



République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de master 2

en Informatique

Spécialité : Génie système d'informatique(SI)

Thème

Detection du Covid-19 à l'aide d'approches baseés sur
l'apprentissage en profondeur

Encadré par

— MOULOUD DEMOUCHE

Réalisé par

— IDOU YAZID

— CHAAB ZAKARIA NOUH

2021/2022

Remerciements

*Tout d'abord et avant tout, nous remercions **ALLAH**, le tout-puissant de nous avoir donné la patience, la santé et la volonté pour réaliser ce mémoire.*

Un grand merci à nos chers parents et nos familles qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Nous tenons à adresser nos sincères remerciements et le grand respect à notre encadreur, Mr. DEMOUCHE Mouloud, pour sa disponibilité, ses conseils, sa gentillesse et toute l'aide qu'il nous ont rapportés.

Nous remercions également tous les membres du jury d'avoir accepté à participer à l'évaluation de notre travail.

Nous adressons toutes sympathie à tous nos collègues et nos amis pour leurs encouragements et pour tous les moments agréables qu'on passés ensemble.

Finalement, nous aimerions aussi remercier tous ceux qui nos ont encouragé et supporté avec les hauts et les bas tout au long de ce travail. Merci à tous et toutes.

Dédicaces

Je tiens à dédier ce modeste travail :

À Mes très chers parents et frères, qui m'ont soutenu, encouragé pour que je puisse mener à bien mes études, et qui attendent ce jour avec impatience.

À mon binôme Yazid Idou ;

À toute ma famille, à tous ceux qui m'aiment et que j'aime.

À chers amis : Sami, Amar, Tarek, Djamel, Sofian, Fateh, Mhamed, Islam, Mondir....

À tous mes amis d'enfance et du long parcours scolaire et universitaire.

Chaab Zakaria Nouh.

Dédicaces

pour que ma réussite soit complète je la partage avec tous les personnes que j'aime, je dédié ce modeste travail :

À mes très chers parents, qui m'ont soutenu, encouragé pour que je puisse mener à bien mes études, et qui attendu ce jour avec impatience.

À tous mes frères, mes sœurs et toute ma famille, en témoignage de mon profond respect.

À mon binôme Zakaria.

À tous mes amis sans exception, Sami, Amine, Amar, Yasser, Khalil, Bachar, Imad..

À tous mes collègues que j'ai partagés avec eux mes bons moments tout au long de mes études.

idou yazid.

Résumé

Le virus COVID-19 s'est largement répandu dans le monde depuis le début de 2020. L'Organisation mondiale de la santé (OMS) a déclaré une urgence sanitaire mondiale. Les personnes atteintes de COVID-19 présentent des symptômes associés à la pneumonie et le virus affecte les organes respiratoires du corps, ce qui rend la respiration difficile. Un tableau de réaction en chaîne par polymérase de transcription inverse (RT-PCR) est utilisé pour diagnostiquer la maladie. Pour développer une alternative, les radiologues ont examiné les changements dans l'imagerie radiographique, comme la tomодensitométrie « CT » thoracique, qui produit des images corporelles complètes d'excellente qualité.

L'objectif de ce travail est de proposer une approche d'apprentissage profond dans le domaine de l'épidémiologie pour détecter le virus Covid-19 à l'aide de la tomодensitométrie thoracique.

Dans ce projet, nous avons proposé trois approches de réseau de neurones convolutifs (CNN) et créé trois modèles basés sur ces approches(modèle de base "CNN à partir de zéro",modèle Xception"transfert d'apprentissage" ,model InceptionResNetV2"transfert d'apprentissage") que nous avons appliqués à un ensemble de données de tomодensitométrie thoracique accessible au public pour le dépistage du COVID-19, qui contient 1252 images CT positives pour l'infection au COVID-19 et 1230 images de patients non infectés, 2482 images au total répartis en formation, test et vérification. Nos modèles ont prouvé leur efficacité en atteignant une grande précision et F1-score ce qui est très prometteur.

Abstract

The COVID-19 virus has spread widely around the world since the beginning of 2020. The World Health Organization (WHO) has declared a global health emergency. People with COVID-19 have symptoms associated with pneumonia, and the virus affects the body's respiratory organs, making it difficult to breathe. A reverse transcription polymerase chain reaction (RT-PCR) array is used to diagnose the disease. To develop an alternative, radiologists have looked at changes in radiographic imaging, such as chest CT scans, which produce excellent quality full-body images.

The objective of this work is to propose a deep learning approach in the field of epidemiology to detect the Covid-19 virus using chest computed tomography.

In this project, we proposed three convolutional neural network (CNN) approaches and created three models based on these approaches (basic model "CNN from scratch", Xception model "transfer learning", InceptionResNetV2 model "transfer learning") which we applied to a - appears publicly available chest CT scan data for COVID-19 screening, which contains 1252 CT images positive for COVID-19 infection and 1230 images from uninfected patients, 2482 images in total split into training, test and verification. Our models have proven their efficiency by achieving high accuracy and F1-score which is very promising.

ملخص

انتشر فيروس كورونا على نطاق واسع في جميع أنحاء العالم منذ بداية عام 2020. أعلنت منظمة الصحة العالمية حالة طوارئ صحية عالمية. يعاني الأشخاص المصابون بـ كورونا من أعراض مرتبطة بالالتهاب الرئوي، ويؤثر الفيروس على أعضاء الجسم التنفسية، مما يجعل التنفس صعبًا. يتم استخدام صفيح تفاعل البوليميراز المتسلسل للنسخ العكسي لتشخيص المرض. لتطوير بديل، نظر أخصائيو الأشعة في التغييرات في التصوير الشعاعي، مثل التصوير المقطعي للصدر، والذي ينتج صورًا عالية الجودة لكامل الجسم.

الهدف من هذا العمل هو اقتراح نهج التعلم العميق في مجال علم الأوبئة للكشف عن فيروس كورونا باستخدام التصوير المقطعي للصدر.

في هذا المشروع، اقترحنا ثلاث طرق للشبكة العصبية التلافيفية وأنشأنا ثلاثة نماذج بناءً على هذه الأساليب التي طبقناها على مجموعة بيانات فحص الصدر بالأشعة المقطعية المتاحة للجمهور لفحص فيروس كورونا، والتي تحتوي على 1252 صورة مقطعية إيجابية لعدوى كورونا و1230 صورة لمرضى غير مصابين، 2482 صورة في المجموع مقسمة إلى تدريب واختبار وتحقق. حيث أثبتت نماذجنا كفاءتها من خلال تحقيق الدقة العالية وهو أمر واعد للغاية.

Table des matières

- Table des matières** **i**

- Table des figures** **iv**

- Liste des tableaux** **vi**

- Liste des abréviations** **vii**

- 1 Covid-19 et imagerie médicale** **3**
 - 1.1 Introduction : 3
 - 1.2 Maladies virales : 3
 - 1.3 COVID-19 4
 - 1.3.1 Définition 4
 - 1.3.2 Comment cela a-t-il commencé? 4
 - 1.3.3 Infection et Symptômes 5
 - 1.3.4 Diagnostique : 6
 - 1.4 Imagerie médicale 6
 - 1.4.1 définition 6
 - 1.4.2 Historique 7
 - 1.4.3 Types 8
 - 1.5 Comment détecter la COVID-19 dans les TDM thoracique "CT Scans" . . 13
 - 1.6 Conclusion 14

- 2 L'apprentissage profond «Deep Learning»** **15**
 - 2.1 Introduction : 15

2.2	Intelligence artificielle	15
2.3	Machine Learning «L'apprentissage automatique »	16
2.4	Méthodes d'apprentissage automatique	16
2.4.1	Apprentissage supervisé	17
2.4.2	Apprentissage non supervisé :	17
2.4.3	Apprentissage par renforcement :	18
2.5	algorithmes d'apprentissage automatique	19
2.5.1	KNN (k-Nearest Neighbors)	19
2.5.2	L'algorithme k-means (K-moyen) :	19
2.5.3	SVM (Support Vecteur Machine) :	20
2.6	L'apprentissage profond «Deep Learning»	20
2.6.1	Définition	20
2.6.2	Domaines d'application de l'apprentissage profonde :	21
2.7	Réseaux de neurones profonds :	22
2.7.1	Les réseaux de neurones convolutifs :	23
2.7.2	Réseau de neurones récurrents :	26
2.8	Conclusion	27
3	Matériels et méthodes.	28
3.1	Introduction :	28
3.2	Travaux connexes	28
3.2.1	Utilisation Un grand ensemble de données CT-scan des patients pour l'identification du SRAS-CoV-2	29
3.2.2	Système automatisé d'identification des infections au COVID-19 dans les images de tomodensitométrie à l'aide de modèles d'ap- prentissage en profondeur	29
3.2.3	Identification de la COVID-19 à partir d'images CT thoraciques une approche fondée sur les réseaux neuronaux convolutionnels pro- fonds	30
3.3	Ensemble de données « Dataset » utilisé	31
3.4	Fonction d'activation et Optimisateurs et fonction de perte utilisé	33
3.4.1	Fonction d'activation	33
3.4.2	Optimisateurs et fonction de perte	34

3.5	Approches proposées	35
3.5.1	Modèle de base :	35
3.5.2	Modèle Xception :	38
3.5.3	model InceptionResNetV2 :	40
3.6	Conclusion	42
4	Implementation et Tests	43
4.1	Introduction	43
4.2	Langage de programmation, Bibliothèques et Environnement utilisé	43
4.2.1	Langage de programmation	43
4.2.2	FrameWork	44
4.2.3	Environnement de développement	44
4.2.4	Bibliothèques utilisées	45
4.3	Mesures (métriques) de performance de classification	46
4.4	Implémentation	47
4.4.1	Prétraitement des ensembles de données	48
4.4.2	Modèles créés :	51
4.5	L'apprentissage et tests	54
4.5.1	L'apprentissage	54
4.5.2	Tests :	58
4.6	Analyse et discussion	60
4.6.1	Comparaison des modèles	61
4.6.2	Comparaison avec d'autres ouvrages	62
4.7	Conclusion	63
	Conclusion générale	64
	Bibliographie	66

Table des figures

1.1	gros plan sur la COVID-19	4
1.2	Techniques d'imagerie médicale.	8
1.3	Radiographie d'un bras cassé.	9
1.4	Image échographique d'un bébé à naître.. . . .	10
1.5	Exemple de scanners CT thoracique.	11
1.6	Exemple d'une IRM.	13
1.7	Images CT montrant le développement du coronavirus dans les 3 jours. . .	14
2.1	L'IA.	16
2.2	exemple d'apprentissage supervisé	17
2.3	exemple d'apprentissage non supervisé	18
2.4	exemple d'apprentissage par renforcement	18
2.5	Machine Learning vers Deep Learning.	21
2.6	Schéma d'un réseau de neurone profond.	23
2.7	Les réseaux de neurones convolutifs.	24
2.8	Les couches de CNN	24
2.9	Exemple de principe du filtre convolutionnel	24
2.10	Exemple de principe du Pooling	25
2.11	Principe de la couche entièrement connectée (FC)	25
2.12	Exemple montrant l'étiquette codée de la couche de sortie CNN	26
2.13	(à gauche) Un RNN. (à droite) Sa version déroulé Source : Nature[15] . . .	27
3.1	La distribution de notre ensemble de données.	31
3.2	Exemples d'images de "CT scan" COVID19 positives/négatives.	32

3.3	Architecture de Modèle de base.	35
3.4	diagramme principale de notre Modèle de base	36
3.5	L'architecture Xception.	38
3.6	Architecture Xception finale.	40
3.7	L'architecture InceptionResNetV2.	41
3.8	Architecture InceptionResNetV2 finale.	41
3.9	Nos procédures de projet.	42
4.1	Code d'étape 1 pour le prétraitement de l'ensemble de données.	48
4.2	Code d'étape 2 pour le prétraitement de l'ensemble de données	48
4.3	Code d'étape 3 pour le prétraitement de l'ensemble de données.	49
4.4	Code d'étape 4 pour le prétraitement de l'ensemble de données.	49
4.5	Code d'étape 5 pour le prétraitement de l'ensemble de données.	50
4.6	Code d'étape 6 pour le prétraitement de l'ensemble de données.	50
4.7	Code d'étape 7 pour le prétraitement de l'ensemble de données.	50
4.8	Code d'étape 8 pour le prétraitement de l'ensemble de données.	51
4.9	Code d'étape 9 pour le prétraitement de l'ensemble de données.	51
4.10	Accuracy du modèle de base	55
4.11	loss du modèle de base.	55
4.12	Accuracy du modèle Xception.	56
4.13	loss du modèle Xception.	56
4.14	Accuracy du modèle InceptionResNetV2.	57
4.15	loss du modèle InceptionResNetV2.	57
4.16	La matrice de confusion du modèle de base.	58
4.17	La matrice de confusion du modèle Xception	59
4.18	La matrice de confusion du modèle InceptionResNetV2..	60
4.19	Comparaison de tous les modèles.	61
4.20	Modèle InceptionResNetV2 VS Modèle xDNN.	62

Liste des tableaux

3.1	Performances du modèle de classification xDNN.	29
3.2	Performances des modèles (CNN/DNN/SAE).	30
3.3	Comparaison de diverses mesures d'évaluation pour la classe COVID.	30
3.4	Comparaison de diverses mesures d'évaluation pour la classe non COVID.. . . .	31
3.5	Nombre d'images et de patients.	32
3.6	Nombre d'images pour chaque ensemble.	33
4.1	évaluations de métriques « metrics »	47
4.2	Résumé du modèle de base.	52
4.3	Résumé du modèle de Xception.	53
4.4	Résumé du modèle de InceptionResNetV2.	53
4.5	Représentation de l'ensemble de données utilisé.	54
4.6	Accuracy/Loss/Time pour le modèle de base.	55
4.7	Accuracy/Loss/Time pour le modèle Xception.	56
4.8	Accuracy/Loss/Time pour le modèle InceptionResNetV2.	57
4.9	Résultats des tests du modèle de base.	58
4.10	Résultats des tests du modèle Xception	59
4.11	du modèle InceptionResNetV2.	60
4.12	Résultats de tous les modèles.	61
4.13	Modèle InceptionResNetV2 VS Modèle xDNN.	62

Liste des abréviations

- PCR :réaction en chaîne par polymérase.
- ADN : Acide DésoxyriboNucléique.
- ARN : acide ribonucléique.
- SARS : Syndrome Respiratoire Aigu Sévère.
- Ace2 : angiotensin-converting enzyme 2.
- ELISA :enzyme-linked immunosorbent assay.
- CT : Computer Tomography.
- IRM : imagerie par résonance magnétique.
- Ia : L'intelligence artificielle.
- KNN :k-Nearest Neighbors.
- SVM :Support Vecteur Machine.
- DL : Deep Learning.
- CNN : Convolutional Neural Network.
- CONV :La couche de convolution .
- cpu : unité de traitement graphique .
- ResNet :Réseaux résiduels.
- LIDC-IDRI :Lung Image Database Consortium and Image Database Resource Initiative.
- ROI :regions-of-interest.
- OCT : Labeled Optical Coherence Tomography.
- OpenCV :Open Source Computer Vision Library.
- BSD :Berkeley Software Distribution.

Introduction générale

Le nouveau coronavirus (COVID-19) est originaire de chauves-souris et a été transmis à l'homme en décembre 2019 par un animal inconnu à Wuhan, en Chine. L'Organisation mondiale de la santé a déclaré le COVID-19 une urgence de santé publique de portée internationale le 30 janvier 2020, et il a été déclaré pandémie mondiale le 11 mars 2020. Où le nombre total de cas à la suite de cette pandémie a atteint plus de 536 millions en juin 2022, alors que l'estimation du nombre de décès est de plus de 6 millions (selon les statistiques de l'Organisation mondiale de la santé). Jusqu'à présent, des tests sont en cours pour détecter la présence de COVID 19 sur la base de la réaction en chaîne par polymérase de transcription inverse (RT-PCR).

Plusieurs revues ont rapporté que la tomodensitométrie thoracique est capable de détecter une infection au COVID-19. Lorsque le virus COVID-19 est détecté à partir d'une tomodensitométrie du thorax via des opacités vitreuses incomplètes bifocales ou une uniformité avec le septum inter lobulaire et un épaissement vasculaire, principalement dans les zones périphériques des poumons. Le processus est une tâche fastidieuse et stressante, impliquant potentiellement erreur humaine et surtout, cela prend beaucoup de temps.

En raison d'avancées significatives dans les domaines de l'apprentissage en profondeur et de la vision par ordinateur, le domaine de l'imagerie médicale a subi des changements importants ces dernières années, et pendant l'épidémie actuelle de COVID19, l'utilisation réussie d'algorithmes d'apprentissage en profondeur peut être particulièrement efficace pour un dépistage rapide et diagnostic de la maladie, qui est exactement le but de notre étude, de sorte qu'il vise Nous avons travaillé pour construire un modèle d'apprentissage en profondeur qui détecte automatiquement Covid-19 sur la base de scanners thoraciques le plus rapidement possible pour une détection précoce des cas.

Notre mémoire est organisé comme suit :

Chapitre 1 : COVID-19 et imagerie médicale. Nous passerons en revue quelques principes médicaux dans ce chapitre, puis décrivons le Coronavirus, quand et comment tout a commencé, ainsi que l'imagerie médicale et ses nombreuses variétés.

Chapitre 2 : Apprentissage en profondeur. Dans ce chapitre, nous aborderons quelques fondamentaux tels que l'Intelligence Artificielle avant de passer au Machine Learning et ses nombreuses formes. Enfin, nous examinerons de plus près l'apprentissage en profondeur.

Chapitre 3 : Matériels et méthodes. Dans ce chapitre, nous allons approfondir nos connaissances, en mentionnant quelques travaux connexes. Après cela, nous parlerons de l'ensemble des données que nous utiliserons, en mentionnant tous ses détails. Après cela, nous passerons aux méthodes que nous utiliserons, et enfin nous parlerons de nos modèles proposés.

Chapitre 4 : Implémentation et tests. Dans ce chapitre, nous allons implémenter nos modèles proposés et discuter des résultats obtenus.

Enfin, nous terminons ce travail par une conclusion générale.

Covid-19 et imagerie médicale

1.1 Introduction :

Pour déterminer si un patient est infecté par le Covid-19 ou non à l'aide de ses photos médicales, il faut d'abord avoir une compréhension fondamentale de deux éléments clés. Imagerie médicale et Covid-19.

Dans ce chapitre, nous passerons en revue certains éléments médicaux essentiels avant de définir le virus Corona, quand et comment il est apparu pour la première fois, ainsi que certains faits sur l'infection, la transmission et les symptômes. Enfin, nous aborderons l'imagerie médicale et ses différentes formes, ainsi que certaines de ses modalités et applications bénignes.

1.2 Maladies virales :

Nous ne pouvons pas parler de reproduction et de transmission du virus sans nous attaquer à certaines maladies que ces virus peuvent causer. Après une période d'incubation, les symptômes peuvent survenir à la suite de maladies qui affligent le corps humain en raison de la propagation généralisée du virus [1].

