



République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

# Mémoire de Master

## en Informatique

*Spécialité : Génie des Systèmes Informatique*

## Thème

---

Conception d'un système automatique d'aide au  
diagnostic du Covid-19

---

Encadré par

— CHIHATI SARAH

Réalisé par

— BELKASSAM MOHAMMED NAZIH

— AISSAT MOHAMMED ABDELLAH

Soutenu publiquement, le 02/07/2022, devant le jury composé de :

- Président : M. ZERROUKI Taha.

- Examineur 1 : Mme. BRAHIMI Farida.

- Examineur 2 : M. DEMMOUCHE Mouloud.

- Encadreur : Mme. CHIHATI Sarah.

2021/2022

# *Remerciements*

Au-dessus de tout, nous remercions Dieu tous puissant, pour nous avoir donné la force et la volonté d'accomplir ce modeste travail .

Tout d'abord, nous tenons particulièrement à remercier notre encadrante :Mme CHIHATI Sarah pour avoir encadrée et dirigée ce travail et pour son attention, son don d'écoute, sa tolérance et sa disponibilité, nous lui témoignons toute notre reconnaissance pour nous avoir guidée et prodiguée des conseils sur le plan scientifique que sur le plan humain.

Nous tenons à remercier les membres du jury de nous avoir fait l'honneur de juger notre travail.

Nous tenons également à remercier l'administration du département avec tous ses employés et enseignants.

A nos chers parents, de leur confiance, encouragement et de leur sacrifice durant toute notre vie.

# *Dédicaces*

Je dédie ce travail à L'âme pure de ma tante, **qu'ALLAH** l'accueille dans son vaste paradis.

À ma très **chère mère** ainsi que mon père **Abdelnasser**  
Qui ont su être à mes côtés dans les moments difficile sans lesquels je n'aurai pas abouti à ce stade d'étude, ainsi que chaque instant de bonheur qu'ils m'ont procuré, fasse Dieu que je puisse les honorer.

Le projet de fin de qu'on cycle a l'honneur de présenté, a été exécuté à mené à bien grâce à la participation exceptionnelle de notre encadrement.

À mes chers frères : **Nidhal** et **Houssam**.

À ma **chère sœur** . À toute **ma famille**.

À mon ami et binôme **Mohamed Aissat** qui était mon frère durant tout le cycle d'étude.

À tous mes amis : Youcef, Hamza, Adel, Larbi... et à toute ma promotion et mon groupe de master : Smail, Hassan, Fares, Madjid, Islam, Lina, Maroua, Asma...

Enfin à tous ceux qui me reconnaissent...

*Belkassam Mohammed Nazih*

# *Dédicaces*

Je dédie ce modeste travail accompagné d'un profond amour :

À celle qui m'a arrosé de tendresse et d'espoirs, à la source d'amour **ma mère**.

À mon supporté dans ma vie, qui m'a appris mon support et ma dirige vers la gloire  
mon père **said**.

À mes chère frères et sœurs : Halim, Aissa, Nassima , Habiba, Sameh, Amine.

À toutes les personnes de ma grande famille.

À mon binôme : **Nazih**.

À mes meilleurs amis : Hassan, Madjid, Fares, Smail, Lyna, Islam, Maroua, lina, asma,  
Fateh, Adel, Ayoub.

À tous les amis de ma promotion master 2.

À tous ceux qui tiennent une place dans mon coeur.

*Aissat Mohammed Abdellah*

# Résumé

Le CORONA Virus ou COVID-19 est une maladie très dangereuse, en raison de son développement rapide et de sa propagation entre les personnes, ce qui a fait beaucoup souffrir la plupart des pays, afin de limiter sa propagation, la difficulté de son diagnostic en raison du grand nombre d'infections, diminuer le temps de donner les résultats, et les médecins peuvent faire des erreurs en raison du grand nombre de tests, nous avons proposé d'évaluer un système de diagnostic de la maladie COVID-19.

