail de thèse de doctorat, nous proposons trois pistes de recherche distinctes :

- Elargissement de notre modèle de classification pour couvrir d'autres types de maladies pulmonaires, ce qui permettra une détection plus fine et un diagnostic fiable.
- Tester d'autres algorithmes d'apprentissage profond comme les transformateurs. Dans le but d'améliorer la détection et l'interprétation des images médicales.
- Elargir les caractéristiques des données cliniques comme l'âge, le sexe et les rapports médicaux des patients. Cette tâche permettra d'améliorer les résultats de la classification.

# Bibliographie

- [1] Pramila P Shinde and Seema Shah, "A review of machine learning and deep learning applications," in *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE, 2018, pp. 1–6.
- [2] "neurone biologique," <https://www.javatpoint.com/artificial-neural-network>.
- [3] Balaji Sathelly, *An Artificial Neural Network Approach to Predict Liver Failure Likelihood*, The University of Toledo, 2018.
- [4] Imdat Taymaz and Yasar Islamoglu, "Prediction of convection heat transfer in converging–diverging tube for laminar air flowing using back-propagation neural network," *International communications in heat and mass transfer*, vol. 36, no. 6, pp. 614–617, 2009.
- [5] Jedediah M Berhold, "Convolutional neural network architecture study for aerial visual localization," Tech. Rep., AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH, 2019.
- [6] Emmanuel Ovalle-Magallanes, Juan Gabriel Avina-Cervantes, Ivan Cruz-Aceves, and Jose Ruiz-Pinales, "Transfer learning for stenosis detection in x-ray coronary angiography," *Mathematics*, vol. 8, no. 9, pp. 1510, 2020.
- [7] Daphne Cornelisse, "An intuitive guide to convolutional neural networks," *free Code Camp*, pp. 1–19, 2018.
- [8] Sumit Saha, "A comprehensive guide to convolutional neural networks—the eli5 way," *Towards data science*, vol. 15, 2018.
- [9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, 2016.
- [10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

- 
- [11] Muneeb ul Hassan, "Alexnet imagenet classification with deep convolutional neural networks," 2018.
- [12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [13] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.
- [15] Burak Gülmez, "A novel deep neural network model based xception and genetic algorithm for detection of covid-19 from x-ray images," *Annals of Operations Research*, pp. 1–25, 2022.
- [16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.
- [17] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian, "Grad-cam++ : Improved visual explanations for deep convolutional networks.," 2018.
- [18] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
- [19] Ilyas Lahsaini, Mostafa El Habib Daho, and Mohammed Amine Chikh, "Convolutional neural network for chest x-ray pneumonia detection," in *Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition*, 2020, pp. 55–59.
- [20] Ilyas Lahsaini, Mostafa El Habib Daho, and Mohamed Amine Chikh, "Deep transfer learning based classification model for covid-19 using chest ct-scans," *Pattern Recognition Letters*, vol. 152, pp. 122–128, 2021.

- [21] Plamen Angelov and Eduardo Almeida Soares, "Sars-cov-2 ct-scan dataset : A large dataset of real patients ct scans for sars-cov-2 identification," *medRxiv*, 2020.
- [22] Joel D Howell, "Early clinical use of the x-ray," *Transactions of the American Clinical and Climatological Association*, vol. 127, pp. 341, 2016.
- [23] "Johns hopkins university. covid-19 dashboard by the center for systems science and engineering (csse)," <https://coronavirus.jhu.edu/map.html>, Accessed : 2020.
- [24] "Coronavirus. world health organization website," <https://www.who.int/health-topics/coronavirus>, Published 2020. Accessed March 24, 2020.
- [25] Sarah Cobey, "Modeling infectious disease dynamics," *Science*, vol. 368, no. 6492, pp. 713–714, 2020.
- [26] "trump white house website," <https://trumpwhitehouse.archives.gov/briefings-statements/call-action-tech-community-new-machine-readable-covid-19-dataset/>, Published 2020. Accessed March 16, 2020.
- [27] Ahmad Alimadadi, Sachin Aryal, Ishan Manandhar, Patricia B Munroe, Bina Joe, and Xi Cheng, "Artificial intelligence and machine learning to fight covid-19," 2020.
- [28] Sonali B Maind, Priyanka Wankar, et al., "Research paper on basic of artificial neural network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 1, pp. 96–100, 2014.
- [29] Jung Me Park and Hong Tae Kang, "Prediction of fatigue life for spot welds using back-propagation neural networks," *Materials & design*, vol. 28, no. 10, pp. 2577–2584, 2007.
- [30] Laurene Fausett, "Fundamentals of neural networks prentice hall," *Englewood Cliffs, NJ*, vol. 7632, 1994.
- [31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout : a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