Les virus sont responsables de certaines maladies humaines. Le rhume et d'autres formes de grippe en font partie :

- * L'hépatite est une maladie du foie (A, B, C).
- * La fièvre Ebola et la fièvre hanta est deux maladies dont il faut être conscient.

- * le VIH, le virus du sida.
- * COVID-19 (récemment rétablie en décembre 2019)

1.3 COVID-19

1.3.1 Définition

Le SRAS-CoV-2, aussi connu sous le nom de COVID-19, est un virus à ARN appartenant à la famille des coronaviridés qui cause des maladies digestives et respiratoires chez les humains et les animaux. Le virus tire son nom de l'apparence de ses particules virales, qui comprennent des excréments semblables à des couronnes[2] .

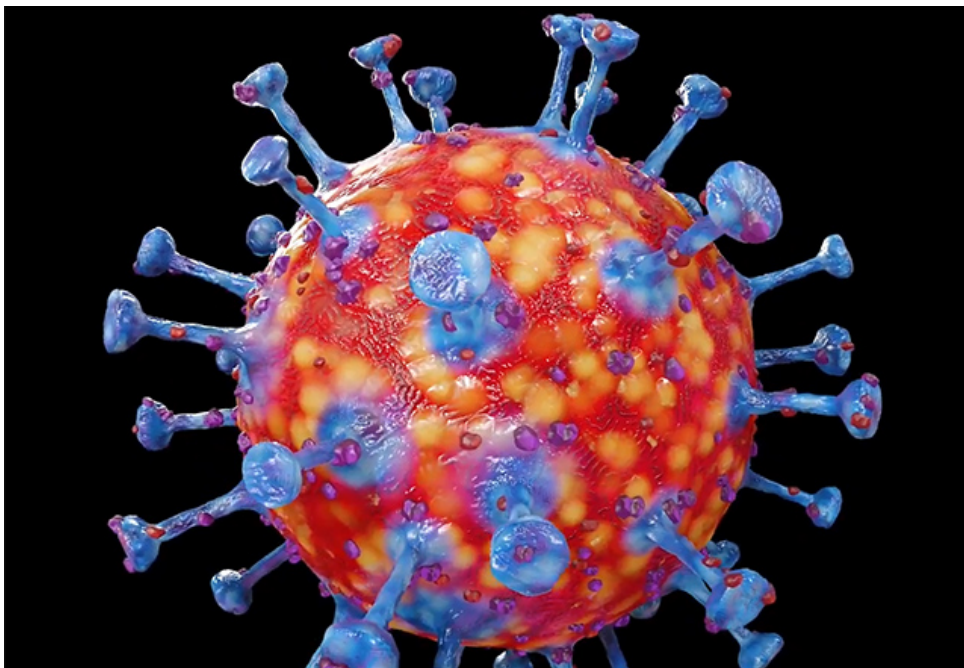


FIGURE 1.1 – gros plan sur la COVID-19

1.3.2 Comment cela a-t-il commencé ?

La maladie semble provenir d'un marché de fruits de mer du Wuhan où certaines créatures sauvages, telles que les chauves-souris et les serpents, sont commercialisées illégalement. Wuhan est une métropole de l'est de la Chine avec une population d'environ 11 millions d'habitants .

Comme on sait que les coronavirus passent des animaux aux humains, il est probable que les premières personnes atteintes de la maladie (un groupe de marchands de fruits de

mer) l'aient contractée par contact avec des animaux .

Bien qu'il y ait de solides prétendants, la source animale de la COVID-19 est encore inconnue. La composition génétique de la COVID-19 est identique à celle d'un coronavirus trouvé chez les chauves-souris, selon une recherche publiée par des virologues de l'Institut de virologie du Wuhan [2].

Le 1er janvier, le marché du poisson du Wuhan a été fermé pour inspection et nettoyage, mais la COVID-19 avait déjà commencé à se propager au-delà du marché. Selon le bureau de l'Ouest du Pacifique de l'Organisation mondiale de la santé, le 21 janvier[2], la maladie se transmettait également entre humains après qu'un médecin eut été infecté par le même virus.

1.3.3 Infection et Symptômes

Infection au niveau de la cellule :

Le virus est un pathogène intracellulaire obligatoire, et doit pénétrer dans une cellule hôte pour pouvoir se multiplier. La première étape de ce processus est donc l'entrée du matériel viral dans le cytoplasme après avoir franchi la membrane cellulaire. L'étape d'entrée débute par l'attachement de la particule virale à la surface de la cellule. Celle-ci repose sur l'interaction entre les spicules à la surface de la particule virale (protéine S du SARS-CoV-2) et la glycoprotéine angiotensine-converting enzyme 2 (ACE2) qui agit en tant que récepteur d'entrée .

Après fixation à ACE2, le spicule virale (S) est coupé en deux parties par une protéase (enzyme qui coupe les protéines) de la cellule hôte. Cet événement moléculaire est nécessaire pour exposer une partie de la séquence polypeptidique de S. S'ensuit un rapprochement entre l'enveloppe du virus et la membrane cellulaire, toutes deux formées par une bicouche lipidique qui fusionneront. Donc la molécule TMPRSS2 qui est présente à la surface de la cellule permet la fusion du virus avec la membrane plasmique de la cellule hôte. Le virus peut également entrer par « endocytose » : la fixation de Spike à ACE2 va induire une invagination de la membrane plasmique, englobant le virus qui rentre dans un « endosome » où une protéase, activée par l'acidité de ce compartiment, permettra de déclencher la fusion entre la membrane endosomale et la membrane virale. La fusion entre les membranes cellulaires et virales libère l'ARN viral dans le cytoplasme cellulaire où se met en place la réplication du virus[3] .

Infection au niveau des personnes et les symptômes :

Après une période d'incubation d'environ cinq jours, 70 des patients infectés développent une toux, de la fièvre ou un essoufflement. Ce stade d'invasion virale est suivi, chez certains patients, d'une réaction immunitaire insuffisante caractérisée par une aggravation des symptômes respiratoires, et un syndrome inflammatoire, généralement huit à dix jours après les premiers symptômes. Dans les cas plus graves, l'infection peut provoquer une pneumonie, un syndrome respiratoire aigu sévère, une insuffisance rénale et même la mort[4] .

1.3.4 Diagnostique :

Parmi les tests diagnostiques qui nous permettent de confirmer le COVID, la réaction de transcription inverse suivie d'une réaction de polymérisation en chaîne quantitative en temps réel (RT-qPCR), et le test de diagnostic rapide basé sur la détection de l'antigène spécifique du SARS-CoV-2 sont deux méthodes utilisées dans la phase précoce des manifestations infectieuses. Les tests de détection des anticorps sériques (ELISA et test de flux latéral) sont utilisés dans la phase ultérieure et après la guérison[5] . Et on a aussi une autre méthode qui se base sur la lecture des images a rayons X, sur la cage thoracique des patients.

1.4 Imagerie médicale

Après avoir vu certains faits de base sur la COVID-19, nous passerons à l'imagerie médicale, à son historique, à ses types et au rôle essentiel qu'elle joue dans le diagnostic de la maladie dans cette partie.

1.4.1 définition

L'imagerie médicale est une technique et un processus pour produire des représentations visuelles des structures internes du corps en utilisant diverses technologies telles que le rayonnement électromagnétique. Ces images sont utilisées pour l'analyse clinique et l'intervention médicale, ainsi que pour la représentation visuelle de la fonction de certains organes ou tissus[6] .

Une image médicale est la matérialisation de l'intérieur du corps humain sous forme d'images ou d'échantillons obtenus par biopsie..., etc., qui sont tous créés par les techniques et les processus d'imagerie médicale[6] .

1.4.2 Historique

1895 : W. Röntgen, physicien allemand, découvre les rayons X.

1896 : H. Becquerel, physicien français, découvre que l'uranium émet des rayonnements invisibles (différents des rayons X). Il donne le nom de « radioactivité » à ce phénomène.

1898 : P. et M. Curie, physiciens français, isolent le polonium et le radium, deux éléments radioactifs jusqu'alors inconnus, présents dans le minerai d'uranium. Le prix Nobel de physique leur est décerné, ainsi qu'à Henri Becquerel, en 1903.

1913 : G. C. de Hevesy, chercheur hongrois, utilise le radium, isotope radioactif naturel, pour en étudier sa distribution dans le corps d'un mammifère, il s'agit du premier traceur.

1913 : H. Geiger et W. Müller créent le premier compteur de particules chargées.

1930 : E. O. Lawrence, physicien américain, met au point un accélérateur électromagnétique de haute fréquence, procédé qui sera ensuite repris pour le cyclotron.

1937 : J.-G. Hamilton effectue la première utilisation clinique du sodium radioactif.

1938 : S. Hertz utilise de l'iode radioactif (^{131}I) pour l'étude de la physiologie thyroïdienne.

1939 : G. C. de Hevesy met au point une méthode de détermination du volume sanguin utilisant des globules rouges marqués au phosphore (^{32}P). J.H. Lawrence utilise ce procédé pour étudier les leucémies.

1942 : J.-G. Hamilton effectue les premières applications thérapeutiques de l' ^{131}I et du ^{32}P .

1950 : Le premier scanner manuel apparaît, doté d'un compteur Geiger-Müller, puis d'un compteur à scintillations. Cette technique sera améliorée l'année suivante par B. Cassen, qui crée un scanner rectilinéaire.

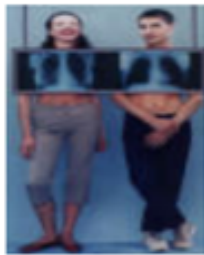
1957 : H.O. Anger invente une caméra à scintillations, gamma-caméra capable de produire des images en corrélation avec le fonctionnement des organes, c'est la scintigraphie.

1963 : Godfrey Newbold Hounsfield, ingénieur britannique, met au point des algorithmes mathématiques, qui sont appliqués ensuite par Allan M. Cormack pour les scanners.

1973 : Première image d'IRM [7].

1.4.3 Types

Il existe plusieurs types de procédures d'imagerie médicale, dont chacune utilise une technique distincte pour générer des images pour diverses raisons. Nous allons passer en revue les techniques d'imagerie les plus répandues, ainsi que celles que nous avons abordées dans l'histoire de l'imagerie médicale.



**Radiographie
(Rayons X)**



**CT-Scan
(Tomographie assistée par ordinateur)**



Echographie



**Imagerie par résonance
magnétique (IRM)**

FIGURE 1.2 – Techniques d'imagerie médicale.

La radiographie (Rayons x) :

L'une des formes les plus répandues d'acquisitions est les rayons X. Selon l'Institut national d'imagerie biomédicale et de bio ingénierie, une radiographie est un procédé typique utilisé pour obtenir des photographies de l'intérieur du corps. Ils sont utilisés pour créer des images d'os, principalement pour détecter si et où ils sont brisés (Figure 1.4) ; les rayons X peuvent également révéler des cancers des os. Ils peuvent être utilisés pour aider les chirurgiens pendant la chirurgie. Son fonctionnement est décrit dans l'histoire de l'imagerie médicale, et voici quelques-uns de ses avantages et inconvénients [8].



FIGURE 1.3 – Radiographie d’un bras cassé.

Avantages :

- * Parce que la machine n’enferme pas complètement le corps, elle ne provoque pas d’anxiété chez les personnes souffrant de claustrophobie, une sorte d’anxiété dans laquelle une crainte déraisonnable d’être piégée ou enfermée peut déclencher une crise de panique.
- * Il suffit de quelques minutes pour effectuer l’opération.

Inconvénients :

- * Les rayons X exposent le patient à un rayonnement indésirable, qui peut causer le cancer, mais la quantité de rayonnement émis est modeste.

Echography (Ultrasons)

Une échographie est un test médical qui utilise des ondes sonores à haute fréquence pour capturer des images en direct de l’intérieur de votre corps.

La technologie est similaire à celle utilisée par le sonar et le radar, qui aident les militaires à détecter les avions et les navires. Une échographie permet à votre médecin de voir les problèmes avec les organes, les vaisseaux et les tissus sans avoir besoin de faire une incision.

Contrairement à d’autres techniques d’imagerie, l’échographie n’utilise aucun rayonnement. Pour cette raison, c’est la méthode préférée pour visualiser un fœtus en développement pendant la grossesse[9].

Avantages :

- * Les échographies n'ont souvent aucun effet secondaire. Cela implique que les activités régulières peuvent reprendre immédiatement ; en outre, les résultats sont affichés en temps réel, éliminant le besoin d'attendre.

Inconvénients :

- * Le latex est présent dans certains capots de sonde, ce qui peut être problématique si le patient est sensible au latex.
- * Les ultrasons endoscopiques peuvent causer des douleurs à la gorge, des ballonnements ou même des saignements internes dans des circonstances graves.



FIGURE 1.4 – Image échographique d'un bébé à naître..

Tomographie par ordinateur (CT : Computer Tomography)

Un scanner génère des images détaillées et de haute qualité du corps. Il s'agit d'une radiographie plus puissante et sophistiquée qui prend une image à 360 degrés de la colonne vertébrale, des vertèbres et des organes internes. Un colorant de contraste peut être injecté dans votre sang afin que le médecin puisse voir plus clairement les structures de votre corps sur le scanner [10].

Une tomodensitométrie crée des images détaillées et de qualité des os, des vaisseaux sanguins, des tissus mous et des organes et peut être utilisée pour aider le médecin à diagnostiquer des conditions médicales telles que :

- Appendicite.
- Cancer.
- Traumatisme.
- Troubles musculo-squelettiques.
- Cardiopathie.
- Maladies infectieuses.

Avantages :

- * Les scanners CT sont rapides, ne prennent que 10 à 20 minutes et donnent des résultats immédiats comparativement à d'autres types d'examens.
- * Les scanners CT sont indolores puisqu'ils ne sont pas invasifs.

Inconvénients :

- * Exposition au rayonnement. Plus le corps d'un patient est examiné, plus il est exposé au rayonnement.

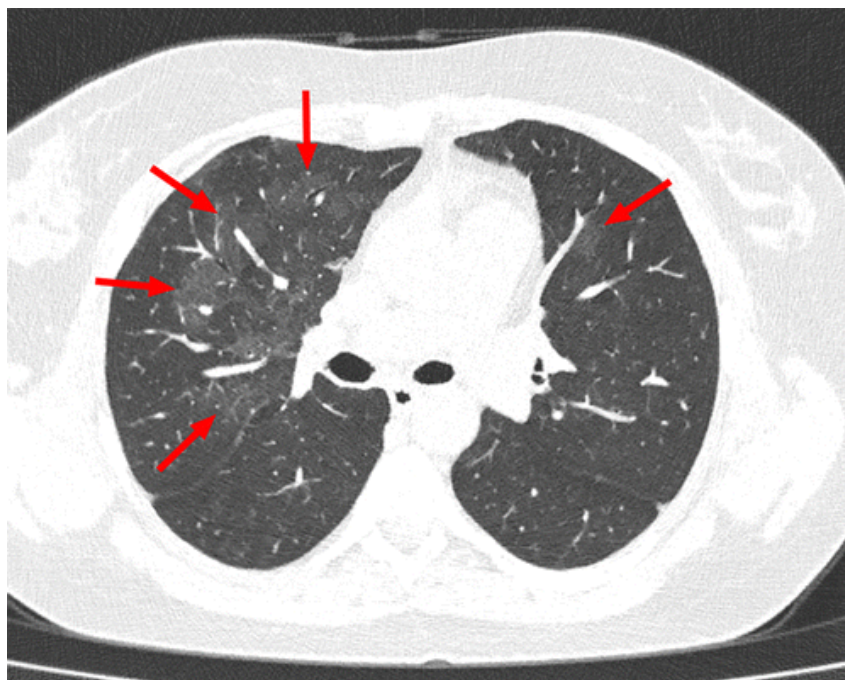


FIGURE 1.5 – Exemple de scanners CT thoracique.

Imagerie par résonance magnétique (IRM)

IRM signifie imagerie par résonance magnétique. Elle utilise un champ magnétique (aimant) et des ondes radio. Aucune radiation ionisante n'est émise.

Son principe consiste à réaliser des images du corps humain grâce aux nombreux atomes d'hydrogène qu'il contient. Placés dans un puissant champ magnétique, tous les atomes d'hydrogène s'orientent dans la même direction : ils sont alors excités par des ondes radio durant une très courte période (ils sont mis en résonance). A l'arrêt de cette stimulation, les atomes restituent l'énergie accumulée en produisant un signal qui est enregistré et traité sous forme d'image par un système informatique[11] .

L'IRM étudie avec une grande précision de nombreux organes tels que le cerveau, la colonne vertébrale, les articulations et les tissus mous. Elle est d'une grande utilité lorsqu'une analyse très fine est nécessaire et que certaines lésions ne sont pas visibles sur les radiographies standards, l'échographie ou le scanner. Elle permet de faire des images en coupes dans différents plans et de reconstruire en trois dimension la structure analysée [11].

Avantages :

- * Les examens par IRM sont non invasifs et indolores, et les champs magnétiques et les ondes radio n'ont aucun effet nocif sur le corps humain.
- * Comme ils n'utilisent pas de radiographies, les femmes enceintes et les nouveau-nés peuvent s'en servir de façon non invasive au besoin.

Inconvénients :

- * Étant donné que les IRM contiennent trop de corps, les personnes souffrant de claustrophobie peuvent se sentir mal à l'aise.
- * Comme le métal ne peut pas se déplacer à l'intérieur d'un appareil d'IRM, les personnes qui ont des implants métalliques, comme les stimulateurs cardiaques, ne peuvent pas les utiliser.

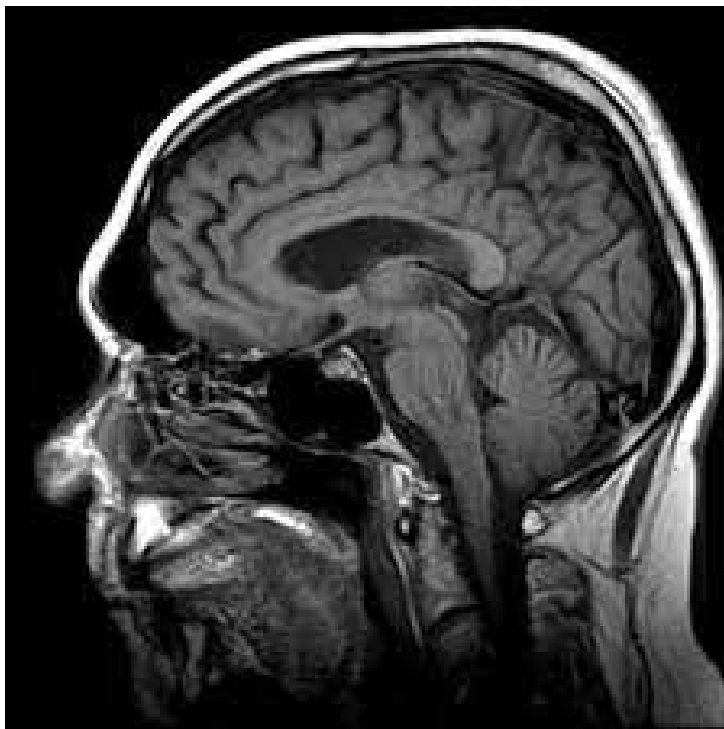


FIGURE 1.6 – Exemple d'une IRM.