À travers ce travail, nous présenterons une solution basée sur l'apprentissage profond utilisant des réseaux de neurones convolutifs qui classent les images médicales thoraciques en trois catégories (cas normaux, cas COVID, cas de pneumonie).

Notre solution est basée sur l'utilisation de deux approches, la première basée sur l'utilisation de l'apprentissage par transfert sur certains réseaux de neurones convolutifs pré-entraînés et en extrayant les caractéristiques à partir d'images radiographiques thoraciques. Nous avons également utilisé un modèle de concaténation entre deux modèles. Ensuite, nous avons étudié l'impact de la taille de la base. Et la deuxième approche basée sur Vision Transformer.

Les modèles proposés ont donné de bons résultats. Là où la première étape de la première approche a donnée les meilleurs résultats, alors que la deuxième étape nous avons constaté une amélioration des résultats avec l'augmentation de la taille de la base de pré-entraînement, quant à la deuxième approche, elle a donné des résultats encourageants.

**Mots-clés :** apprentissage par transfert, apprentissage profond, Vision Transformer, COVID-19, pneumonie, réseau de neurones convolutifs, classification, images radiographiques thoraciques.

## abstract

CORONA Virus or COVID-19 is a very dangerous disease, due to its rapid development and spread between people, which caused most countries to suffer a lot, in order to limit its spread, the difficulty of its diagnosis in due to the large number of infections, decrease the time to give the results, and doctors can make errors due to the large number

of tests, we proposed to evaluate a diagnostic system for COVID-19 disease.

Through this work, we will present a solution based on deep learning using convolutional neural networks that classify chest medical images into three categories (normal cases, COVID cases, pneumonia cases).

Our solution is based on the use of two approaches, the first based on using transfer learning on some pre-trained convolutional neural networks and extracting features from chest x-ray images. We also used a concatenation model between two models. Next, we studied the impact of the size of the base. And the second approach based on Vision Transformer.

The models proposed gave good results. Where the first stage of the first approach gave the best results, while the second stage we saw an improvement in results with the increase in the size of the pre-training base, as for the second approach, it has yielded encouraging results.

**Keywords :** transfer Learning, Deep Learning, Vision Transformer, COVID-19, pneumonia, convolutional neural network, classification, chest X-ray images.

## الملخص

يعد فيروس كورونا أو كوفيد-19 من الأمراض الخطيرة جدًا، نظرًا لتطوره السريع وانتشاره بين الناس مما تسبب في معاناة معظم الدول، من أجل الحد من انتشاره و صعوبة تشخيصه بسبب كثرة عدد حالات الإصابة به وكثرة العدوى، وتقليل الوقت لإعطاء النتائج ، حيث يمكن للأطباء ارتكاب أخطاء بسبب العدد الكبير من الاختبارات ، اقترحنا تقييم نظام تشخيص لمرض كوفيد-19.

من خلال هذا العمل ، سنقدم حلاً يعتمد على التعلم العميق باستخدام الشبكات العصبية التلافيفية التي تصنف الصور الطبية للصدر إلى ثلاث فئات (الحالات العادية ، وحالات الكوفيد ، وحالات الالتهاب الرئوي).

يعتمد حلنا على استخدام نهجين، الأول يعتمد على استخدام التعلم بالنقل على بعض الشبكات العصبية التلافيفية المدربة مسبقاً واستخراج الميزات من صور الأشعة السينية للصدر و استخدمنا أيضاً نموذج التسلسل بين نموذجين. بعد ذلك، درسنا تأثير حجم قاعدة البيانات، أما النهج الثاني يعتمد على محول الرؤية.

أعطت النماذج التي تم إنشاؤها نتائج جيدة. حيث أعطت المرحلة الأولى من النهج الأول أفضل النتائج، بينما شهدت المرحلة الثانية تحسناً في النتائج مع زيادة حجم قاعدة ما قبل التدريب، أما بالنسبة للنهج الثاني فقد أسفر عن نتائج مشجعة.

الكلمات الرئيسية: نقل التعلم ، التعلم العميق ، محول الرؤية ، كوفيد-19 ، الالتهاب الرئوي ، الشبكة العصبية التلافيفية ، التصنيف ، صور الصدر بالأشعة السينية.