- 
- [32] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller, "Efficient backprop," in *Neural networks : Tricks of the trade*, pp. 9–48. Springer, 2012.
- [33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [34] Arden Dertat, "Applied deep learning-part 4 : Convolutional neural networks," *Towards Data Science*, vol. 26, 2017.
- [35] Sergey Ioffe and Christian Szegedy, "Batch normalization : Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
- [36] Ning Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [37] Yurii E Nesterov, "A method for solving the convex programming problem with convergence rate  $o(1/k^2)$ ," in *Dokl. akad. nauk Sssr*, 1983, vol. 269, pp. 543–547.
- [38] Matthew D Zeiler, "Adadelata : an adaptive learning rate method," *arXiv preprint arXiv :1212.5701*, 2012.
- [39] Yann Dauphin, Harm De Vries, and Yoshua Bengio, "Equilibrated adaptive learning rates for non-convex optimization," *Advances in neural information processing systems*, vol. 28, 2015.
- [40] Diederik P Kingma and Jimmy Ba, "Adam : A method for stochastic optimization," *arXiv preprint arXiv :1412.6980*, 2014.
- [41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet : A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [42] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.
- [43] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv :1409.1556*, 2014.

- [44] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2017, vol. 31.
- [45] François Chollet, "Xception : Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807.
- [46] Beijing Chen, Xin Liu, Yuhui Zheng, Guoying Zhao, and Yun-Qing Shi, "A robust gan-generated face detection method based on dual-color spaces and an improved xception," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3527–3538, 2022.
- [47] Beijing Chen, Xingwang Ju, Bin Xiao, Weiping Ding, Yuhui Zheng, and Victor Hugo C de Albuquerque, "Locally gan-generated face detection based on an improved xception," *Information Sciences*, vol. 572, pp. 16–28, 2021.
- [48] Kashif Shaheed, Aihua Mao, Imran Qureshi, Munish Kumar, Sumaira Husain, Inam Ullah, and Xingming Zhang, "Ds-cnn : A pre-trained xception model based on depth-wise separable convolutional neural network for finger vein recognition," *Expert Systems with Applications*, vol. 191, pp. 116288, 2022.
- [49] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam : Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [50] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian, "Grad-cam++ : Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.
- [51] Stefan Jaeger, Alexandros Karargyris, Sema Candemir, Les Folio, Jenifer Siegelman, Fiona Callaghan, Zhiyun Xue, Kannappan Palaniappan, Rahul K Singh, Sameer Antani, et al., "Automatic tuberculosis screening using chest radiographs," *IEEE transactions on medical imaging*, vol. 33, no. 2, pp. 233–245, 2013.
- [52] "Who pneumonia. world health organization," <https://www.who.int/news-room/fact-sheets/detail/pneumonie>, Published 2019.