1.5 Comment détecter la COVID-19 dans les TDM thoracique "CT Scans"

Selon la littérature, les opacités multifocales bilatérales irrégulières du verre au sol (OGG) ou la consolidation avec épaissement septique et vasculaire inter lobulaire, principalement dans les régions périphériques des poumons, sont des constatations caractéristiques de l'imagerie TDM thoracique des personnes atteintes de la COVID-19. Les opacités irrégulières et circulaires sont les morphologies les plus répandues, suivies par les opacités triangulaires et linéaires. À mesure que la maladie progresse, les résultats de la TDM peuvent varier [12].

Comme le montre la Figure 1.7 – A , la TDM thoracique précoce indique de minuscules OGG lobulaires et sous-segmentaires, des altérations interstitielles et un épaissement des lumens vasculaires partout.

Les lésions finissent par atteindre de nombreuses ICG dans les poumons au cours de la phase progressive (après 3 jours), et certaines personnes peuvent avoir une consolidation épaisse dans les lobes, comme le montre la Figure 1.7 – B [12].

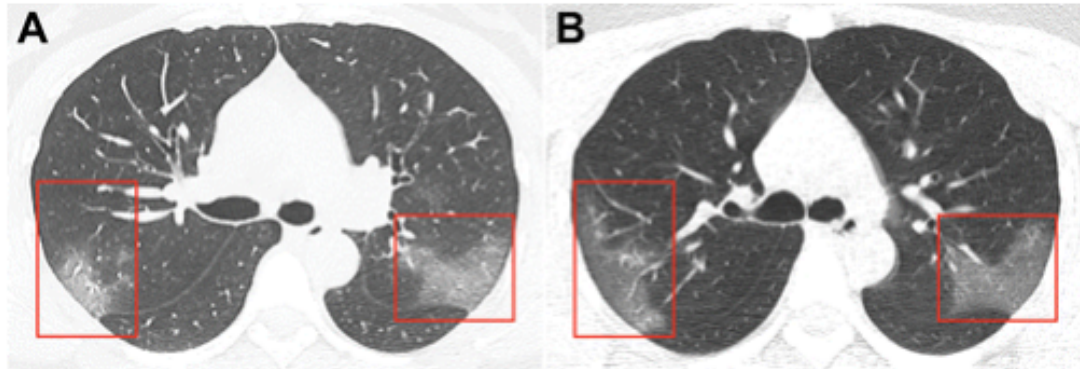


FIGURE 1.7 – Images CT montrant le développement du coronavirus dans les 3 jours.

1.6 Conclusion

Ce chapitre a été consacré à la présentation de concepts de base comme les virus, suivie d'un examen approfondi du nouveau coronavirus. Enfin, nous avons passé en revue un aspect important de la médecine connu sous le nom d'imagerie médicale, dans lequel nous avons expliqué ce qu'est l'imagerie médicale, quand et où elle a commencé, certains de ses types, et son rôle critique dans le monde des soins de santé, en particulier dans la prévention, le diagnostic et le traitement de maladies.

Dans le prochain chapitre, nous allons présenter Machine Learning plus spécifique, Deep Learning (DL) et ses différentes méthodes d'apprentissage.

Chapitre 2

L'apprentissage profond «Deep Learning»

2.1 Introduction :

Nous devons introduire un élément essentiel, l'apprentissage profond, pour construire un algorithme qui peut automatiser l'identification de la COVID-19 à l'aide de l'imagerie médicale.

Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond, tels que les méthodes d'intelligence artificielle, l'apprentissage automatique et ses algorithmes. Enfin, nous expliquerons Deep Learning, en donnant quelques définitions, et nous détaillerons également ses différents types.

2.2 Intelligence artificielle

Le but de l'intelligence artificielle (IA) est d'imiter la façon dont le cerveau humain traite l'information. Afin de permettre aux robots de simuler un type d'intelligence réelle, l'intelligence artificielle implique l'adoption d'un certain nombre de technologies. L'apprentissage automatique et l'apprentissage en profondeur sont deux des sous-domaines les plus connus de l'IA [13].

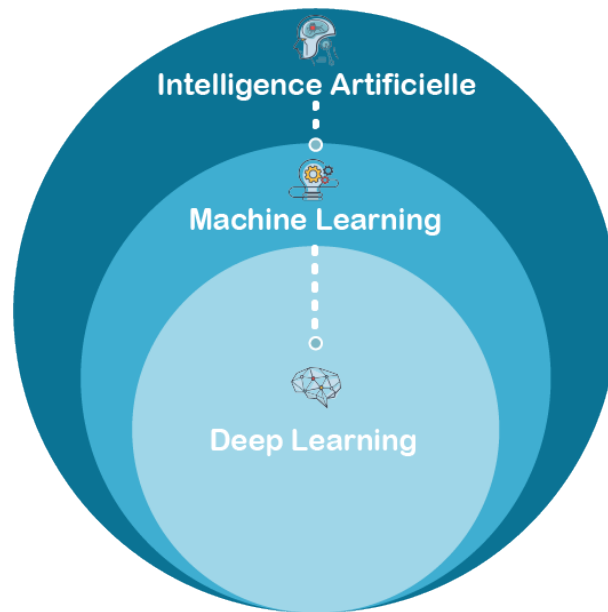


FIGURE 2.1 – L'IA.

2.3 Machine Learning «L'apprentissage automatique»

Les formes vivantes supérieures doivent avoir la capacité de s'adapter et d'apprendre de leurs expériences antérieures. Dans les premiers stades de la vie, il est crucial pour les humains de maîtriser des compétences fondamentales telles que la reconnaissance vocale, la reconnaissance faciale, la compréhension auditive, l'apprentissage de la marche et de la parole [14].

L'objectif de l'apprentissage automatique est de comprendre et d'imiter cette capacité d'apprentissage dans les systèmes informatisés. Très grossièrement, il s'agit de créer des algorithmes capables d'apprendre à partir d'un grand nombre d'exemples (les données équivalentes à "l'expérience passée") et d'appliquer ce qu'ils ont découvert à de nouvelles situations [14].

2.4 Méthodes d'apprentissage automatique

L'apprentissage automatique est divisé en trois types :

2.4.1 Apprentissage supervisé

C'est celui où vous pouvez prétendre que vous êtes enseigné par un instructeur (Train Me!). Nous avons un ensemble de données qui sert d'enseignant, et son travail est de former la machine ou le modèle. Lorsque des données supplémentaires sont présentées au modèle, celui-ci peut commencer à générer des prédictions ou des décisions (Figure 2.2) [15].

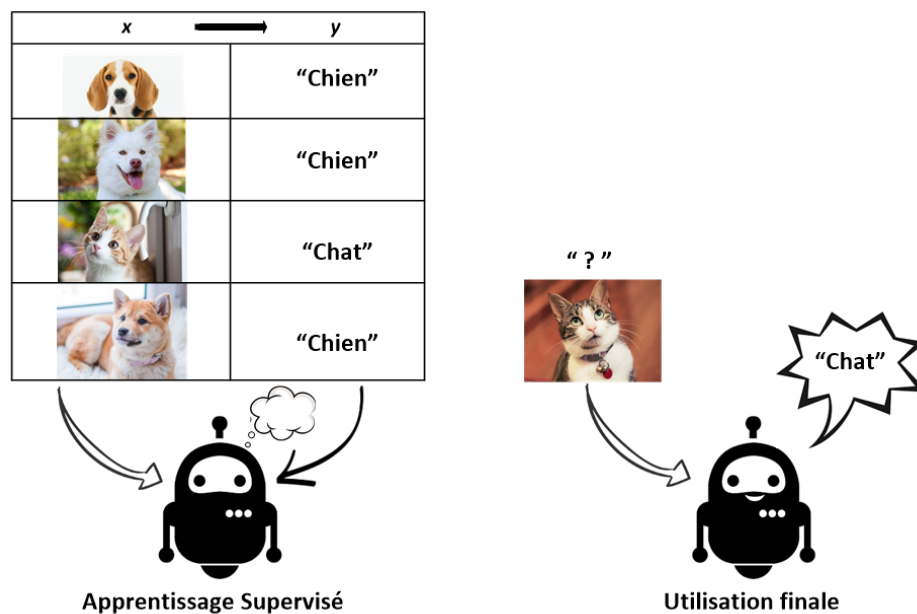


FIGURE 2.2 – exemple d'apprentissage supervisé

2.4.2 Apprentissage non supervisé :

Le modèle apprend en observant les données et en cherchant des modèles. Lorsque le modèle reçoit un ensemble de données, il crée automatiquement des grappes dans l'ensemble de données pour détecter les tendances et les liens (je suis autonome dans l'apprentissage!). Il ne peut pas ajouter d'étiquettes à la grappe, comme "c'est un groupe de chats ou de chiens", mais il peut séparer tous les chats des chiens .

Si nous fournissons les photos modèles de pommes et de bananes, il construira des grappes et divisera l'ensemble de données en ces groupes en fonction de modèles et de corrélations spécifiques. Si le modèle reçoit de nouvelles données, il les ajoute à l'une des grappes précédemment formées (Figure 2.3) [15].

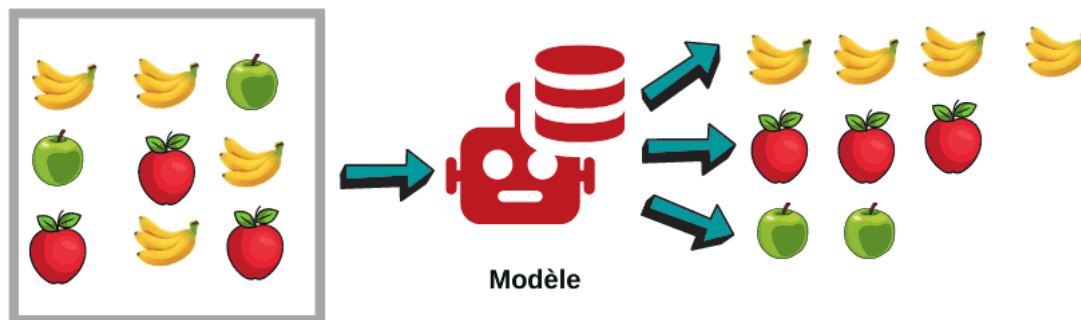


FIGURE 2.3 – exemple d'apprentissage non supervisé

2.4.3 Apprentissage par renforcement :

Il s'agit de la capacité d'un agent d'interagir avec l'environnement et de déterminer le résultat optimal. Il est basé sur la stratégie hit-or-miss (Ma vie, mes règles!).

Pour une réponse correcte ou incorrecte, l'agent est récompensé ou pénalisé par un point, et le modèle s'entraîne lui-même en fonction des points de récompense positifs gagnés. Une fois formé, il est prêt à prévoir de nouvelles données (Figure 2.4)[15].

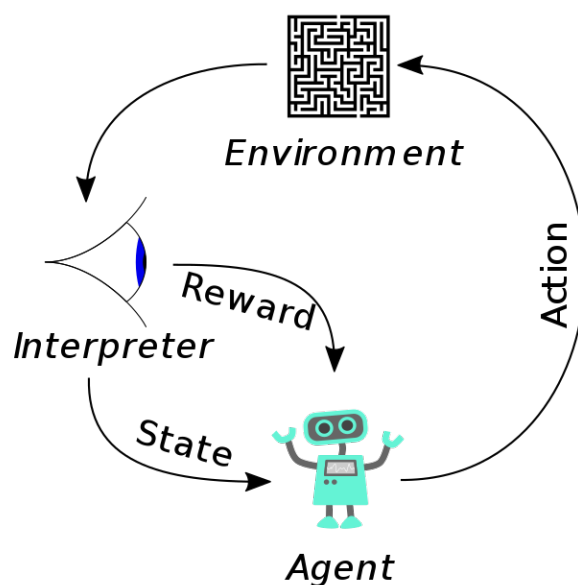


FIGURE 2.4 – exemple d'apprentissage par renforcement

2.5 algorithmes d'apprentissage automatique

2.5.1 KNN (k-Nearest Neighbors)

La reconnaissance de formes, la recherche dans les données multimédias, la compression vectorielle, les statistiques informatiques et l'exploration de données ne sont que quelques-uns des problèmes informatiques qui impliquent fréquemment la technique du k-plus proche voisin (KNN).

La technique KNN diffère essentiellement des autres méthodes par sa simplicité et par le fait qu'aucun modèle n'est introduit à partir des exemples pendant le processus de classification.

Cet algorithme recherche les k voisins les plus proches de la nouvelle instance, puis sélectionne le candidat qui est à la fois le résultat le plus proche et le plus commun parmi eux pour déterminer la classe du nouveau cas. Afin de comparer les cas précédemment classés avec le nouveau cas, cette approche utilise principalement deux paramètres : le nombre k et une fonction de similarité [16].

2.5.2 L'algorithme k-means (K-moyen) :

Une collection de données peut être divisée en k classes homogènes en utilisant l'approche des k-moyennes, qui est considérée comme un outil de classification efficace.

K-means ne construit qu'un seul niveau de clusters, contrairement à d'autres approches ostensiblement hiérarchiques qui construisent une structure "Cluster tree" pour représenter les groupements. Il fournit une partition des données où les éléments de chaque cluster sont aussi séparés que possible les uns des autres et des clusters en dehors de celui-ci. Les objets et le centre de gravité de chaque cluster de la partition servent d'identificateurs. La somme des distances entre chaque objet et le centre de gravité de son cluster est minimisée à l'aide d'un algorithme itératif [17].

2.5.3 SVM (Support Vecteur Machine) :

Vladimir Vapnik a développé l'approche Support Vector Machine, qui est un système d'apprentissage automatique à super visée. Son objectif est de résoudre les problèmes de classification et de régression. Elle dépend essentiellement de l'utilisation de deux notions : la notion de marge maximale et la notion de fonction du noyau (facilite le processus de séparation).

De nombreux problèmes de bioinformatique, de recherche d'informations, de vision par ordinateur, peuvent être résolus à l'aide de la technique SVM [17].

La SVM peut être de deux types [18] :

1. **SVM linéaire** : Il est utilisé pour les données séparables linéairement, ce qui implique que si un ensemble de données peut être catégorisé en deux groupes en utilisant une seule ligne droite, il est appelé données séparables linéairement, et le classificateur utilisé est le classificateur SVM linéaire.
2. **SVM non linéaire** : C'est pour les données séparées non linéairement, ce qui implique que si un ensemble de données ne peut pas être catégorisé en ligne droite, ce sont des données non linéaires, et le classificateur employé est un classificateur SVM non linéaire.

2.6 L'apprentissage profond «Deep Learning»

2.6.1 Définition

L'intelligence artificielle s'appuie fortement sur l'apprentissage en profondeur, souvent appelé "Deep Learning", un groupe de techniques d'apprentissage autonomes. Un programme peut désormais faire des inférences sur de nouvelles données grâce aux nombreuses avancées significatives rendues possibles par cette méthode. En fait, l'apprentissage profond repose sur un réseau de neurones artificiels constitué de milliers d'unités, dont chacune reçoit et interprète les données de la couche inférieure. Par exemple, en reconnaissance visuelle, les couches inférieures d'unités identifient des lignes, des courbes, des angles, etc., tandis que les couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes, etc.

Du fait de l'émergence des ordinateurs et des bases de données massives, ces stratégies ont considérablement facilité les avancées. Cette idée a été présentée pour déplacer l'apprentissage automatique vers son objectif principal d'intelligence artificielle. De plus, il couvre les algorithmes d'apprentissage automatique qui visent à modéliser des relations compliquées entre des données en apprenant à plusieurs niveaux de représentation qui correspondent à différents niveaux d'abstraction. En raison des plusieurs couches de traitement, qui sont constituées de différentes transformations linéaires et non linéaires, il a également une bonne capacité à extraire des caractéristiques à partir de données brutes [19, 20].

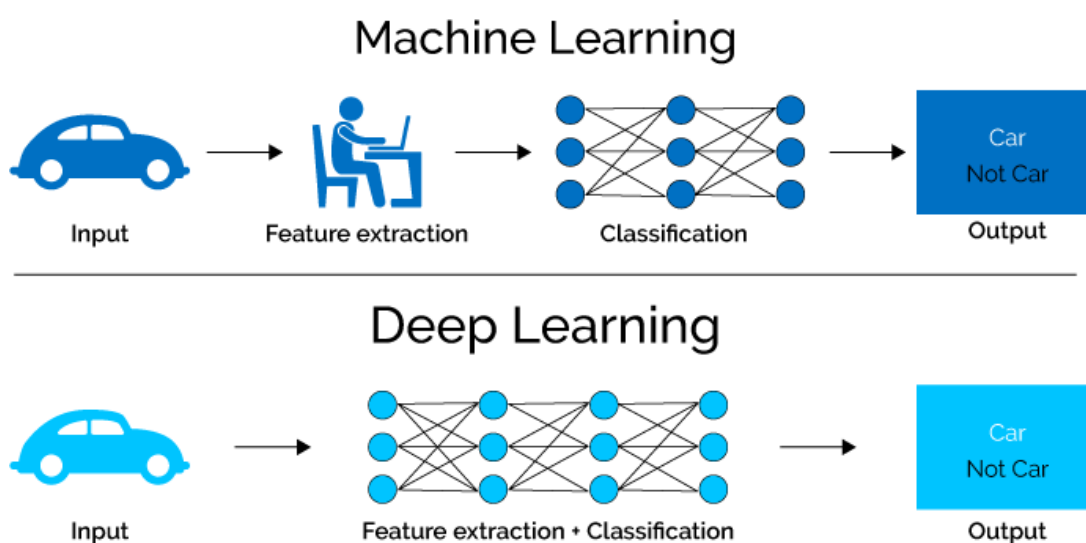


FIGURE 2.5 – Machine Learning vers Deep Learning.

2.6.2 Domaines d'application de l'apprentissage profonde :

La technologie d'apprentissage en profondeur est l'une des techniques les plus utilisées dans de nombreux domaines, notamment :

- **La reconnaissance faciale :**

Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.

- **Le traitement automatique de langage naturel :**

Le traitement automatique de langage naturel est une autre application du DL. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse

de sentiments.

— **Voitures autonomes :**

Les entreprises qui construisent de tels types de services d'aide à la conduite, doivent apprendre à un ordinateur à maîtriser certaines parties essentielles de la conduite à l'aide de systèmes de capteurs numériques au lieu de l'esprit humain. Pour ce faire, les entreprises commencent généralement par entraîner des algorithmes utilisant une grande quantité de données.

— **Traduction automatique :**

Il s'agit d'une tâche dans laquelle des mots, expressions ou phrases donnés dans une langue sont automatiquement traduits dans une autre langue. La traduction automatique existe depuis longtemps, mais DL permet d'obtenir les meilleurs résultats dans deux domaines spécifiques :

1. Traduction automatique de texte.
2. Traduction automatique d'images.

— **Recherche vocale et assistants à commande vocale :**

L'un des domaines d'utilisation les plus populaires de DL est la recherche vocale et les assistants intelligents à commande vocale. Avec les grands géants de la technologie ont déjà fait d'importants investissements dans ce domaine.