# Table des matières

<b>Table des matières</b>	<b>ii</b>
<b>Table des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Liste des abréviations</b>	<b>x</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Présentation du COVID-19</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Définition du Coronavirus (COVID-19) . . . . .	3
1.3 Comparaison entre la COVID-19 et la grippe (influenza) . . . . .	4
1.4 Transmission de la COVID-19 . . . . .	6
1.5 Les symptômes de la COVID-19 . . . . .	6
1.5.1 Symptômes les plus courants . . . . .	6
1.5.2 Symptômes moins courants . . . . .	6
1.5.3 Symptômes graves . . . . .	6
1.6 La pneumonie . . . . .	7
1.6.1 Définition de la pneumonie . . . . .	7
1.6.2 Les types de la pneumonie . . . . .	7
1.7 Diagnostic de la COVID-19 . . . . .	8
1.7.1 Les tests par PCR . . . . .	8
1.7.2 Les tests Sérologiques . . . . .	9

1.7.3	Les tests antigéniques . . . . .	9
1.7.4	Imagerie médicale . . . . .	10
1.8	Conclusion . . . . .	12
<b>2</b>	<b>Deep Learning et Travaux Connexes</b>	<b>14</b>
2.1	Introduction . . . . .	14
2.2	Intelligence Artificielle (IA) . . . . .	14
2.3	Apprentissage automatique (Machine Learning) . . . . .	15
2.3.1	Les types de l'apprentissage automatique . . . . .	15
2.4	Apprentissage profond (Deep Learning) . . . . .	18
2.4.1	Définition . . . . .	18
2.4.2	Domaine d'applications . . . . .	18
2.4.3	Historique . . . . .	18
2.4.4	Différence entre l'apprentissage automatique et l'apprentissage pro- fond . . . . .	20
2.4.5	Les réseaux de neurones convolutifs . . . . .	21
2.4.6	Le problème du sur-apprentissage (OVERFITTING) . . . . .	30
2.5	Base de données (DataSet) . . . . .	31
2.6	Les travaux connexes pour le diagnostic de COVID-19 . . . . .	32
2.7	Conclusion . . . . .	36
<b>3</b>	<b>Approches et solutions proposées</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Vue globale de notre système . . . . .	37
3.3	Présentation détaillée de notre système . . . . .	38
3.3.1	Base de données . . . . .	38
3.3.2	Prétraitement de la base de données . . . . .	40
3.3.3	Modèle de classification . . . . .	43
3.4	Conclusion . . . . .	51
<b>4</b>	<b>Résultats et discussions</b>	<b>52</b>
4.1	Introduction . . . . .	52
4.2	Environnement de travail . . . . .	52
4.2.1	Le langage de programmation . . . . .	52

4.2.2	Les bibliothèques utilisées . . . . .	53
4.2.3	La plateforme et environnements de développement . . . . .	54
4.3	Indicateurs de performances de la classification . . . . .	55
4.3.1	Matrice de confusion . . . . .	55
4.4	Résultats et discussion . . . . .	56
4.4.1	comparaison de performances entre notre modèle et autres travaux connexes . . . . .	67
4.5	Prédiction . . . . .	68
4.6	Conclusion . . . . .	69
	<b>Conclusion générale et perspectives</b>	<b>70</b>
	<b>Bibliographie</b>	<b>72</b>