- [53] Mark I Neuman, Edward Y Lee, Sarah Bixby, Stephanie Diperna, Jeffrey Hellinger, Richard Markowitz, Sabah Servaes, Michael C Monuteaux, and Samir S Shah, "Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children," *Journal of hospital medicine*, vol. 7, no. 4, pp. 294–298, 2012.
- [54] Gabrielle J Williams, Petra Macaskill, Marianne Kerr, Dominic A Fitzgerald, David Isaacs, Miriam Codarini, Mary McCaskill, Kristina Prelog, and Jonathan C Craig, "Variability and accuracy in interpretation of consolidation on chest radiography for diagnosing pneumonia in children under 5 years of age," *Pediatric Pulmonology*, vol. 48, no. 12, pp. 1195–1200, 2013.
- [55] Sheeba Lal, Saeed Ur Rehman, Jamal Hussain Shah, Talha Meraj, Hafiz Tayyab Rauf, Robertas Damaševičius, Mazin Abed Mohammed, and Karrar Hameed Abdulkareem, "Adversarial attack and defence through adversarial training and feature fusion for diabetic retinopathy recognition," *Sensors*, vol. 21, no. 11, pp. 3922, 2021.
- [56] Hafiz Tayyab Rauf, M Lali, Muhammad Attique Khan, Seifedine Kadry, Hanan Alolaiyan, Abdul Razaq, and Rizwana Irfan, "Time series forecasting of covid-19 transmission in asia pacific countries using deep neural networks," *Personal and Ubiquitous Computing*, pp. 1–18, 2021.
- [57] Saleh Albahli, Hafiz Tayyab Rauf, Abdulelah Algosaibi, and Valentina Emilia Balas, "Ai-driven deep cnn approach for multi-label pathology classification using chest x-rays," *PeerJ Computer Science*, vol. 7, pp. e495, 2021.
- [58] Saleh Albahli, Hafiz Tayyab Rauf, Muhammad Arif, Md Tabrez Nafis, and Abdulelah Algosaibi, "Identification of thoracic diseases by exploiting deep neural networks," *Neural Netw*, vol. 5, no. 6, 2021.
- [59] Tej Bahadur Chandra and Kesari Verma, "Pneumonia detection on chest x-ray using machine learning paradigm," in *Proceedings of 3rd international conference on computer vision and image processing*. Springer, 2020, pp. 21–33.
- [60] Kuang Ming Kuo, Paul C Talley, Chi Hsien Huang, and Liang Chih Cheng, "Predicting hospital-acquired pneumonia among schizophrenic patients : a machine learning approach," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, pp. 1–8, 2019.
- [61] Hongmei Yue, Qian Yu, Chuan Liu, Yifei Huang, Zicheng Jiang, Chuxiao Shao, Hongguang Zhang, Baoyi Ma, Yuancheng Wang, Guanghang Xie,

- et al., "Machine learning-based ct radiomics method for predicting hospital stay in patients with pneumonia associated with sars-cov-2 infection : a multicenter study," *Annals of translational medicine*, vol. 8, no. 14, 2020.
- [62] Venkatesan Rajinikanth, Seifedine Kadry, Robertas Damaševičius, David Taniar, and Hafiz Tayyab Rauf, "Machine-learning-scheme to detect choroidal-neovascularization in retinal oct image," in *2021 seventh international conference on bio signals, images, and instrumentation (ICBSII)*. IEEE, 2021, pp. 1–5.
- [63] Seifedine Kadry, Yunyoung Nam, Hafiz Tayyab Rauf, Venkatesan Rajinikanth, and Isah A Lawal, "Automated detection of brain abnormality using deep-learning-scheme : A study," in *2021 Seventh International Conference On Bio Signals, Images, And Instrumentation (ICBSII)*. IEEE, 2021, pp. 1–5.
- [64] Venkatesan Rajinikanth, Seifedine Kadry, David Taniar, Robertas Damaševičius, and Hafiz Tayyab Rauf, "Breast-cancer detection using thermal images with marine-predators-algorithm selected features," in *2021 seventh international conference on bio signals, images, and instrumentation (ICBSII)*. IEEE, 2021, pp. 1–6.
- [65] Hafida Belfilali, Frédéric Bousefsaf, and Mahammed Messadi, "Left ventricle analysis in echocardiographic images using transfer learning," *Physical and Engineering Sciences in Medicine*, vol. 45, no. 4, pp. 1123–1138, 2022.
- [66] Harsh Sharma, Jai Sethia Jain, Priti Bansal, and Sumit Gupta, "Feature extraction and classification of chest x-ray images using cnn to detect pneumonia," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 227–231.
- [67] Okeke Stephen, Mangal Sain, Uchenna Joseph Maduh, and Do-Un Jeong, "An efficient deep learning approach to pneumonia classification in healthcare," *Journal of healthcare engineering*, vol. 2019, 2019.
- [68] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al., "Chexnet : Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv :1711.05225*, 2017.
- [69] Holger R Roth, Le Lu, Ari Seff, Kevin M Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M Summers, "A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations," in *Medical Image Computing and*