— **Recherche en marketing :**

En plus de rechercher de nouvelles fonctionnalités susceptibles d'améliorer votre application, DL peut également être utile en arrière-plan. La segmentation du marché, l'analyse des campagnes marketing et bien d'autres peuvent être améliorés à l'aide de modèles de régression et de classification DL [21].

2.7 Réseaux de neurones profonds :

Les réseaux de neurones profonds sont l'une des méthodes d'apprentissage en profondeur les plus importantes pour créer des modèles de vision par ordinateur capables de simuler la vision humaine. Ce qui le distingue, c'est qu'il contient plus d'une couche cachée (multicouche)[22].

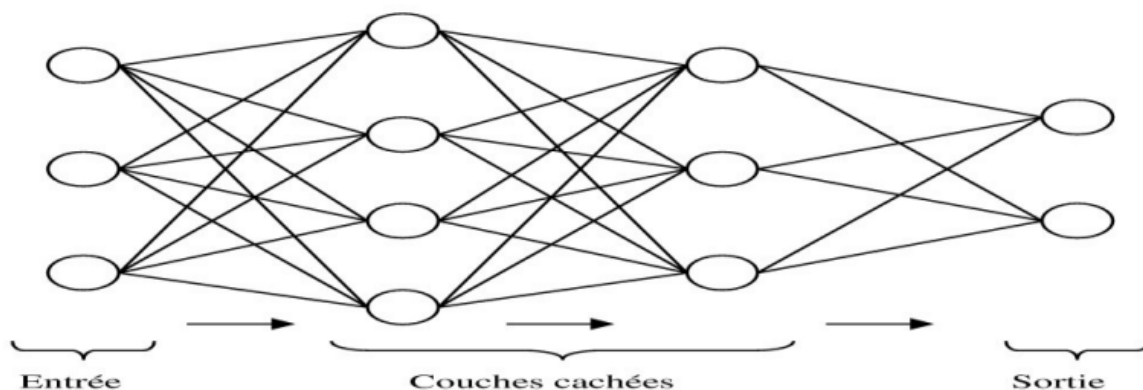


FIGURE 2.6 – Schéma d'un réseau de neurone profond.

Les réseaux de neurones profonds contiennent de nombreux types, dont les plus importants sont :

2.7.1 Les réseaux de neurones convolutifs :

Les progrès de l'apprentissage en profondeur sont l'une des raisons du développement rapide de la vision par ordinateur. Lors de la discussion sur la vision par ordinateur, le terme réseau de neurones convolutifs CNN est utilisé. Il est divisé en deux composants uniques :

La composante convolutive elle-même constitue la première section d'un CNN. Il sert d'extracteur de caractéristiques visuelles. Les cartes de convolution sont produites en faisant passer une image à travers une série de filtres, ou noyaux de convolution. Certains filtres intermédiaires utilisent une opération de maximum local pour réduire la résolution de l'image. Les cartes de convolution sont finalement aplaties et concaténées pour créer un vecteur de caractéristiques connu sous le nom de code CNN.

L'entrée d'un second composant composé de couches entièrement connectées est couplée au code CNN à la sortie de la partie convolutive. La fonction de cette partie est de combiner les propriétés du code CNN afin de classer l'image. Un neurone par type constitue la dernière couche produite. une valeur de distribution de probabilité à travers les catégories [22].

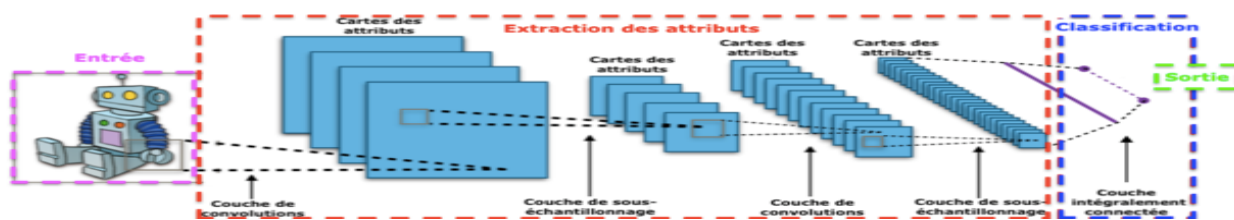


FIGURE 2.7 – Les réseaux de neurones convolutifs.

Il y a plusieurs couches différentes dans CNN comme le montre la figure 2.9 :

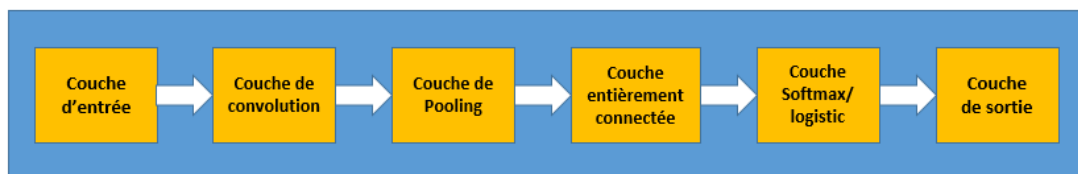


FIGURE 2.8 – Les couches de CNN

— La couche de convolution (CONV) :

Puisque les propriétés de l'image sont extraites dans cette couche, c'est la couche la plus cruciale et le centre des éléments constitutifs du réseau convolutif. C'est aussi la couche qui demande le plus de travail de calcul.

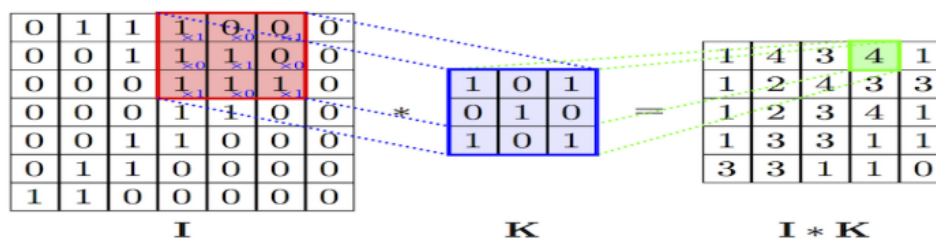


FIGURE 2.9 – Exemple de principe du filtre convolutif

— La couche de Pooling :

est utilisé pour diminuer le volume spatial de l'image d'entrée après convolution. Son but est de réduire progressivement la taille spatiale de la représentation afin de diminuer le nombre de paramètres et de calculs dans le réseau et, par conséquent, de contrôler le overfitting.

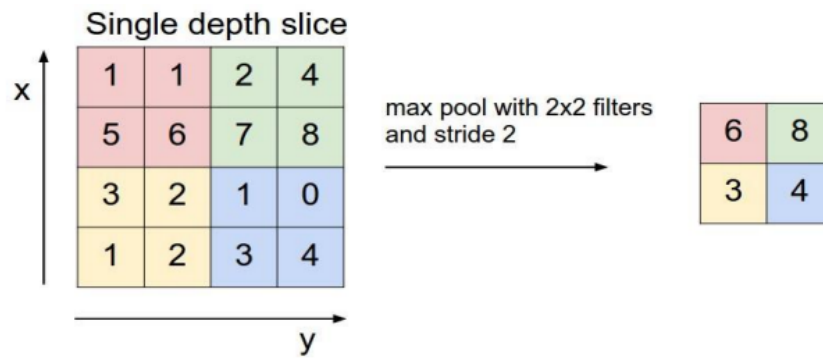


FIGURE 2.10 – Exemple de principe du Pooling

— **Couche entièrement connectée(FC) :**

Après plusieurs couches de convolution et de max-pooling le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées[22] . Il connecte les neurones d'une couche aux neurones d'une autre couche. Il est utilisé pour classer les images entre différentes catégories par formation.

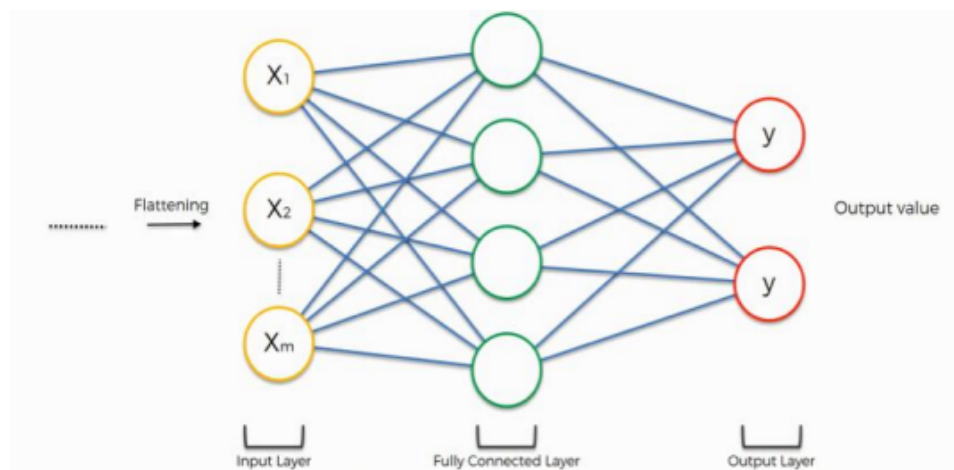


FIGURE 2.11 – Principe de la couche entièrement connectée (FC)

— **Couche Logistique ou Softmax :**

Est la fin de la couche entièrement connectée .La logistique est utilisée pour la classification binaire et Softmax est pour la multi-classification.

— Couche de sortie (output layer) :

La couche de sortie contient l'étiquette qui est sous forme codée comme le montre la figure 2.12.

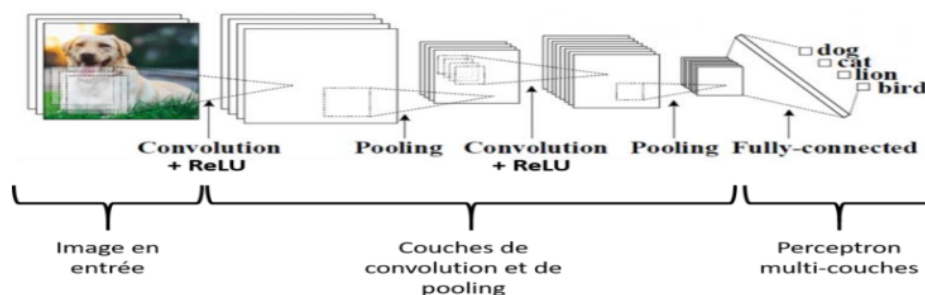


FIGURE 2.12 – Exemple montrant l'étiquette codée de la couche de sortie CNN

2.7.2 Réseau de neurones récurrents :

Les réseaux de neurones ne sont pas un modèle nouveau. En effet ils ont été introduits dès les années 50 – le perceptron est présenté en 1958,[23] . Et connaissent actuellement leur heure de gloire grâce aux architectures profondes – Deep learning – et à la puissance de calcul proposée par les GPU.

L'idée derrière les RNN est d'utiliser des informations séquentielles. Si on veut prédire le prochain mots qui dans une phrase, il faut connaître les mots venus avant . les RNN est qu'ils ont une «mémoire» qui capture des informations sur ce qui a été calculé jusqu'à présent.

En théorie, les RNNs peuvent utiliser les informations en séquence, Ils sont arbitrairement longs, mais en pratique ils se limitent à ne regarder que quelques pas en arrière. Voici à quoi ressemble un RNNs typique :

Le schéma ci-dessus montre un RNN déroulé. En déroulant, nous signifions simplement qu'on montre le réseau pour la séquence complète. Par exemple, si la séquence qui nous intéresse est une phrase de 5 mots, le réseau serait déroulé en un réseau de neurones de 5 couches, une couche pour chaque mot.

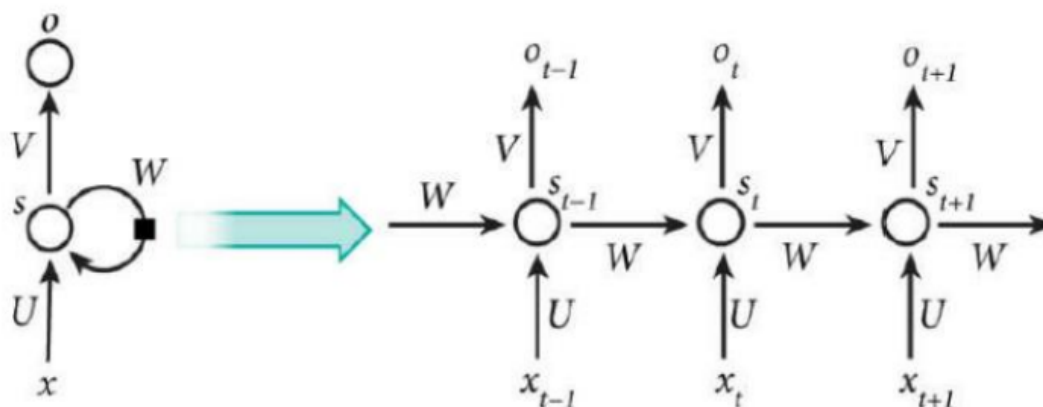


FIGURE 2.13 – (à gauche) Un RNN. (à droite) Sa version déroulée Source : Nature[15]

Les formules qui régissent les calculs dans un RNN sont les suivantes :

- x_t est l'entrée au moment t .
- $U; V; W$ sont les paramètres que le réseau va apprendre des données de l'apprentissage.
- s_t est l'état caché au moment t . C'est la « mémoire » du réseau. s_t est calculé en fonction de l'état

Caché précédent et de l'entrée à l'étape actuelle :

$$\mathbf{S}_t = \mathbf{F}(\mathbf{U}_{x_t} + \mathbf{W}_{s_{t-1}})$$

2.8 Conclusion

Nous avons couvert l'apprentissage automatique et ses nombreuses sortes dans ce chapitre. Nous avons spécifiquement discuté de l'apprentissage profond et de la distinction entre DL et ML. En outre, nous avons fourni divers domaines d'application DL. Enfin, nous avons discuté des différentes formes de réseaux neuronaux, notamment CNN, sur lesquels notre solution est basée.

Dans le but de sélectionner les meilleures approches pour notre projet. au prochain chapitre Nous discuterons de certains travaux connexes comme exemples de la façon dont le DL est utilisé avec l'imagerie médicale pour l'identification de la maladie, et en fin nous allons présenter le jeu de données que nous allons utiliser. Ensuite, nous choisirons et étudierons la méthode de classification qui correspond à notre problème, et proposerons quelques modèles afin d'arriver à choisir le meilleur modèle en fonction de ses résultats.

Matériels et méthodes.

3.1 Introduction :

Les avancements récentes dans le domaine de la DL, en particulier dans le domaine de l'imagerie médicale, indiquent l'utilisation potentielle de diverses architectures Deep CNN. Dans notre solution proposée, nous utiliserons les architectures CNN avec un ensemble de données de 2481 images de 120 patients entre hommes et femmes. Toutes les données passeront par notre modèles, où CNN jouera un rôle majeur dans la classification de nos données en deux classes : X et Y, qui définissent simplement si un patient est infecté par le coronavirus ou non.

Dans ce chapitre, nous verrons les algorithmes DL utilisés dans le traitement des images médical, sous le titre de travaux connexe. Ensuite, nous allons passer en revue l'ensemble de données que nous utiliserons, et ainsi de suite. Ensuite, où nous discuterons de nos modèles proposés et de certains de leurs fondements (fonctions d'activation, fonctions de perte, optimiseurs).

3.2 Travaux connexes

Les dernières améliorations apportées aux modèles d'apprentissage profond et la disponibilité d'énormes ensembles de données ont aidé les algorithmes à surpasser le personnel médical dans de nombreuses tâches d'imagerie médicale. Et parmi les projets existant dans ce domaine, nous mentionnons :

3.2.1 Utilisation Un grand ensemble de données CT-scan des patients pour l'identification du SRAS-CoV-2

Le 12 May 2020, Eduardo Soares, Plamen Angelov, Sarah Biaso , Michele Higa Froes , and Daniel Kanda Abe ont soumis un article sur “SARS-CoV-2 CT-scan dataset : A large dataset of real patients CT scans for SARS-CoV-2 identification [24].

Dans cette travail, Une approche de classification eXplainable Deep Learning « xDNN » est proposé pour l'identification du SRAS-CoV-2 .

L'ensemble de données utilisé dans ce travail est composé de 2482 images de CT scan, qui se répartissent entre 1252 pour les patients infectés par le Covid-19, et 1230 CT scan pour les patients non infectés par le Covid-19 Tous les résultats obtenus sont illustrés dans les tableaux ci-dessous.

Modèle	xDNN
accuracy	97.36
Recall	97.2
Specificity	95.52
Precision	91.6
F1-Score	97.31

TABLE 3.1 – Performances du modèle de classification xDNN.

3.2.2 Système automatisé d'identification des infections au COVID-19 dans les images de tomodensitométrie à l'aide de modèles d'apprentissage en profondeur

Le 30 Mar 2022, Karrar Hameed Abdulkareem, Salama A. Mostafa, Zainab N. Al-Qudsy, Mazin Abed Mohammed, Alaa S. Al-Waisy, Seifedine Kadry, Jinseok Lee, and Yunyoung Nam ont proposé trois modèles d'apprentissage en profondeur : CNN, DNN et SAE pour l'identification du Covid-19 [25].

L'ensemble de données utilisé dans ce travail se compose de 746 cas de patients COVID-19 avec leurs images de CT scan (349 images avec une infection COVID-19 confirmée/et

397 à des personnes en bon état de santé).

Tous les résultats obtenus sont illustrés dans les tableau ci-dessous.

Modèle	accuracy(%)	Sensitivity(%)	Specificity (%)
CNN	88.30	87.65	87.97
DNN	86.23	84.41	86.77
SAE	86.75	85.62	87.84

TABLE 3.2 – Performances des modèles (CNN/DNN/SAE).

3.2.3 Identification de la COVID-19 à partir d'images CT thoraciques une approche fondée sur les réseaux neuronaux convolutionnels profonds

Le 01 févr 2022, S.V. Kogilavani, J. Prabhu, R. Sandhiya, M. Sandeep Kumar, UmaShankar Subramaniam, Alagar Karthick, M. Muhibbullah et Sharmila Banu Sheik Imam ont soumis un article sur l'identification du covid-19 à partir d'images de "CT-scan = TDM" thoracique, ils ont utilisé diverses approches profondes basées sur CNN et ont proposé une approche basée sur Architectures CNN telles que VGG16, DenseNet121, MobileNet, NASNet, Xception et EfficientNet[26].

L'ensemble de données a été extrait de Kaggle et se composait de 3873 images. COVID et non-COVID sont les deux catégories dans lesquelles les données sont divisées.

Tous les résultats obtenus sont illustrés dans les tableaux ci-dessous.