# Table des figures

1.1	Les composants du coronavirus (COVID-19) [4]. . . . .	4
1.2	test PCR [13]. . . . .	8
1.3	test sérologie [13]. . . . .	9
1.4	test antigénique [13] . . . . .	10
1.5	La radiographie thoracique (Chest X-ray) [16]. . . . .	11
1.6	Radiographie thoracique (Une personne A : cas normal, Une personne B : cas de COVID) [18]. . . . .	11
1.7	La tomodensitométrie (CT Scan) cas de COVID-19 [20]. . . . .	12
2.1	Différents types de Machine Learning . . . . .	15
2.2	La classification et la régression [24] . . . . .	16
2.3	Apprentissage supervisé. . . . .	16
2.4	Apprentissage non supervisé. . . . .	17
2.5	Apprentissage semi supervisé [26]. . . . .	17
2.6	Apprentissage par renforcement [28]. . . . .	18
2.7	la relation entre IA, ML et DL. . . . .	20
2.8	Les différentes couches de CNN. . . . .	22
2.9	La couche de convolution [33]. . . . .	22
2.10	Exemple d'application de pas "stride" [34]. . . . .	23
2.11	Exemple d'application de rembourrage [36]. . . . .	24
2.12	La couche ReLU [32]. . . . .	24
2.13	Types de Pooling [39]. . . . .	25
2.14	Couche entièrement connectée (FC) [38]. . . . .	26

2.15 Description du L'abandon [49]. . . . .	31
3.1 Représentation de l'architecture globale de notre système. . . . .	38
3.2 Diagramme global de notre base de données. . . . .	39
3.3 code python pour la division des données. . . . .	40
3.4 code python utilisé pour le traitement (n-clahe et gamma) . . . . .	41
3.5 exemple de traitement d'image avec N-CLAHE ET GAMMA. . . . .	42
3.6 augmentation de données. . . . .	42
3.7 Notre architecture basé sur ResNet50. . . . .	44
3.8 Notre architecture basé sur DenseNet121. . . . .	45
3.9 Notre architecture basé sur Xception. . . . .	46
3.10 Notre architecture basé sur Xception concatiné avec ResNet50V2. . . . .	47
3.11 Notre architecture basé sur Big Transfer. . . . .	48
3.12 la méthode de redimensionnement d'image en fonction de CNN. . . . .	49
3.13 la division d'image en patches. . . . .	49
3.14 L'architecture de l'encodeur du transformateur. . . . .	50
3.15 l'architecture globale de le Vision transformer (ViT). . . . .	51
4.1 Matrice de confusion. . . . .	55
4.2 Aperçu sur les résultats initiaux et finals de chaque modèle. . . . .	57
4.3 Courbes accuracy et loss obtenus pour le modèle ResNet50. . . . .	58
4.4 Matrice de confusion pour le modèle ResNet50. . . . .	58
4.5 Courbes accuracy et loss obtenus pour le modèle DenseNet121. . . . .	59
4.6 Matrice de confusion pour le modèle DenseNet121. . . . .	59
4.7 Courbes accuracy et loss obtenus pour le modèle Xception. . . . .	60
4.8 Matrice de confusion pour le modèle Xception. . . . .	60
4.9 Courbes accuracy et loss obtenus pour le modèle concaténation (Resnet50 + Xception). . . . .	61
4.10 Matrice de confusion pour le modèle concaténation (Resnet50 + Xception). . . . .	61
4.11 Courbes accuracy et loss obtenus pour le modèle BiT avec ImageNet1k. . . . .	62
4.12 Matrice de confusion pour le modèle BiT avec ImageNet1k. . . . .	63
4.13 Courbes accuracy et loss obtenus pour le modèle BiT avec ImageNet21k. . . . .	63
4.14 Matrice de confusion pour le modèle BiT avec ImageNet21k. . . . .	64

4.15	Courbes accuracy et loss obtenus pour le modèle ViT. . . . .	65
4.16	Matrice de confusion pour le modèle ViT. . . . .	65
4.17	comparaison entre non modèles. . . . .	67
4.18	Résultat de teste de notre modèle sur quelques exemple de teste avec l'utilisation de Grad-CAM. . . . .	69