- Computer-Assisted Intervention–MICCAI 2014 : 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part I 17*. Springer, 2014, pp. 520–527.
- [70] Lixin Liang, Gaobo et Zheng, “Une méthode d’apprentissage par transfert avec un réseau résiduel profond pour le diagnostic de pneumonie pédiatrique,” *Méthodes et programmes informatiques en biomédecine*, vol. 187, pp. 104964.
- [71] Halil Murat Ayan, Enes et Ünver, “Diagnostic de pneumonie à partir d’images radiographiques pulmonaires utilisant le deep learning,” in *Réunion scientifique 2019 sur l’électro-électronique & le génie biomédical et l’informatique (EBBT)*, pp. 1–5.
- [72] Chaolin Huang, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, et al., “Clinical features of patients infected with 2019 novel coronavirus in wuhan, china,” *The lancet*, vol. 395, no. 10223, pp. 497–506, 2020.
- [73] Fan Wu, Su Zhao, Bin Yu, Yan-Mei Chen, Wen Wang, Zhi-Gang Song, Yi Hu, Zhao-Wu Tao, Jun-Hua Tian, Yuan-Yuan Pei, et al., “A new coronavirus associated with human respiratory disease in china,” *Nature*, vol. 579, no. 7798, pp. 265–269, 2020.
- [74] Wenling Wang, Yanli Xu, Ruqin Gao, Roujian Lu, Kai Han, Guizhen Wu, and Wenjie Tan, “Detection of sars-cov-2 in different types of clinical specimens,” *Jama*, vol. 323, no. 18, pp. 1843–1844, 2020.
- [75] Yadunath Pathak, Prashant Kumar Shukla, Akhilesh Tiwari, Shalini Stalin, and Saurabh Singh, “Deep transfer learning based classification model for covid-19 disease,” *Irbm*, vol. 43, no. 2, pp. 87–92, 2022.
- [76] Zi Yue Zu, Meng Di Jiang, Peng Peng Xu, Wen Chen, Qian Qian Ni, Guang Ming Lu, and Long Jiang Zhang, “Coronavirus disease 2019 (covid-19) : a perspective from china,” *Radiology*, vol. 296, no. 2, pp. E15–E25, 2020.
- [77] Hema Shekar Basavegowda and Guesh Dagneu, “Deep learning approach for microarray cancer data classification,” *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 22–33, 2020.
- [78] Dilbag Singh, Vijay Kumar, and Manjit Kaur, “Classification of covid-19 patients from chest ct images using multi-objective differential evolution-based convolutional neural networks,” *European Journal of Clinical Microbiology & Infectious Diseases*, vol. 39, pp. 1379–1389, 2020.