Performance mmeasure	VGG16	DenseNet121	MobileNet	Xception	NASNet	EfficientNet
Precision	1.00	0.96	0.99	0.90	0.96	0.91
Rocal	0.96	0.99	0.94	0.96	0.83	0.46
F1-score	0.98	0.98	0.96	0.93	0.89	0.61
Support	357	357	357	357	357	315

TABLE 3.3 – Comparaison de diverses mesures d'évaluation pour la classe COVID.

Performance mmeasure	VGG16	DenseNet121	MobileNet	Xception	NASNet	EfficientNet
Precision	0.96	0.99	0.94	0.95	0.84	0.63
Rocal	1	0.96	0.99	0.89	0.96	0.95
F1-score	0.98	0.97	0.96	0.92	0.90	0.76
Support	334	334	334	334	334	307

TABLE 3.4 – Comparaison de diverses mesures d'évaluation pour la classe non COVID..

3.3 Ensemble de données « Dataset » utilisé

L'ensemble de données proposé est composé de 2482 images de CT scan, qui se répartissent entre 1252 pour les patients infectés par le Covid-19, et 1230 CT scan pour les patients non infectés par le Covid-19, mais qui présentaient d'autres maladies pulmonaires. Les données ont été recueillies auprès des hôpitaux de Sao Paulo, au Brésil. Le nombre détaillé de patients est illustré par la figure (3.3). Les caractéristiques détaillées de chaque patient ont été omises par les hôpitaux pour des raisons éthiques. La figure (3.4) illustre quelques exemples de CT scan pour des patients infectés et non infectés par le Covid-19 qui composent l'ensemble de données[27] .

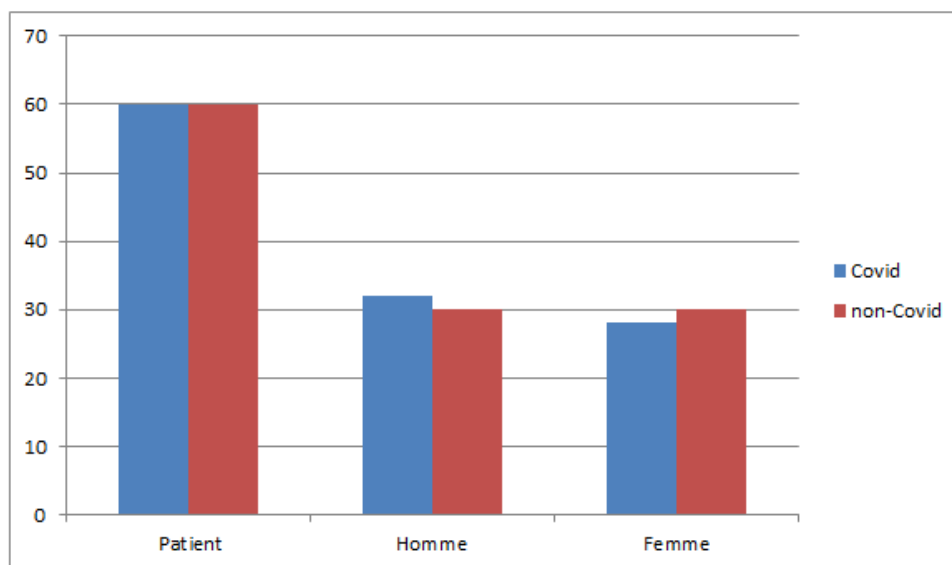


FIGURE 3.1 – La distribution de notre ensemble de données.

Description de la figure 3.3 : La figure illustre le nombre de patients considérés pour composer l'ensemble de données. Dans ce cas, nous avons considéré les données de 60 patients infectés par le Covid-19, dont 32 hommes et 28 femmes. Nous avons également pris en compte les données de 60 patients non infectés par le Covid-19, dont 30 hommes et 30 femmes.

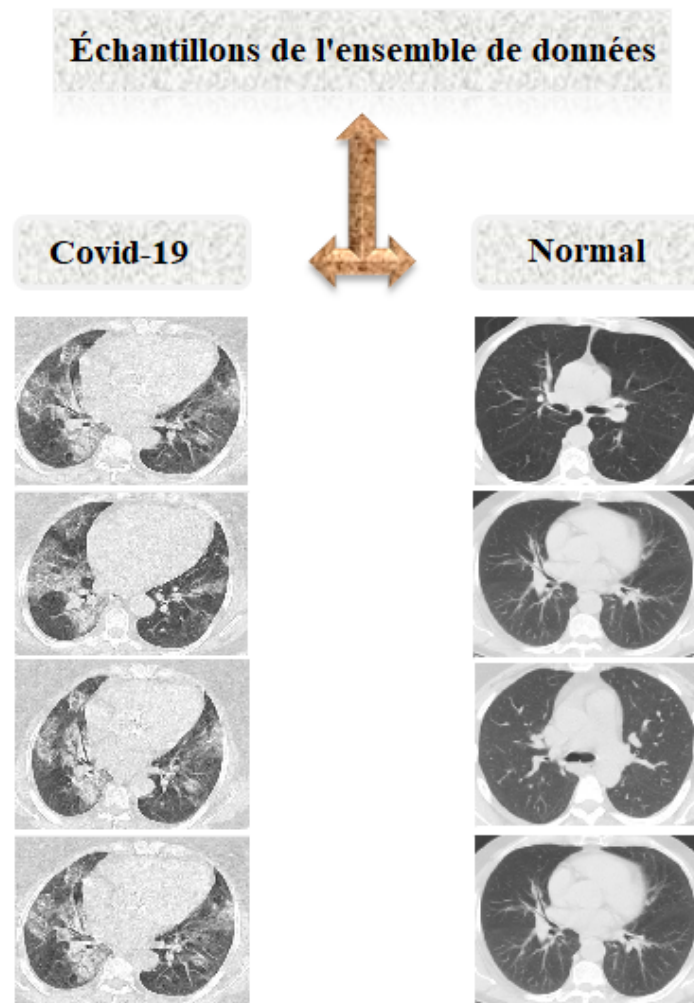


FIGURE 3.2 – Exemples d'images de "CT scan" COVID19 positives/négatives.

Patients COVID-19	Patients non COVID-19	Image COVID-19	Image non COVID-19
60	60	1252	1230

TABLE 3.5 – Nombre d'images et de patients.

Les images seront divisées en trois sous-ensembles importants comme suit :

1. **Ensemble d'entraînement** : 60 % sont utilisés dans la phase d'entraînement.
2. **Ensemble de test** : 20% sont utilisés dans la phase de test.
3. **Ensemble de validation** : 20 % sont utilisés dans la phase de validation.

Dataset	Nombre d'image
Ensemble d'entraînement	1489
Ensemble de test	497
Ensemble de validation	497

TABLE 3.6 – Nombre d'images pour chaque ensemble.

3.4 Fonction d'activation et Optimisateurs et fonction de perte utilisé

3.4.1 Fonction d'activation

C'est une fonction utilisée dans les réseaux neuronaux artificiels qui produit une petite valeur pour les petits intrants, et une plus grande valeur si ses intrants dépassent un seuil. Si les entrées sont assez grandes, la fonction d'activation « se déclenche », sinon elle ne fait rien. En d'autres termes, une fonction d'activation est comme une porte qui vérifie qu'une valeur entrante est supérieure à un nombre critique[28].

Fonction d'activation utilisées :

1. **La fonction d'activation linéaire rectifiée ou ReLU** pour court est une fonction linéaire à la pièce qui produira l'entrée directement si elle est positive, sinon, il sortira zéro. Il est devenu la fonction d'activation par défaut pour de nombreux types de réseaux neuronaux parce qu'un modèle qui l'utilise est plus facile à former et atteint souvent de meilleures performances. Nous avons utilisé ReLU parce qu'il surmonte le problème de gradient qui disparaît, permettant à notre modèle d'apprendre plus vite et de mieux performer. Il s'agit également de l'activation par défaut lors du développement de réseaux neuronaux multicouches et convolutionnels[29].
2. **Softmax** : C'est une fonction d'activation très intéressante car non seulement elle mappe notre sortie sur une plage de $[0, 1]$, mais aussi chaque sortie de manière à ce que la somme totale soit 1. La sortie de Softmax est donc une distribution de probabilité. La fonction softmax est souvent utilisée dans la couche finale d'un classificateur réseau neuronal (comme nous le ferons)[30].

3.4.2 Optimisateurs et fonction de perte

1. **Les optimiseurs** : mettent à jour les paramètres de poids pour minimiser la fonction de perte. La fonction de perte sert de guide au terrain, indiquant à l'optimiseur s'il se déplace dans la bonne direction pour atteindre le fond de la vallée, le minimum global[31] .
2. **Fonction de perte** : il calcule l'erreur entre la prévision du réseau et la valeur réelle. Lors d'une tâche de classification, la variable aléatoire est discrète, car elle peut prendre uniquement les valeurs 0 ou 1, représentant l'appartenance (1) ou non (0) à une classe. C'est pourquoi la fonction de perte la plus courante et la plus adaptée est la fonction d'entropie croisée (en anglais cross-entropy).

Celle-ci est issue du domaine de la théorie de l'information, et mesure la différence globale entre deux distributions de probabilité (celle de la prévision du modèle, celle du réel) pour une variable aléatoire ou un ensemble d'évènements [32].

Fonction de perte utilisée :

- **L'entropie croisée binaire « Binary crossentropy »** : est une fonction de perte utilisée dans les tâches de classification binaire.

Ce sont des tâches qui répondent à une question avec seulement deux choix (oui ou non, A ou B, 0 ou 1, gauche ou droite). Il est possible de répondre à plusieurs questions indépendantes de ce type en même temps, comme dans la classification multi-étiquettes ou dans la segmentation d'images binaires[33].

Optimiseur utilisé :

- **adam** : L'algorithme d'optimisation Adam est utilisé pour la formation de modèles d'apprentissage profond. Il s'agit d'une extension de la descente de gradient stochastique. Dans cet algorithme d'optimisation, les moyennes courantes des gradients et des seconds moments des gradients sont utilisées. Il est utilisé pour calculer les taux d'apprentissage adaptatifs pour chaque paramètre, il s'agit de l'un des optimiseurs les plus utilisés car c'est la méthode la plus rapide, très rapide et qui converge également rapidement pour corriger la latence du taux d'apprentissage et le contraste élevé[34].

3.5 Approches proposées

Dans notre projet, nous allons proposer trois approches, dans chacune d'elles un modèle sera créé et formé, le résumé de nos propositions est le suivant :

1. **Utilisation de CNN à partir de zéro** : nous créerons un modèle CNN à partir de zéro.
2. **Utilisation de l'architecture Xception** : nous allons créerons un modèle CNN basé sur l'architecture Xception.
3. **Utilisation de l'architecture InceptionResNetV2** : nous allons créerons un modèle CNN basé sur l'architecture InceptionResNetV2.

3.5.1 Modèle de base :

Dans ce modèle, nous allons construire une architecture CNN à partir de zéro, puis tester notre modèle et voir les résultats que nous obtiendrons. La figure 3.5 donne un aperçu de notre modèle.

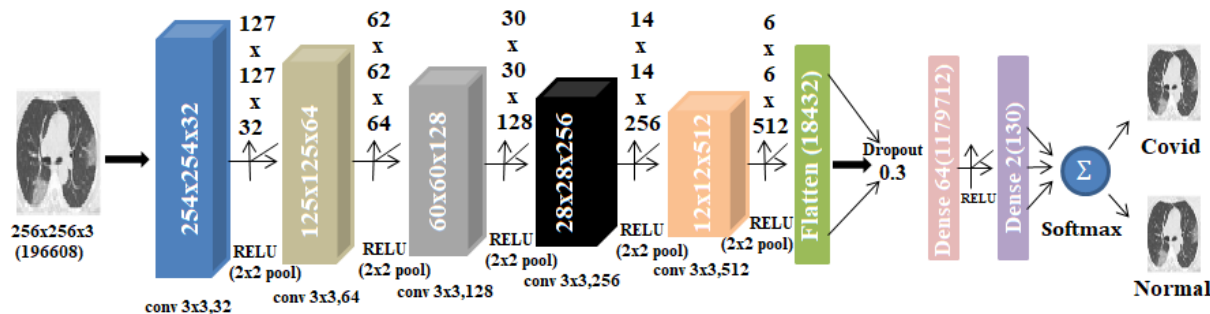


FIGURE 3.3 – Architecture de Modèle de base.

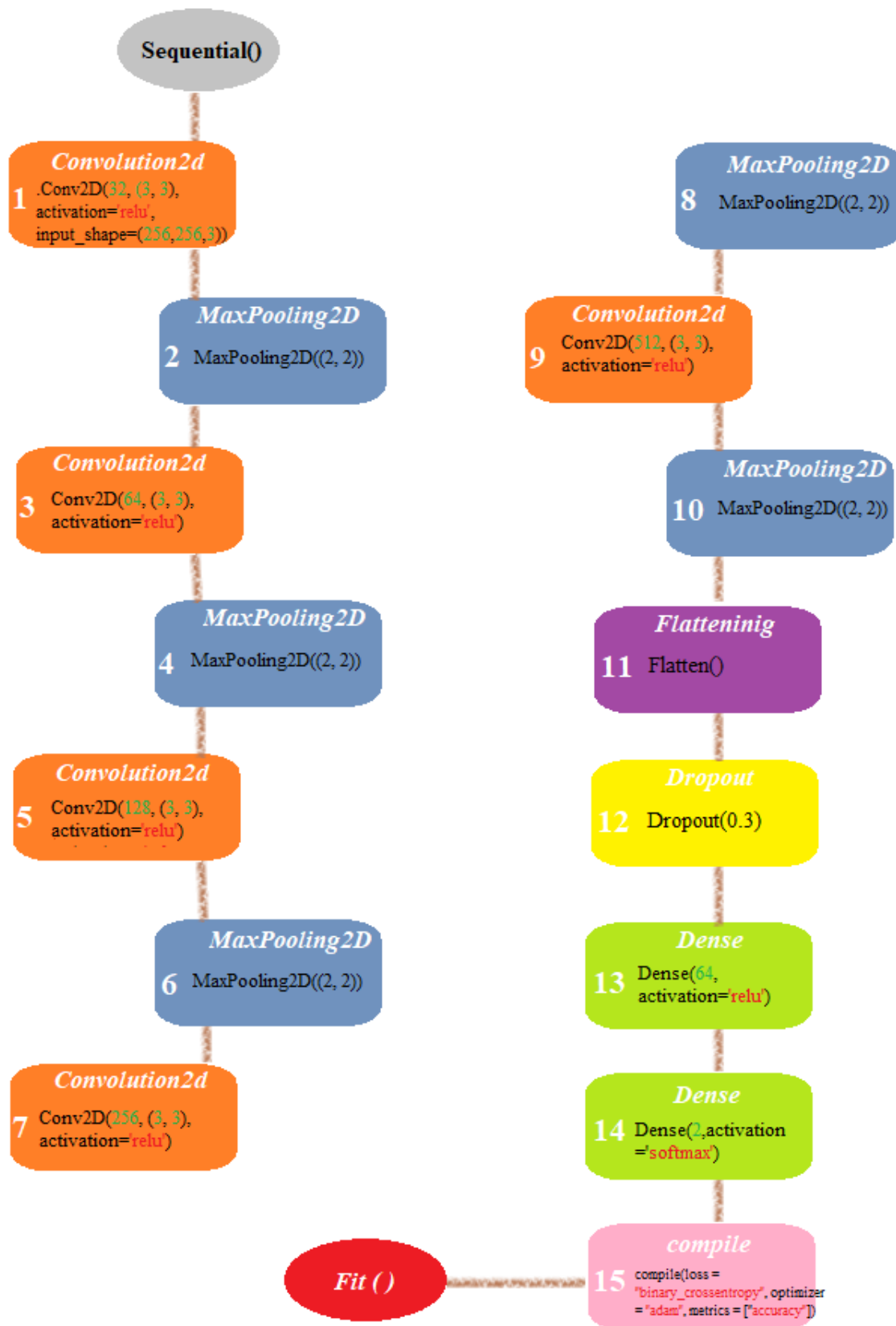


FIGURE 3.4 – diagramme principale de notre Modèle de base

Description du modèle :

Dans ce modèle, qui contient de nombreuses couches, et grâce à de nombreuses expérimentations, le résultat du choix de ces hyper paramètres était le meilleur parmi de nombreuses autres options.

Qui est choisi dans :

- **Cinq couches de convolution2D** qui diffèrent par le nombre de filtres, de sorte qu'il double au fur et à mesure que l'on s'enfonce, en commençant par la première couche, qui contient 32 filtres, et en terminant par la dernière couche, qui contient 512 filtres, tout cela afin d'extraire de nombreuses caractéristiques.
- **Cinq couches de max pooling** Après chaque couche convolution2D afin d'extraire les caractéristiques dominantes et réduire la puissance de calcul nécessaire pour traiter les données. En outre, Il élimine complètement les activations bruyantes et réduit également le bruit.
- **Couche de flattening** afin de rendre l'information en une seule dimension qui peut être introduit dans un classificateur de réseau neuronal entièrement connecté.
- **drop out** : Le processus de classification est effectué en prenant la caractéristique pondérée aplatie obtenue à partir de la **couche de flattening**, puis en supprimant **0,3** de ses paramètres pour empêcher tous les neurones de converger vers le même objectif, et aide les activations des unités cachées à devenir clairsemées, ce qui est caractéristique également souhaitable.
- **Une couches de Dense** Une couche dense d'une taille de 64, reçoit un vecteur d'entrée et produit un nouveau vecteur de sortie.

RMQ : Dans toutes ces couches, la fonction d'activation relu a été utilisée car elle est la plus avancée par rapport aux autres fonctions.

- Enfin, **la couche de dense** avec deux unités, en raison du nombre de classes finales pour que l'image d'entrée appartienne à une classe (COVID, NORMAL), a été utilisée dans laquelle la fonction SoftMax a été utilisée car c'est la fonction la plus utilisée dans les modèles multi-classes.

3.5.2 Modèle Xception :

Dans ce modèle, nous allons essayer de mettre en œuvre le modèle CNN basé sur l'architecture **Xception**, nous devons d'abord avoir une vue rapprochée à ce dernier.

— **Xception :**

Xception est une architecture efficace qui repose sur les deux points principaux : Convolution séparable Depthwise et raccourcis entre blocs de convolution comme dans ResNet, les données passent d'abord par le flux d'entrée, puis par le flux intermédiaire qui est répété huit fois, et enfin par le flux de sortie [35].