# Liste des tableaux

1.1	la comparaison entre le COVID-19 et l'influenza [6]. . . . .	5
2.1	L'historique de l'apprentissage profond [30] . . . . .	19
2.2	La différence Entre l'apprentissage automatique et l'apprentissage profond.	21
2.3	Les types de pooling [38]. . . . .	25
2.4	Les architectures très connues de CNN. . . . .	28
2.5	Méthodes d'optimisation les plus utilisées [48]. . . . .	29
2.6	Exemples de bases de données utilisées pour le diagnostic du COVID-19. .	32
2.7	Comparaison des travaux connexes pour le diagnostic de COVID-19 . . .	36
3.1	représentation de données. . . . .	40
4.1	Métrique d'évaluation de la classification. . . . .	56
4.2	Mesure de performance du modèle ResNet50. . . . .	58
4.3	Mesure de performance du modèle DenseNet121. . . . .	59
4.4	Mesure de performance du modèle Xception. . . . .	60
4.5	Mesure de performance du modèle concaténation (Resnet50 + Xception). .	61
4.6	comparaison entre les modèles d'étape 1. . . . .	62
4.7	Mesure de performance du modèle BiT avec ImageNet1k. . . . .	63
4.8	Mesure de performance du modèle BiT avec ImageNet21k. . . . .	64
4.9	comparaison entre les modèles d'étape 2. . . . .	64
4.10	Mesure de performance du modèle ViT. . . . .	65
4.11	comparaison entre non modèles. . . . .	66

4.12 Comparaison entre notre modèle proposé (DenseNet121 + Xception) et un  
travail de littérature. . . . . 68

# Liste des abréviations

Adam :	Adaptive Moment Estimation.
AI/IA :	Intelligence Artificielle.
API :	Application Programming Interface.
BiT :	Big Transfer.
CNN :	Convolutional Neural Network.
CPU :	Central Processing Unit.
CT :	Computed Tomography.
DL :	Deep Learning.
FC :	Fully Connected.
FN :	False Negative.
FP :	False Positive.
GPU :	Graphical Processing Unit.
ML :	Machine Learning.
MLP :	MultiLayer Perceptron.
OpenCv :	Open Computer Vision.
PCam :	Patch Camelyon.
PCR :	Polymerase Chain Reaction.
PNG :	Portable Network Graphics.
Pre :	Précision.
ReLU :	Rectified Linear Unit.
RN :	Réseaux de neurones.

- Se : Sensibility.  
Sen : Sensitivité.  
SGD : Stochastic Gradient Descent.  
Sp : Specificity.  
TDM : La tomodensitométrie.  
TN : Correct Negative.  
TP : Correct Positive.  
TRA : Test Rapide d'Antigene.  
ViT : Vision Transformer.

# Introduction générale

La nouvelle maladie coronavirus, appelée COVID-19 par l'Organisation Mondiale de la Santé, est causée par une nouvelle classe de coronavirus appelée SARS-CoV-2 (syndrome respiratoire aigu sévère coronavirus 2). Ce virus à ARN simple brin (acide ribonucléique) peut provoquer de graves infections respiratoires pouvant entraîner une hospitalisation et la mort. Près de 538 millions de personnes ont été infectées dans le monde et 6,32 millions sont décédées.

Les scientifiques travaillent actuellement avec diligence sur des thérapies et des vaccins pour protéger la population générale du COVID-19. Jusqu'à ce que leurs efforts portent leurs fruits, l'une des meilleures solutions a été de détecter le virus à ses débuts, puis d'isoler les personnes infectées par des quarantaines pour empêcher la propagation de la maladie. Le test de réaction en chaîne par polymérase de transcription inverse en temps réel utilisant des écouvillons nasopharyngés mesure les niveaux d'ARN dans le corps et a été utilisé pour un diagnostic plus précis de COVID-19.

Cependant, les tests prennent des heures et les retards peuvent entraîner des temps d'attente plus longs. Une méthode meilleure et plus précise pour diagnostiquer le COVID-19 consiste à utiliser les rayons X (X-ray) et la tomодensitométrie (TDM). Elle est réalisée via des opacités vitreuses multifocales bilatérales (GGO) ou une fusion avec le septum interlobulaire et un épaissement vasculaire, principalement dans les champs périphériques des poumons.