- [79] Amit Kumar Jaiswal, Prayag Tiwari, Sachin Kumar, Deepak Gupta, Ashish Khanna, and Joel JPC Rodrigues, "Identifying pneumonia in chest x-rays : A deep learning approach," *Measurement*, vol. 145, pp. 511–518, 2019.
- [80] Paras Lakhani and Baskaran Sundaram, "Deep learning at chest radiography : automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.
- [81] Sangheum Hwang, Hyo-Eun Kim, Jihoon Jeong, and Hee-Jin Kim, "A novel approach for tuberculosis screening based on deep convolutional neural networks," in *Medical imaging 2016 : computer-aided diagnosis*. SPIE, 2016, vol. 9785, pp. 750–757.
- [82] Guanqiu Qi, Huan Wang, Matthew Haner, Chenjie Weng, Sixin Chen, and Zhiqin Zhu, "Convolutional neural network based detection and judgement of environmental obstacle in vehicle operation," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 80–91, 2019.
- [83] Bhawna Nigam, Ayan Nigam, Rahul Jain, Shubham Dodia, Nidhi Arora, and B Annappa, "Covid-19 : Automatic detection from x-ray images by utilizing deep learning methods," *Expert Systems with Applications*, vol. 176, pp. 114883, 2021.
- [84] Shayan Hassantabar, Mohsen Ahmadi, and Abbas Sharifi, "Diagnosis and detection of infected tissue of covid-19 patients based on lung x-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals*, vol. 140, pp. 110170, 2020.
- [85] Ezz El-Din Hemdan, Marwa A Shouman, and Mohamed Esmail Karar, "Covidx-net : A framework of deep learning classifiers to diagnose covid-19 in x-ray images," *arXiv preprint arXiv :2003.11055*, 2020.
- [86] Mesut Toğaçar, Burhan Ergen, and Zafer Cömert, "Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches," *Computers in biology and medicine*, vol. 121, pp. 103805, 2020.
- [87] Jianpeng Zhang, Yutong Xie, Yi Li, Chunhua Shen, and Yong Xia, "Covid-19 screening on chest x-ray images using deep learning based anomaly detection," *arXiv preprint arXiv :2003.12338*, vol. 27, pp. 141, 2020.
- [88] Ioannis D Apostolopoulos and Tzani A Mpesiana, "Covid-19 : automatic detection from x-ray images utilizing transfer learning with convolutional

- neural networks," *Physical and engineering sciences in medicine*, vol. 43, pp. 635–640, 2020.
- [89] Xiaowei Xu, Xiangao Jiang, Chunlian Ma, Peng Du, Xukun Li, Shuangzhi Lv, Liang Yu, Qin Ni, Yanfei Chen, Junwei Su, et al., "A deep learning system to screen novel coronavirus disease 2019 pneumonia," *Engineering*, vol. 6, no. 10, pp. 1122–1129, 2020.
- [90] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, et al., "A deep learning algorithm using ct images to screen for corona virus disease (covid-19)," *European radiology*, vol. 31, pp. 6096–6104, 2021.
- [91] Ali Narin, Ceren Kaya, and Ziyneet Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, pp. 1207–1220, 2021.
- [92] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya, "Automated detection of covid-19 cases using deep neural networks with x-ray images," *Computers in biology and medicine*, vol. 121, pp. 103792, 2020.
- [93] Morteza Heidari, Seyedehnafiseh Mirniaharikandehei, Abolfazl Zargari Khuzani, Gopichandh Danala, Yuchen Qiu, and Bin Zheng, "Improving the performance of cnn to predict the likelihood of covid-19 using chest x-ray images with preprocessing algorithms," *International journal of medical informatics*, vol. 144, pp. 104284, 2020.
- [94] Umashankar Subramaniam, M Monica Subashini, Dhafer Almakhlles, Alagar Karthick, and S Manoharan, "An expert system for covid-19 infection tracking in lungs using image processing and deep learning techniques," *BioMed Research International*, vol. 2021, pp. 1–17, 2021.
- [95] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers, "Chestx-ray8 : Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2097–2106.