Une description complète des spécifications du réseau Xception est donnée à la figure 3.5

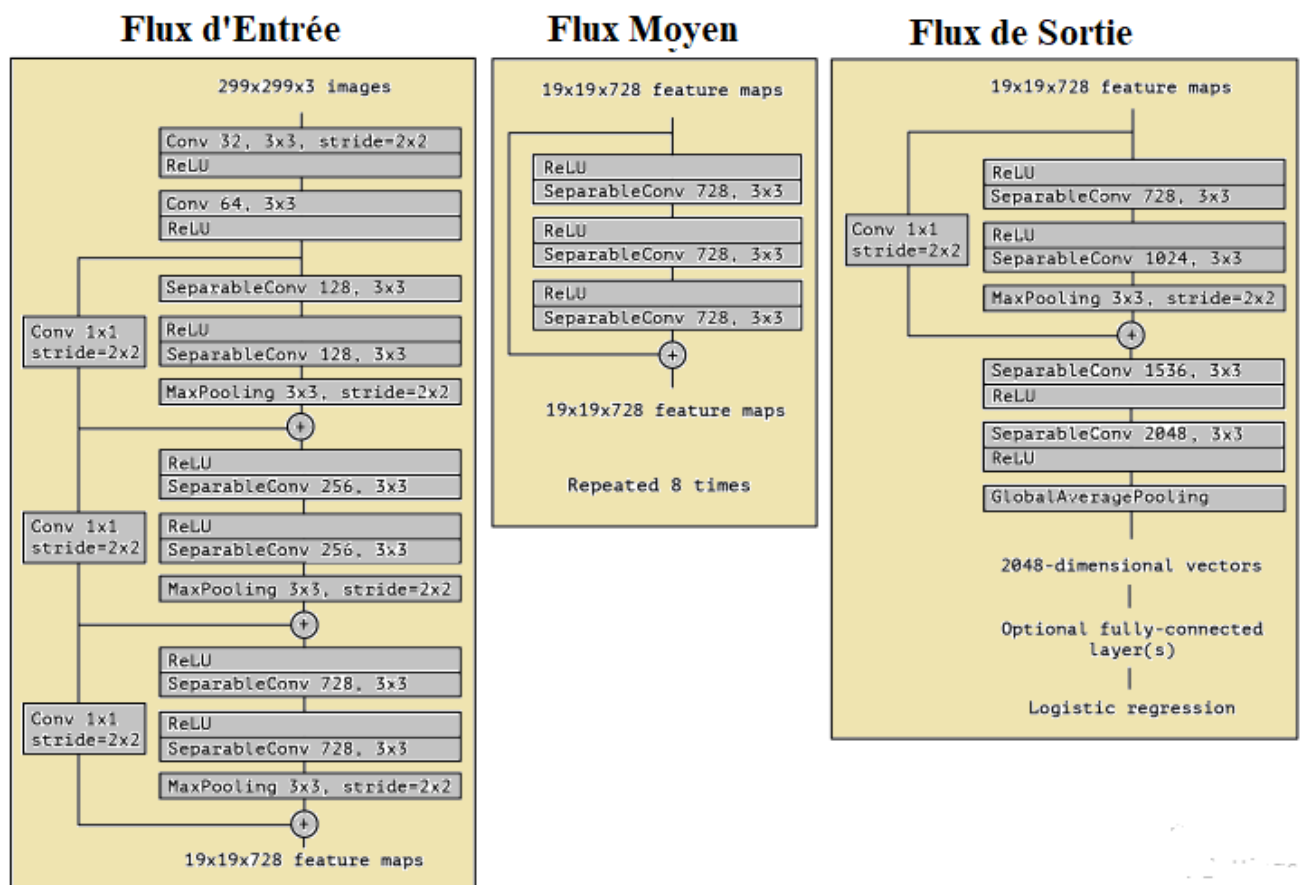


FIGURE 3.5 – L'architecture Xception.

Le réseau Xception a remplacé le réseau Inception. La création extrême est souvent appelée Xception. Au lieu de couches de convolution typiques, le réseau Xception uti-

lise des couches de convolution séparables en profondeur. Xception inclut la cartographie des corrélations spatiales et inter-canaux, qui dans les cartes de caractéristiques CNN peuvent être complètement dissociées. L'architecture sous-jacente d'Inception a survécu plus longtemps que Xception. Les 36 couches de convolution du modèle Xception peuvent être séparées en 14 modules différents. Une fois les première et dernière couches supprimées, chaque couche est entourée d'un lien résiduel continu. L'image d'entrée est convertie en corrélations spatiales dans chaque canal de sortie pour obtenir les corrélations entre canaux dans une image d'entrée. Après cela, une méthode de convolution 1×1 en profondeur est effectuée. Au lieu de cartes 3D, les relations peuvent être visualisées sous forme de carte $2D+1D$. Dans Xception, la première étape consiste à effectuer une corrélation spatiale 2D, suivie de corrélations spatiales 1D[36] .

Les données passent d'abord par le flux d'entrée, puis par le flux moyen, qui est répété huit fois, et enfin par le flux de sortie [37].

Après avoir appris ce qu'est Xception, nous allons maintenant lui appliquer quelques modifications pour avoir notre modèle personnalisé (Figure 3.8) :

1. Modifiez les poids du premier calque à (256, 256, 3) car les images sont 256×256 dans `IMREAD COLOR` (3 canaux).
2. Supprimez le classificateur de Xception.
3. Ajoutez une couche de regroupement de moyenne globale (`GlobalAveragePooling2D`) qui sera connectée à la dernière couche de l'architecture Xception.
4. Ajoutez une couche dense de deux neurones, en leur appliquant la fonction d'activation softmax pour sortir nos deux classes (COVID19/NORMAL), le nouveau classifieur de ce modèle sera la combinaison de la couche de pooling moyenne globale et de la couche dense.

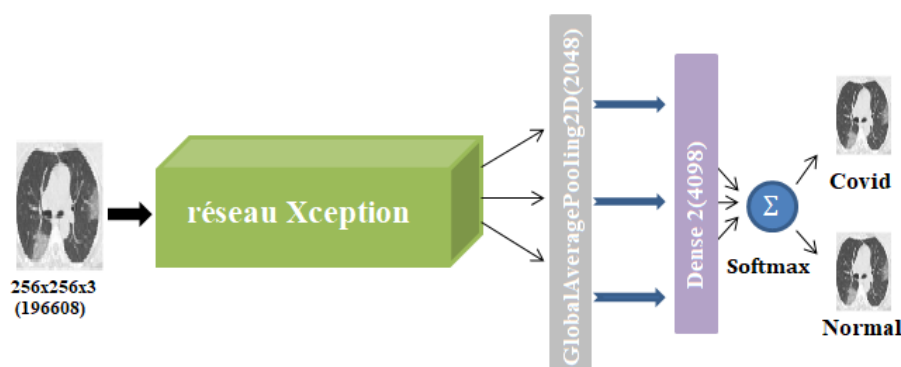


FIGURE 3.6 – Architecture Xception finale.

Description du modèle :

Tout d’abord, nous donnerons des images comme entrée au réseau de Xception, après que la sortie passera par la couche de « Global average pooling », cette opération est conçue pour remplacer les couches entièrement connectées en CNNs classiques (couche dense), au lieu d’ajouter des couches entièrement connectées au-dessus des cartes de caractéristiques générées par les couches convolutionnelles, nous prenons la moyenne de chaque carte de caractéristiques et le vecteur résultant est introduit directement dans la couche softmax.

3.5.3 model InceptionResNetV2 :

InceptionResNetV2 : Inception-Resnet-v2 [38], est formulé sur la base d’une combinaison de la structure Inception et de la connexion Residual. Dans le bloc Inception-Resnet[38] , des filtres convolutifs de tailles multiples sont combinés par des connexions résiduelles. L’utilisation des connexions résiduelles évite non seulement le problème de dégradation éliminé par les structures profondes, mais également réduit le temps de formation. La figure 3 montre l’architecture réseau de base d’Inception-Resnet-v2.

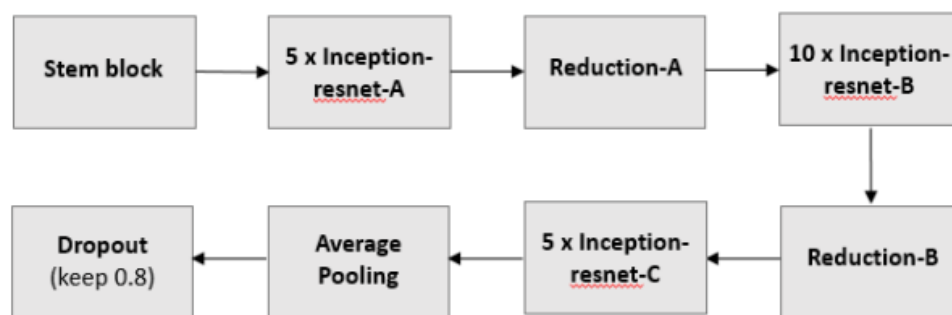


FIGURE 3.7 – L'architecture InceptionResNetV2.

Le modèle InceptionResNetV2 est illustré à la figure 5, où la couche de stem est la même que le modèle InceptionV4, et le reste se compose de (a) module InceptionResNet-A de grille 35×35 , (b) 35×35 à 17×17 Réduction- Un module, (c) un module InceptionResNet-B à grille 17×17 , (d) un module Reduction-B 17×17 à 8×8 et le dernier (e) un module InceptionResNet-C à grille 8×8 [39] .

Pour personnaliser notre modèle afin qu'il corresponde à notre objectif, les mêmes modifications que le modèle précédent seront également apportées (Figure 3.10).

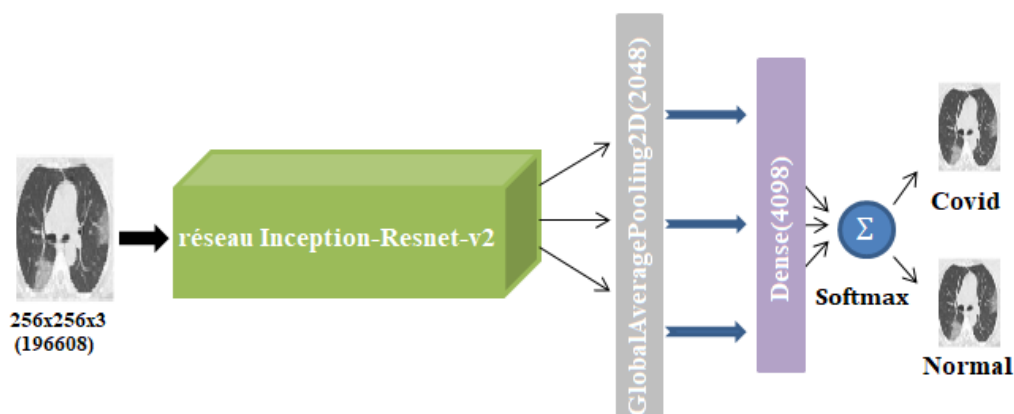


FIGURE 3.8 – Architecture InceptionResNetV2 finale.

Description du modèle : Tout d'abord, nous donnerons des images comme entrée au réseau de InceptionResNetV2, après que la sortie passera par la couche de « Global average pooling », cette opération est conçue pour remplacer les couches entièrement connectées en CNNs classiques (couche dense), au lieu d'ajouter des couches entièrement connectées au-dessus des cartes de caractéristiques générées par les couches convolutionnelles, nous prenons la moyenne de chaque carte de caractéristiques et le vecteur résultant est introduit

directement dans la couche softmax.

Enfin, les procédures que nous avons effectuées dans notre projet sur la base des modèles que nous avons réalisés sont présentées dans la figure ci-dessous :

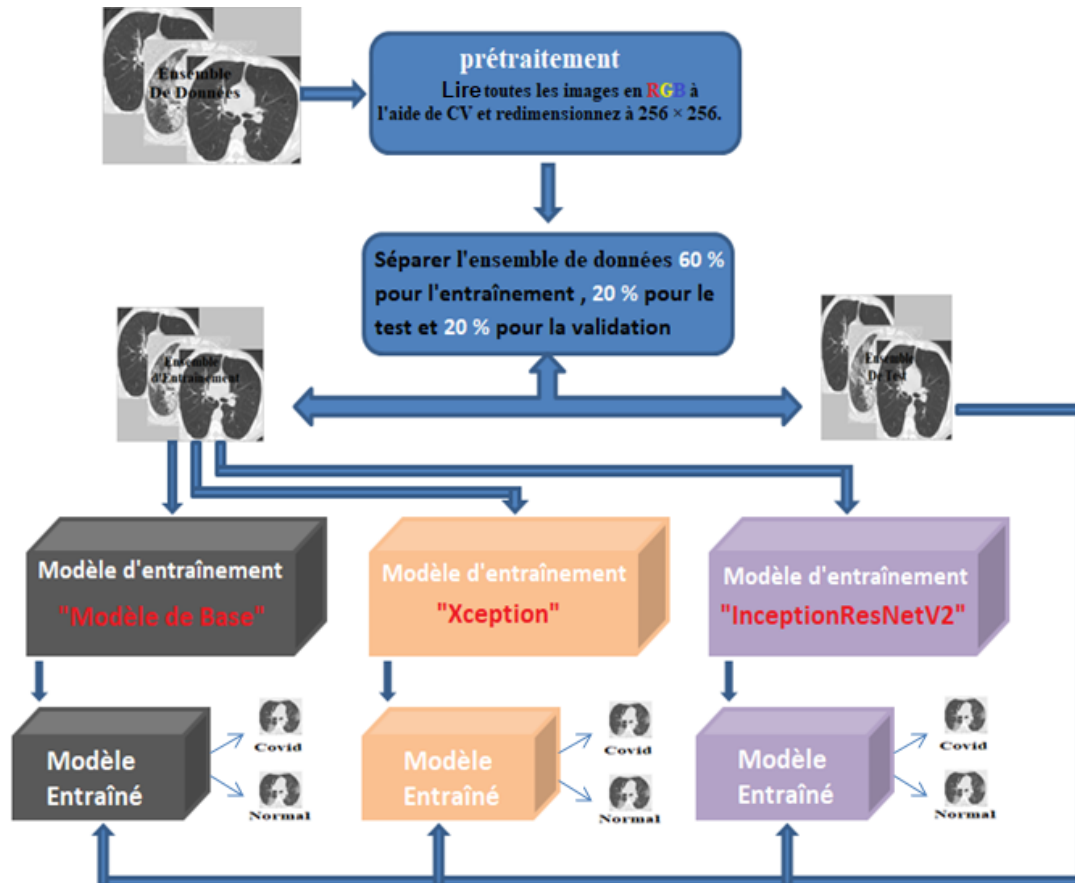


FIGURE 3.9 – Nos procédures de projet.

3.6 Conclusion

Nous avons décrit l'ensemble de données que nous utiliserions dans notre recherche dans ce chapitre. Ensuite, nous avons discuté de l'étape de préparation de la dernière. Enfin, nous avons fourni des moyens de faire avancer notre objectif. Dans le prochain chapitre, les résultats et les instruments de cette mise en œuvre seront discutés.

Chapitre **4**

Implementation et Tests

4.1 Introduction

Dans ce chapitre, nous parlerons des différentes bibliothèques utilisées, de l'environnement et du langage de programmation choisi pour implémenter notre modèle. Nous présenterons également les indicateurs de performance de la classification. Ensuite, nous décrirons la mise en œuvre de nos modèles et présentons son entraînement sur le dataset, puis dans la phase de test et de prédiction. Nous évaluerons nos modèles entraînés. A la fin, nous discuterons des résultats obtenus.

4.2 Langage de programmation, Bibliothèques et Environnement utilisé

Le langage de programmation que nous avons utilisé précédemment pour notre méthode était Python, et les modèles que nous avons générés dans ce chapitre sont également écrits en Python.

4.2.1 Langage de programmation

Python

Python est le langage de programmation le plus populaire actuellement et dernières années parmi les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En

effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages[40] .

4.2.2 FrameWork

TensorFlow

TensorFlow est un Framework open source développé par les chercheurs de Google destiné pour l'implémentation des algorithmes d'apprentissage automatique l'apprentissage en profondeur et d'autres charges de travail d'analyse statistique et prédictive elle offre aussi une grande flexibilité dans le cadre de l'utilisation pour le développement d'un réseau CNN, alors nous avons utilisé cette bibliothèque pour définir les composants de base de notre l'architecture [41].

4.2.3 Environnement de développement

Tout apprentissage du réseau de neurones nécessite une forte puissance de calcul. Notre méthode est basée sur un réseau convolutif profond, qui implique un grand nombre de points à former. La formation sur le PC portable aurait également pris beaucoup de temps. Nous avons utilisé l'environnement de bloc-notes Google Colab.

Google colab

Google Collaboratory ou Colab, un outil Google simple et gratuit pour vous initier au deep Learning Permet d'écrire et d'exécuter du code en Python, Importation et l'enregistrement des blocs-notes depuis et vers Google Drive, basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud [42, 43, 44] . Google Collab permet également :

- D'améliorer vos compétences de codage en langage de programmation Python.
- De développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV.

- D'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.

Mais la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuitement.

4.2.4 Bibliothèques utilisées

Keras

Keras est une bibliothèque d'apprentissage en profondeur écrite en Python, s'exécutant sur la plate-forme d'apprentissage automatique TensorFlow. Il a été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat le plus rapidement possible est essentiel pour faire de bonnes recherches[45, 46] .

Par conséquent, nous avons utilisé cette bibliothèque pour construire le réseau avec la plupart de ses couches comme la convolution, max pooling, flattting, dense etc ... et aussi dans le téléchargement et le l'enregistrement des données et l'affichage des résultats.

NumPy

NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués) et un assortiment de routines pour des opérations rapides sur des tableaux, notamment mathématiques, logiques, manipulation de forme, tri, sélection, nous avons utilisé cette bibliothèque pour adapter le type d'entrée et aussi pour la manipulation des tableau[47].

Matplotlib

Matplotlib est une bibliothèque complète permettant de créer des visualisations statiques, animées et interactives en Python tandis que nous utilisons cette bibliothèque pour visualiser tous les graphiques donnés et les statistiques sous forme de graphiques[48] .

OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque de logiciels open source de vision par ordinateur et d'apprentissage automatique. OpenCV a été conçu pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception par machine dans les produits commerciaux. En tant que produit sous licence BSD, OpenCV permet aux entreprises d'utiliser et de modifier facilement le code [49].

Scikit-learn

Nous avons utilisé cette bibliothèque pour effectuer les calculs nécessaires à la visualisation de certains graphiques et statistiques tels que la matrice de confusion et la fonction d'efficacité du récepteur ainsi que les arguments d'apprentissage du modèle[50] .

Pandas

Pandas est un outil d'analyse et de manipulation de données open source rapide, puissant, flexible et facile à utiliser, construit sur le langage de programmation Python[51] .

4.3 Mesures (métriques) de performance de classification

Afin d'effectuer une évaluation expérimentale des modèles considérés, diverses mesures de performance telles que l' Accuracy, F1-Score, la spécificité, la précision et le Recall (sensibilité) sont utilisées dans ce travail. Ces mesures d'évaluation sont particulièrement utiles lors de l'évaluation d'un système de dépistage médical, c'est pourquoi elles sont choisies pour la tâche de prédiction du COVID19.

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

1. **TP : VP (Vrai Positif)** : la prédiction est positive et le patient a le COVID19.

2. **TN : VN (Vrai Négatif)** : la prédiction est négative et le patient est en bonne santé.
3. **FP (Faux Positif)** : la prédiction est positive et le patient est en bonne santé.
4. **FN (Faux Négatif)** : la prédiction est négative et le patient a le COVID19.