Les moyens et solutions mentionnés ci-dessus constituent une charge pour les médecins et les spécialistes car ils les obligent à examiner manuellement plusieurs images par jours, et cette méthode peut être lente, ardue et fatigante et peut également impliquer des erreurs humaines en raison du grand nombre de malades.

Notre travail vise à résoudre ce problème en construisant un modèle d'apprentissage en profondeur qui détecte automatiquement le Covid-19 dans des images radiographiques médicales. Le modèle proposé doit, à partir d'une image de radiographie thoracique donnée en entrée, prédire si le patient est sain ou infecté par le Covid-19 ou la pneumonie.

**Organisation du manuscrit** Notre mémoire est organisé comme suit :

- **Chapitre 1** : Dans ce chapitre, nous définirons le virus Corona, ses symptômes et ses méthodes de diagnostic.
- **Chapitre 2** : Dans ce chapitre, nous définirons quelques notions de base sur l'apprentissage profond, puis nous présenterons quelques travaux connexes.
- **Chapitre 3** : Dans ce chapitre nous présenterons notre base de données ainsi que les solutions que nous avons proposées.
- **Chapitre 4** : Ce chapitre est composé de deux parties :
  - Dans la première partie, nous présenterons les outils de programmation et de l'implémentation.
  - Dans la deuxième partie, nous discuterons des résultats obtenus par nos modèles.

Enfin, nous terminons ce travail par une conclusion générale.

# Présentation du COVID-19

## 1.1 Introduction

Les coronavirus sont une vaste famille de virus susceptibles de provoquer des maladies diverses chez l'homme, allant d'un simple rhume au syndrome respiratoire du Moyen-Orient (MERS) et au syndrome respiratoire aigu sévère (SRAS).

Un nouveau coronavirus (COVID-19) a été identifié en 2019 à Wuhan, en Chine.

Le SRAS-CoV-2 et la maladie COVID-19 qui en résulte sont l'un des plus grands défis du 21e siècle. En ce moment, environ 409 millions de personnes ont été testées positives et 5,8 millions de personnes en sont décédées, alors qu'en Algérie selon les dernières statistiques de la presse nationale, environ 260 mille cas positifs de COVID-19 et 6,7 mille décès [1].

Dans ce chapitre, nous définirons la COVID-19 et ses symptômes. Nous présenterons les méthodes de diagnostic également l'imagerie médicale.

## 1.2 Définition du Coronavirus (COVID-19)

Est un groupe de virus qui causent des maladies chez les mammifères et les oiseaux. Le virus provoque des infections respiratoires chez l'homme notamment des rhumes qui peuvent être mortelles chez certaines personnes. Depuis le début de ce siècle, trois coronavirus (CoV) ont causé de graves maladies respiratoires humaines, notamment le syndrome respiratoire aigu sévère (SRAS), le syndrome respiratoire du Moyen-Orient (MERS) et la COVID-19 (Coronavirus Disease 2019) [2].

Vu sa propagation rapide la COVID-19 est devenu une pandémie il n'est plus une épidémie, de ce fait il ne faut pas confondre entre une pandémie qui est une maladie illimitée dans le temps et dans l'espace dont la maladie dépasse les frontières ex : grippe H1N1, et une épidémie qui est limitée dans le temps et dans l'espace ex : MTH (maladies à transmission hydrique (choléra)) [3].

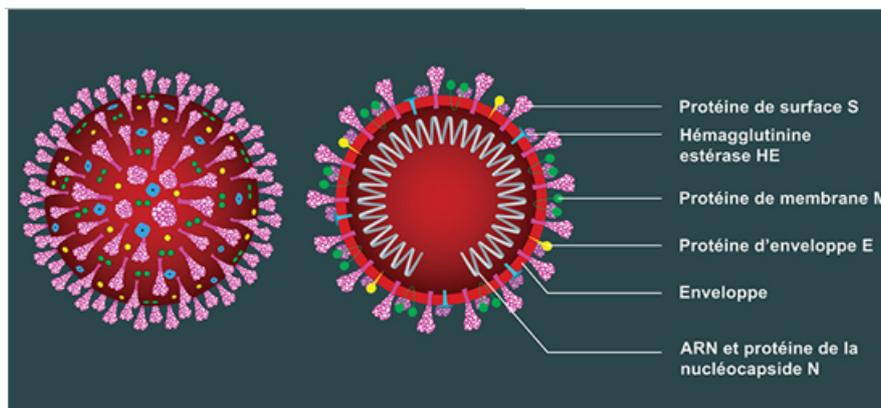


FIGURE 1.1 – Les composants du coronavirus (COVID-19) [4].