Toutes les métriques utilisées sont expliquées dans le tableau ci-dessous :

Indicateur	Description	Equation
Accuracy	c'est le rapport des sujets correctement étiquetés.	$Acc = \frac{TP + TN}{TP + FP + TN + FN}$
Precision	la précision est le rapport des étiquetés correctement positifs par notre programme sur tous les étiquetés positifs.	$Pre = \frac{TP}{TP + FP}$
Recall (aka sensitivity)	le rappel est le rapport entre les personnes correctement étiquetées positives par notre programme et toutes celles qui sont infectées dans la réalité	$Recall = \frac{TP}{TP + FN}$
F1-Score	c'est la moyenne harmonique (average) de la précision et du rappel.	$FScore = \frac{2 * (Recall + Precision)}{Recall + Precision}$

TABLE 4.1 – évaluations de métriques « metrics »

4.4 Implémentation

Dans cette section, nous allons parler de l'implémentation de notre projet, sera essentiellement orienté pour décrire la création de nos modèles.

4.4.1 Prétraitement des ensembles de données

Ici, nous expliquerons les modifications que nous avons apportées à notre ensemble de données pour s'adapter à notre projet.

Étape 1 :

Importez l'ensemble de données de kaggle et placez-le dans un dossier créé précédemment appelé (Dataset).

```
!mkdir /root/.kaggle/  
!cp /content/kaggle.json /root/.kaggle  
!mkdir dataset  
!kaggle datasets download -d plameneduardo/sarscov2-ctscan-dataset  
!unzip /content/sarscov2-ctscan-dataset.zip -d /content/dataset
```

FIGURE 4.1 – Code d'étape 1 pour le prétraitement de l'ensemble de données.

Étape 2 :

Utilisez la bibliothèque "glob" pour créer une liste contenant le chemin d'accès « path » de chaque image dans le dossier que nous avons créé à l'étape 1.

```
images = glob('/content/dataset/**/*.*png', recursive=True)
```

FIGURE 4.2 – Code d'étape 2 pour le prétraitement de l'ensemble de données

Étape 3 :

Créer deux listes vides, une pour la Classe 0 : (non covid) et une pour la Classe 1 : (covid). Ensuite, nous parcourir la liste que nous avons créée à l'étape 2 en fonction du nom et mettons le chemin d'accès à chaque image dans la liste avec le même nom.

```
class0=list() #negative (non covid)
class1=list() #positive (covid)
for filename in images :
    if 'Non-Covid' in filename:
        class0.append(filename)
    else :
        class1.append(filename)
```

FIGURE 4.3 – Code d'étape 3 pour le prétraitement de l'ensemble de données.

Étape 4 :

Nous lisons les images des deux listes que nous avons créées à l'étape 3 et utilisons la bibliothèque « OpenCV » pour extraire les données de chaque image dans une liste et utilisons la fonction "append" pour ajouter son étiquette "lable" (0 ou 1) à l'intérieur du liste de chaque image.

```
def get_image_arrays(data , label) :
    img_arrays = list()
    for filename in data :
        if filename.endswith('.png'):
            img = cv2.imread(filename , cv2.IMREAD_COLOR)
            img_sized = cv2.resize(img , (256,256) , interpolation = cv2.INTER_LINEAR)
            img_arrays.append([img_sized,label])
    return img_arrays
```

FIGURE 4.4 – Code d'étape 4 pour le prétraitement de l'ensemble de données.

Étape 5 :

Nous formons deux tableaux « arrays », une pour la Classe 0 : (non covid) et une pour la Classe 1 : (covid). Dans chaque tableau « array », nous mettons les listes d'images de la même classe que nous avons extraites à l'étape 4.

```
class_0_array = get_image_arrays(class0 , 0)
class_1_array = get_image_arrays(class1 , 1)
```

FIGURE 4.5 – Code d'étape 5 pour le prétraitement de l'ensemble de données.

Étape 6 :

Nous concaténer les tableaux « arrays » que nous avons extraites à l'étape 5. Ensuite, nous les randomisons à l'aide de la fonction 'shuffle' afin d'obtenir une tableau « array » avec les données de chaque image et son étiquette 'lable'.

```
combined_data = class_0_array + class_1_array
random.seed(45)
random.shuffle(combined_data)
```

FIGURE 4.6 – Code d'étape 6 pour le prétraitement de l'ensemble de données.

Étape 7 :

Nous avons divisé le tableau « array » que nous avons extraite à l'étape 6 en deux listes, l'une contenant les données d'image et l'autre contenant les étiquettes d'image 'lable' (1 ou 0).

```
x = list()
y = list()

for image in combined_data:
    x.append(image[0])
    y.append(image[1])
```

FIGURE 4.7 – Code d'étape 7 pour le prétraitement de l'ensemble de données.

Étape 8 :

Nous avons converti la liste de données que nous avons extraite à l'étape 7 en un tableau numérique «numpy array » afin de pouvoir le soumettre au modèle (car le tableau normal ne contient pas la forme "shape").

```
X = np.array(X).reshape(-1,256,256,3)
```

FIGURE 4.8 – Code d'étape 8 pour le prétraitement de l'ensemble de données.

Étape 9 :

convertir la liste d'étiquettes 'lables' en une liste catégorique puisque nous allons avoir 2 neurones dans la sortie du modèle.

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

FIGURE 4.9 – Code d'étape 9 pour le prétraitement de l'ensemble de données.

4.4.2 Modèles créés :

Au cours de nos expérimentations, nous avons créé plusieurs modèles avec différentes architectures basées sur les approches proposées. L'architecture des modèles est illustrée ci-dessous :

1. **Modèle de base** :L'architecture de ce modèle est représentée dans le tableau ci-dessous :


```

Model: "sequential_3"

```

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_13 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_13 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_14 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_14 (MaxPooling2D)	(None, 30, 30, 128)	0
conv2d_15 (Conv2D)	(None, 28, 28, 256)	295168
max_pooling2d_15 (MaxPooling2D)	(None, 14, 14, 256)	0
conv2d_16 (Conv2D)	(None, 12, 12, 512)	1180160
max_pooling2d_16 (MaxPooling2D)	(None, 6, 6, 512)	0
flatten_3 (Flatten)	(None, 18432)	0
dropout_3 (Dropout)	(None, 18432)	0
dense_6 (Dense)	(None, 64)	1179712
dense_7 (Dense)	(None, 2)	130

```

Total params: 2,748,418
Trainable params: 2,748,418
Non-trainable params: 0

```

TABLE 4.2 – Résumé du modèle de base.

2. **Modèle de Xception** :L'architecture de ce modèle est représentée dans le tableau ci-dessous :

```

↳ Model: "sequential_2"

```

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 8, 8, 2048)	20861480
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 2048)	0
dense_2 (Dense)	(None, 2)	4098

```

=====
Total params: 20,865,578
Trainable params: 20,811,050
Non-trainable params: 54,528
=====

```

TABLE 4.3 – Résumé du modèle de Xception.

3. **modèle InceptionResNetV2** :L'architecture de ce modèle est représentée dans le tableau ci-dessous :

```

↳ Model: "sequential"

```

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Functiona l)	(None, 6, 6, 1536)	54336736
global_average_pooling2d (G lobalAveragePooling2D)	(None, 1536)	0
dense (Dense)	(None, 2)	3074

```

=====
Total params: 54,339,810
Trainable params: 54,279,266
Non-trainable params: 60,544
=====

```

TABLE 4.4 – Résumé du modèle de InceptionResNetV2.

4.5 L'apprentissage et tests

Dans cette section, nous parlerons des phases de formation et de test de nos modèles. Il est divisé en deux sous-sections, la première sous-section sera dédiée à l'entraînement de nos modèles à l'aide d'images de train, que nos modèles sont entraînés sur 20 époques, et la seconde contiendra le test de nos modèles à l'aide d'images de test.

Et pour faire le fitting de notre modèle, nous avons sélectionné et choisi quelque paramètre pour le meilleur résultat :

- Targe tsize = (256, 256) : pour rendre tous les image a la meme size .
- Batch size = 32 : par défaut ,La taille de lot de 32 nous a donné le meilleur résultat.
- Class mode = 'categorical' parce qu'on a deux classes, la classe normale et la classe covid.

4.5.1 L'apprentissage

Tous les modèles sont entraînés sur une partie de l'ensemble de données, comme il est résumé dans le tableau ci-dessous.

Dataset	Nombre d'images
Ensemble d'entraînement	1489
Ensemble de test	497
Ensemble de Validation	497

TABLE 4.5 – Représentation de l'ensemble de données utilisé.

Tous les résultats (accuracy et loss values) obtenus lors de la phase d'entraînement sont présentés ci-dessous pour chaque modèle.

1. Modèle de base :

Modèle de base	
Accuracy	91.18%
loss	0.26
Time	2min

TABLE 4.6 – Accuracy/Loss/Time pour le modèle de base.

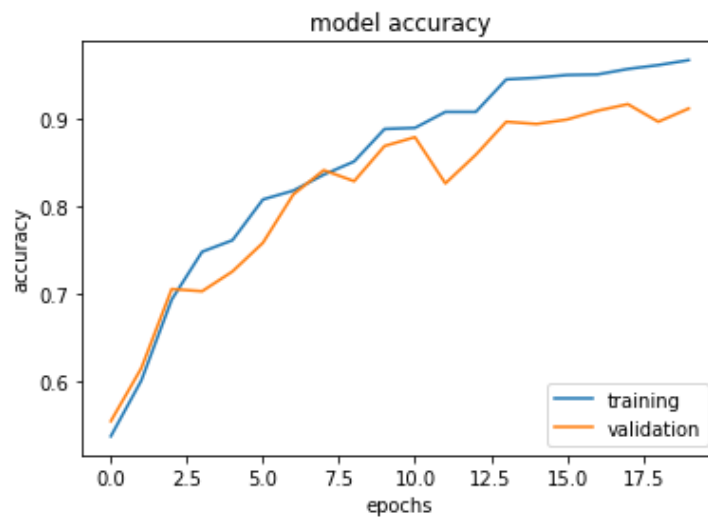


FIGURE 4.10 – Accuracy du modèle de base

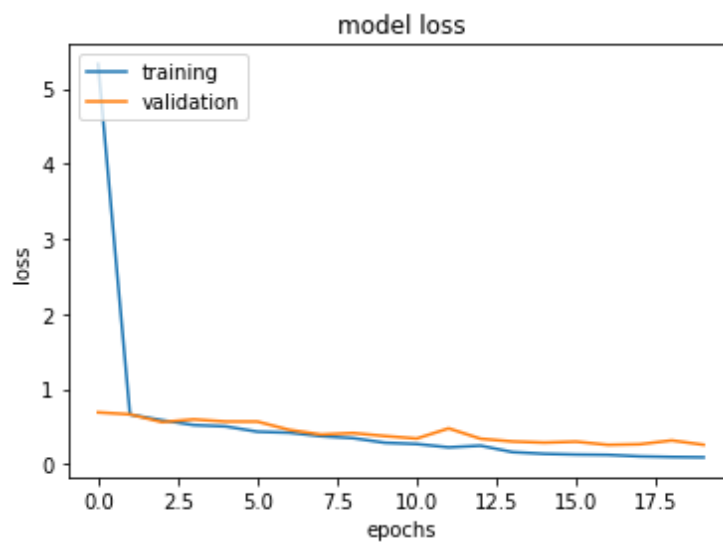


FIGURE 4.11 – loss du modèle de base.

2. Modèle XCeption :

Modèle de XCeption	
Accuracy	97.48%
loss	0.068
Time	15min

TABLE 4.7 – Accuracy/Loss/Time pour le modèle XCeption.

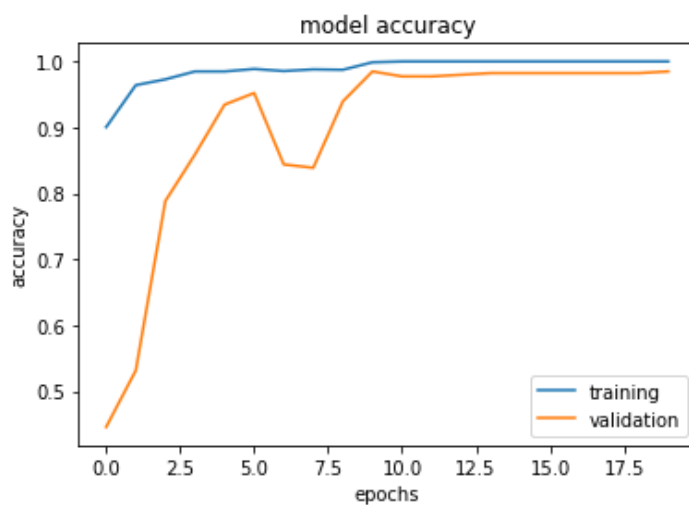


FIGURE 4.12 – Accuracy du modèle XCeption.

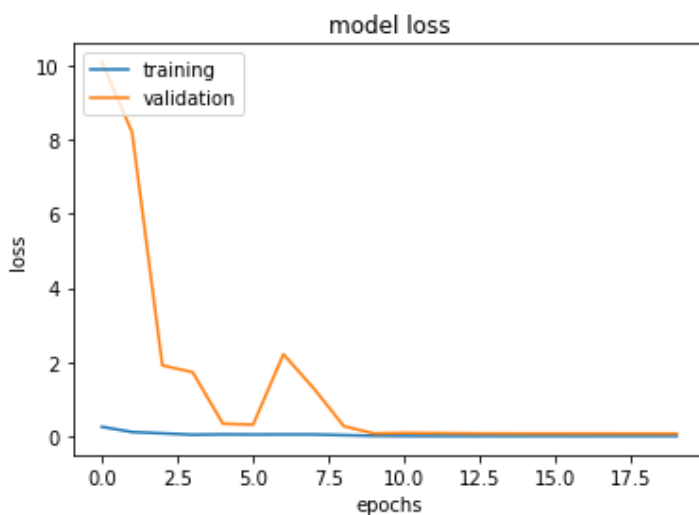


FIGURE 4.13 – loss du modèle XCeption.

3. Modèle InceptionResNetV2 :

Modèle de InceptionResNetV2	
Accuracy	98.49%
loss	0.05
Time	15min et30s

TABLE 4.8 – Accuracy/Loss/Time pour le modèle InceptionResNetV2.

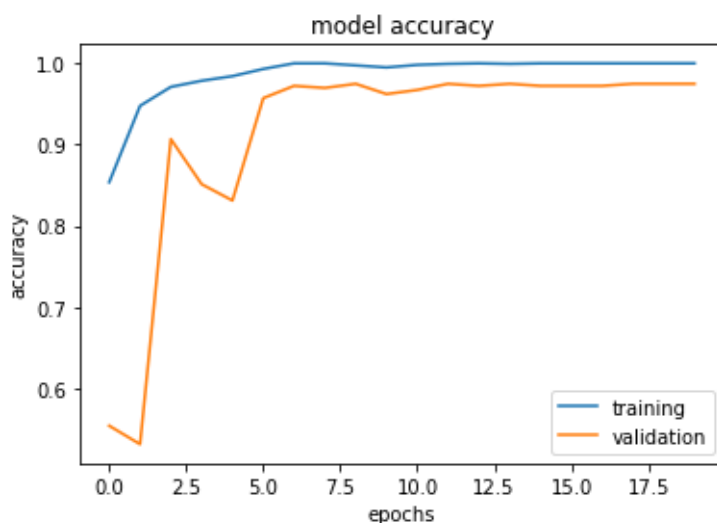


FIGURE 4.14 – Accuracy du modèle InceptionResNetV2.

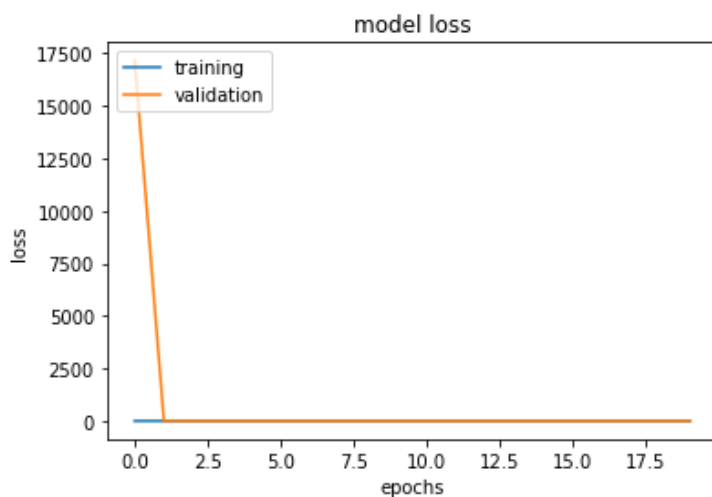


FIGURE 4.15 – loss du modèle InceptionResNetV2.

4.5.2 Tests :

1. Modèle de base

— La matrice de confusion :



FIGURE 4.16 – La matrice de confusion du modèle de base.

— Estimation de la performance du modèle :

D'après La matrice de confusion du modèle de base on a $TP=231$; $TN=217$; $FP=27$; $FN=22$, Sur lequel nous avons appliqué les formules métriques mentionnées dans le tableau (4.1) pour obtenir les résultats comme indiqué dans le tableau ci-dessous :

Accuracy	90%
Recall	91.3%
Specificity	88.9%
Precision	89.5
F1-Score	90.3

TABLE 4.9 – Résultats des tests du modèle de base.

2. Modèle XCeption :

— La matrice de confusion

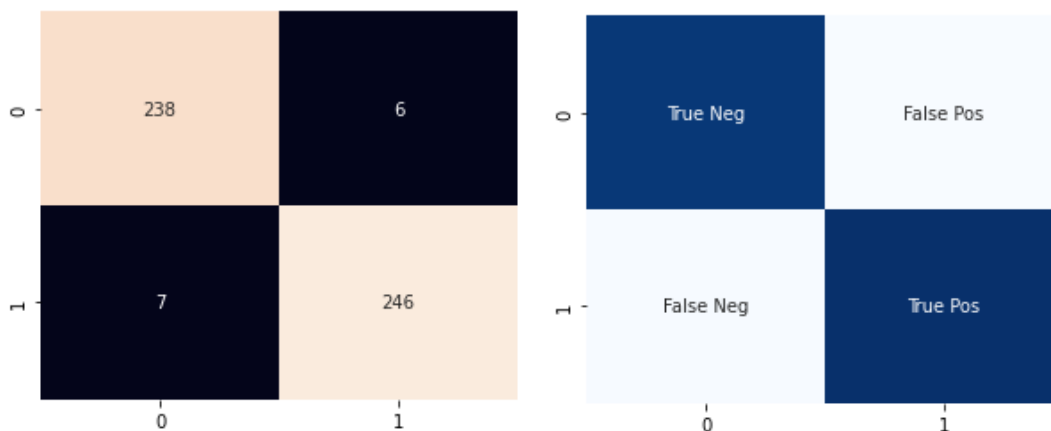


FIGURE 4.17 – La matrice de confusion du modèle XCeption

— Estimation de la performance du modèle

D'après La matrice de confusion du modèle XCeption on a $TP=246$; $TN=238$; $FP=6$; $FN=7$, Sur lequel nous avons appliqué les formules métriques mentionnées dans le tableau(4.1) pour obtenir les résultats comme indiqué dans le tableau ci-dessous :

Accuracy	97.4%
Recall	97.2%
Specificity	97.5%
Precision	97.6
F1-Score	92

TABLE 4.10 – Résultats des tests du modèle XCeption

3. Modèle InceptionResNetV2 :

— La matrice de confusion :

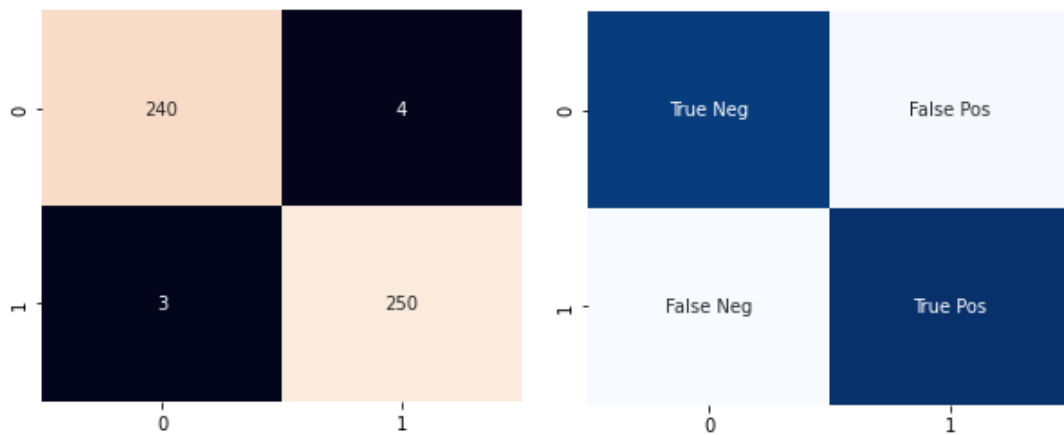


FIGURE 4.18 – La matrice de confusion du modèle InceptionResNetV2..

— Estimation de la performance du modèle :

D'après La matrice de confusion du modèle InceptionResNetV2 on a $TP=250$; $TN=240$; $FP=4$; $FN=3$, Sur lequel nous avons appliqué les formules métriques mentionnées dans le tableau(4.1) pour obtenir les résultats comme indiqué dans le tableau ci-dessous :

Accuracy	98.5%
Recall	98.8%
Specificity	98.3%
Precision	98.4
F1-Score	98.5

TABLE 4.11 – du modèle InceptionResNetV2.

4.6 Analyse et discussion

Dans cette section, nous allons comparer les modèles en fonction de leurs résultats obtenus à partir de la formation, et déterminer quel modèle parmi eux est le meilleur.

4.6.1 Comparaison des modèles

À ce stade, nous avons 3 modèles avec les résultats, résumés comme suit :

Resultats De Test			
Modèle	Base	Xception	InceptionResNetV2%
Accuracy	90%	97.4%	98.5 %
Recall	91.3%	97.2	98.8%
Specificity	88.9%	97.5%	98.3%
Precision	89.5%	97.6%	98.4%
F1-Score	90.3%	92%	98.5%

Table 4.12 : Résultats de tous les modèles.

Maintenant, nous comparons ces trois modèles pour choisir notre meilleur modèle :

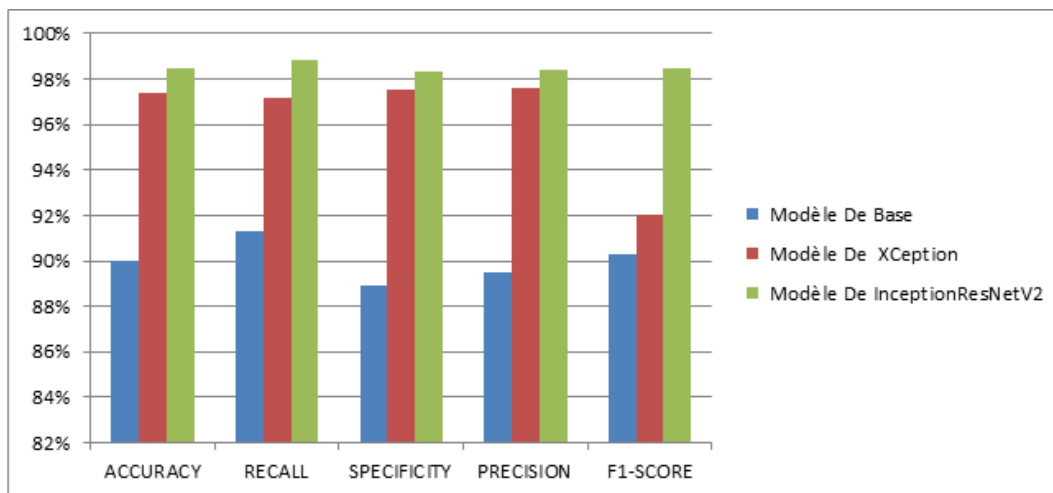


Figure 4.19 : Comparaison de tous les modèles.

Enfin, nous pouvons dire que dans tous les modèles d'approches proposés, InceptionResNetV2 a les résultats les plus élevés pour la détection de COVID-19 dans tous les termes de métriques.

4.6.2 Comparaison avec d'autres ouvrages

Après avoir connu le meilleur modèle que nous avons créé, il est maintenant temps de le comparer à un autre projet existant, qui est Identifier COVID19 à partir d'images CT Chest : Une approche de classification eXplainable Deep Learning « xDNN » [green52], Il a été appliqué au même ensemble de données de CT scan SARS-CoV-2 que nous avons utilisé dans notre projet. Comme indiqué dans le Tableau 4.13 et la Figure 4.20.

Modèle InceptionResNetV2 VS Modèle xDNN		
Modèle	InceptionResNetV2	xDNN
Accuracy	98.5%	97.38%
Recall	98.8%	97.2
Specificity	98.3%	95.53%
Precision	98.4%	91.6%
F1-Score	98.5%	97.31%

Table 4.13 : Modèle InceptionResNetV2 VS Modèle xDNN.

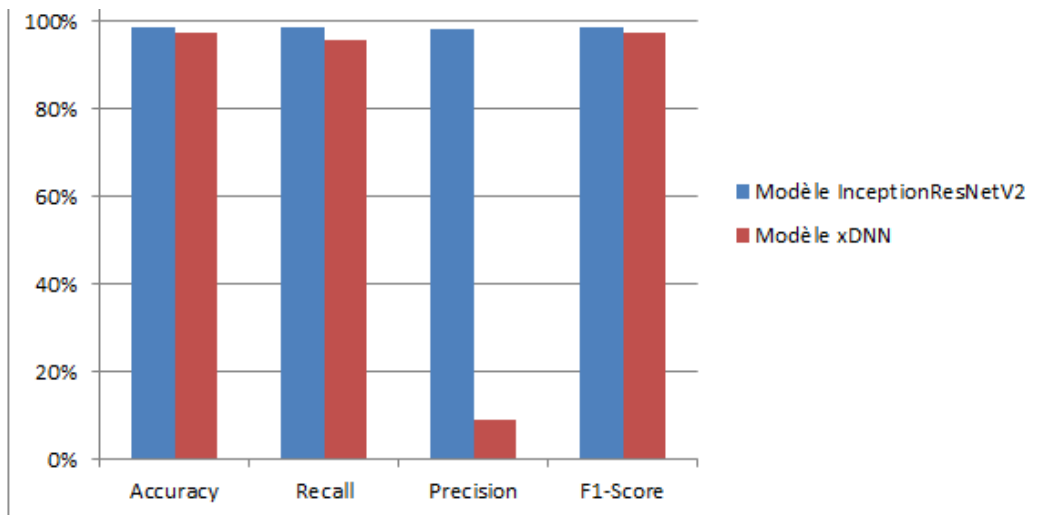


Figure 4.20 : Modèle InceptionResNetV2 VS Modèle xDNN.

D'après la figure ci-dessus, il est évident que notre modèle (modèle InceptionResNetV2) est meilleur que le modèle xDNN dans toutes les métriques.

4.7 Conclusion

Dans le dernier chapitre, nous avons présenté le langage de programmation, les bibliothèques et l'environnement utilisés pour implémenter notre projet. Ensuite, nous avons introduit les indicateurs de performance de classification (métriques), puis nous sommes passés à l'implémentation qui décrit comment nous traitons l'ensemble de données et les modèles que nous avons créés, puis les phases d'entraînement et de test ainsi que les différents résultats obtenus à partir de chaque modèle et les a comparés, Et enfin la comparaison finale avec d'autre ouvrage.

Conclusion générale

Étant donné que la classification des maladies pulmonaires en général par l'intelligence artificielle et l'apprentissage en profondeur est indispensable dans la période actuelle en raison du développement technologique rapide dans tous les domaines, en particulier la médecine, et de la disponibilité des outils et technologies nécessaires, et avec l'émergence de la Virus COVID-19, les chercheurs ont tenté de trouver des solutions alternatives pour diagnostiquer le virus COVID.-19 Parmi elles se trouve l'utilisation de l'apprentissage en profondeur sur la tomographie informatisée de la poitrine, qui s'est avérée être une méthode efficace et peut résoudre le problème du diagnostic COVID-19, qui prend plus de temps en utilisant des méthodes traditionnelles telles que (RT-PCR).

Sur la base de ce que nous avons mentionné, dans cette thèse, nous avons proposé trois méthodes d'apprentissage en profondeur basées principalement sur les réseaux de neurones convolutifs qui se sont avérés efficaces pour classer les images médicales, en commençant par le modèle CNN de base développé à partir de zéro, puis en utilisant l'architecture Xception et enfin en utilisant l'architecture InceptionResNetV2. Sur la base de ces trois approches, nous avons construit trois modèles (modèle CNN à partir de zéro, modèle CNN basé sur l'architecture Xception et modèle CNN basé sur l'architecture InceptionResNetV2), tous ces modèles ont été formés sur une partie d'un ensemble de données qui est des images de scanner thoracique, qui contiennent des images de personnes infectées et de personnes en bonne santé, afin de rendre nos modèles capables de classer les tomographies du thorax dans les classes COVID-19 et Normal, les résultats que nous avons obtenus après avoir testé nos modèles prouvent l'efficacité de nos modèles proposés.

Dans ce projet, nous avons introduit certaines des bases médicales en définissant COVID-19 et en couvrant et types d'imagerie médicale. Ensuite, nous avons parlé d'intelligence artificielle, d'apprentissage automatique et d'apprentissage en profondeur et nous avons abordé certains des travaux liés à l'utilisation de l'apprentissage en profondeur pour détecter des maladies à partir d'images médicales, puis nous sommes passés aux matériaux et aux méthodes que nous avons utilisés pour mettre en œuvre nos modèles où nous avons expliqué et présenté l'ensemble de données et les modèles que nous

avons proposés. Enfin, nous avons discuté des résultats obtenus en mettant en œuvre et en testant nos modèles et en les comparant à d'autres travaux, qui ont prouvé leur supériorité .Enfin, nous espérons que ce projet aidera à trouver une solution pour réduire cette pandémie.

Bibliographie

- [1] P. crosta, medically reviewed by university of illinois, "what to know about viruses". (2017). [online]. available at: <https://www.medicalnewstoday.com/articles/158179>.
- [2] Peng Zhou, Xing-Lou Yang, Xian-Guang Wang, Ben Hu, Lei Zhang, Wei Zhang, Hao-Rui Si, Yan Zhu, Bei Li, Chao-Lin Huang, et al. Discovery of a novel coronavirus associated with the recent pneumonia outbreak in humans and its potential bat origin. *BioRxiv*, 2020.
- [3] AR Fehr, S Perlman, HJ Maier, E Bickerton, and P Britton. An overview of their replication and pathogenesis; section 2 genomic organization. *Methods in Molecular Biology*, 1282:1–23, 2015.
- [4] 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*, 395(10223):497–506, 2020.
- [5] Coronavirus pandemic www.who.int 13/04/2022.
- [6] Adam A Grainger and Allison D Grainger. Allison’s diagnostic radiology: a textbook of medical imaging. *RG Grainger, DJ Allison Churchill Livingstone.-1997.-2626 p*, 2008.
- [7] Guy Pallardy and M Pallardy. Histoire abregee du radiodiagnostic et de l’imagerie medicale. *Histoire des sciences médicales*, 41(1):34, 2007.
- [8] Randolph R Resnik, Louis T Kircos, and Carl E Misch. Diagnostic imaging and techniques. *Contemporary Implant Dentistry. Missouri: Mosby*, pages 38–67, 2007.
- [9] Medically reviewed by university of illinois — written by brian krans — updated on december 11, 2019 <https://www.healthline.com/health/ultrasound>.
- [10] <https://www.independentimaging.com/abcs-imaging-difference-xray-ultrasound-mri-ct-scan/>.
- [11] Dr sébastien ronze radiologue ‘<https://www.doctissimo.fr/html/sante/imagerie/irm.htm>’.

-
- [12] Meng Li. Chest ct features and their role in covid-19. *Radiology of Infectious Diseases*, 7(2):51–54, 2020.
- [13] Daniel Berrar, Naoyuki Sato, and Alfons Schuster. Quo vadis, artificial intelligence? *Advances in Artificial Intelligence (16877470)*, 2010.
- [14] P Vincent. Modeles a noyaux a structure locale (french text). 2003.
- [15] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.
- [16] Runya Li and Shenglian Li. Multimedia image data analysis based on knn algorithm. *Computational Intelligence and Neuroscience*, 2022, 2022.
- [17] Samir Brahim Belhaouari, Shahnawaz Ahmed, and Samer Mansour. Optimized k-means algorithm. *Mathematical Problems in Engineering*, 2014, 2014.
- [18] A Al-Anazi and ID Gates. A support vector machine algorithm to classify lithofacies and model permeability in heterogeneous reservoirs. *Engineering Geology*, 114(3-4):267–277, 2010.
- [19] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3-4):197–387, 2014.
- [20] Nene Adama Dian DIALLO. La reconnaissance des expressions faciales. 2019.
- [21] Vartul Mittal. Top 15 deep learning applications that will rule the world in 2018 and beyond. URL: <https://medium.com/@vratulmittal/top-15-deeplearning-applications-that-will-rule-the-world-in-2018-andbeyond-7c6130c43b01>, 2017.
- [22] Priyanka Malhotra, Sheifali Gupta, Deepika Koundal, Atef Zaguia, and Wegayehu Enbeyle. Deep neural networks for medical image segmentation. *Journal of Healthcare Engineering*, 2022, 2022.
- [23] Tristan Stérin. Réseaux de neurones récurrents et mémoire: application à la musique. 2016.
- [24] Eduardo Soares, Plamen Angelov, Sarah Biaso, Michele Higa Froes, and Daniel Kanda Abe. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *MedRxiv*, 2020.
- [25] Karrar Hameed Abdulkareem, Salama A Mostafa, Zainab N Al-Qudsy, Mazin Abed Mohammed, Alaa S Al-Waisy, Seifedine Kadry, Jinseok Lee, and Yunyoung Nam. Automated system for identifying covid-19 infections in computed tomography images using deep learning models. *Journal of Healthcare Engineering*, 2022, 2022.

- [26] SV Kogilavani, J Prabhu, R Sandhiya, M Sandeep Kumar, UmaShankar Subramaniam, Alagar Karthick, M Muhibbullah, and Sharmila Banu Sheik Imam. Covid-19 detection based on lung ct scan using deep learning techniques. *Computational and Mathematical Methods in Medicine*, 2022, 2022.
- [27] Eduardo Soares, Plamen Angelov, Sarah Biaso, Michele Higa Froes, and Daniel Kanda Abe. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *MedRxiv*, 2020.
- [28] HT Do, V Raghavan, and G Yonezawa. Pixel-based and object-based terrace extraction using feed-forward deep neural network. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:1–7, 2019.
- [29] Kaung Si Thu and Weerakorn Ongsakul. Simulation of blockchain based power trading with solar power prediction in prosumer consortium model. In *2020 International Conference and Utility Exhibition on Energy, Environment and Climate Change (ICUE)*, pages 1–10. IEEE, 2020.
- [30] Himanshu Sharma. Activation functions: Sigmoid, relu, leaky relu and softmax basics for neural networks and deep learning, 2019.
- [31] Frans Van den Bergh and Andries P Engelbrecht. Cooperative learning in neural networks using particle swarm optimizers. *South African Computer Journal*, 2000(26):84–90, 2000.
- [32] Bradley J Erickson and Felipe Kitamura. Magician’s corner: 9. performance metrics for machine learning models. *Radiology: Artificial Intelligence*, 3(3), 2021.
- [33] A Nikhil Chandran, Karthik Sreekumar, and DP Subha. Eeg-based automated detection of schizophrenia using long short-term memory (lstm) network. In *Advances in Machine Learning and Computational Intelligence*, pages 229–236. Springer, 2021.
- [34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [35] Deng Pan, Lixian Sun, Rui Wang, Xingjian Zhang, and Richard O Sinnott. Deepfake detection through deep learning. In *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pages 134–143. IEEE, 2020.
- [36] 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*, 176:114883, 2021.
- [37] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

-
- [38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [39] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [40] Python : définition et utilisation de ce langage informatique <https://www.journaldunet.fr/web-tech/dictionnaire-duwebmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique>. 02/05/2022.
- [41] Tensorflow ,searchdatamanagement.techtarget.com/definition/tensorflow.01/05/2022.
- [42] Initiation au deep learning avec google colab <https://moov.ai/fr/blog/deep-learning-avec-google-colab>. 02/05/2022.
- [43] What is google colab <https://blog.education-ecosystem.com/what-is-google-colab>. 03/05/2022.
- [44] Anshel Julio Roman. *Künstliche Neuronale Netze in biomedizinischen Anwendungen*. PhD thesis, BTU Cottbus-Senftenberg, 2021.
- [45] F Chollet. others keras [internet]. *GitHub*, 2015.
- [46] About keras <https://keras.io/about>.06/05/2022.
- [47] Numpy <https://numpy.org/doc/stable/user/whatisnumpy.html>.06/05/2022.
- [48] Matplotlib <https://matplotlib.org> .07/05/2022.
- [49] <https://opencv.org/>.09/05/2022.
- [50] Scikit-learn machine learning in python <https://scikit-learn.org/stable/>09/05/2022.
- [51] <https://pandas.pydata.org/>.09/05/2022.
- [52] Eduardo Soares, Plamen Angelov, Sarah Biaso, Michele Higa Froes, and Daniel Kanda Abe. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *MedRxiv*, 2020.
- [53] Ranjit Varma and Devendra Bharadwaj. Introduction to machine learning through hands-on in python. In *Transformation in Healthcare with Emerging Technologies*, pages 261–285. Chapman and Hall/CRC, 2022.
- [54] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung

- image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.
- [55] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018.
- [56] Jaime Melendez, Bram Van Ginneken, Pragnya Maduskar, Rick HHM Philipsen, Klaus Reither, Marianne Breuninger, Ifedayo MO Adetifa, Rahmatulai Maane, Helen Ayles, and Clara I Sánchez. A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest x-rays. *IEEE transactions on medical imaging*, 34(1):179–192, 2014.