### 1.3 Comparaison entre la COVID-19 et la grippe (influenza)

La grippe est une maladie contagieuse fréquente. Ses principaux symptômes sont : la fièvre, les courbatures, la toux sèche et la grande fatigue.

La COVID-19 et la grippe sont des maladies respiratoires infectieuses et partagent certains symptômes. Cependant, ils sont causés par différents virus et il existe des différences entre les groupes de personnes qui sont plus susceptibles de contracter des formes graves les uns des autres. Et bien sûr, il existe également plusieurs différences dans la façon de traiter lequel d'entre eux. Il est important de connaître la différence entre la COVID-19 et la grippe pour mieux se protéger [5].

La différence entre COVID-19 et grippe saisonnière se résume à quatre points principaux :

1. Le taux de reproduction du SARS-CoV-2 est 2 à 5 fois supérieur à celui du virus de la grippe. Cela indique que les personnes infectées par le COVID-19 infecteront en moyenne 2 à 5,7 personnes, et que la grippe saisonnière infectera 1 à 3 personnes.
2. Le taux de létalité de COVID-19 est environ 10 à 20 fois plus élevé que celui de la grippe saisonnière.
3. Le temps d'incubation du COVID-19 varie de 2 à 14 jours, ce qui est beaucoup plus long que la grippe saisonnière.
4. Le taux d'hospitalisation pour COVID-19 est environ 3 à 105 fois supérieur à celui de la grippe saisonnière, car le COVID-19 entraîne généralement des complications telles que la pneumonie. La figure suivante résume ce que nous venons de dire :

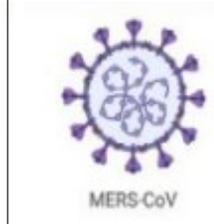
Maladie	Maladie à coronavirus 2019 (COVID-19)	Grippe saisonnière	Syndrome respiratoire aigu sévère (SRAS)	Syndrome respiratoire du Moyen-Orient (MERS)
Virus causant la maladie				
Taux de reproduction (R0)	2 à 5,7	1,3	3	0,3 à 0,8
Taux de létalité	Autour de 1%	0,05 à 0,1%	9,6 à 11%	34,4%
Temps d'incubation	2 à 14 jours	1 à 4 jours	2 à 7 jours	6 jours
Taux d'hospitalisation	6 à 19%	2%	La plupart des cas 10 à 60%	La plupart des cas 4 à 13%
Taux potentiel d'infection communautaire	35 à 80%	10 à 20%		

TABLE 1.1 – la comparaison entre le COVID-19 et l'influenza [6].

## 1.4 Transmission de la COVID-19

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

- Des gouttelettes respiratoires sortent lorsqu’une personne infectée tousse ou éternue ou même parle.
- Contact étroit et prolongé entre personnes (plus de 15 minutes, moins de 2 mètres).
- Contact indirect, que ce soit par des surfaces ou les objets contaminé [7].

## 1.5 Les symptômes de la COVID-19

La COVID-19 affecte les gens de différentes manières. La plupart des personnes infectées développent une forme légère à modérée de la maladie et se rétablissent sans hospitalisation.

### 1.5.1 Symptômes les plus courants

- Fièvre.
- Toux.
- Fatigue.
- perte du goût ou de l’odorat [8].

### 1.5.2 Symptômes moins courants

- maux de gorge.
- maux de tête.
- courbatures et douleurs.
- diarrhée.
- éruption cutanée ou décoloration des doigts ou des orteils.
- rougeur ou irritation des yeux [8].

### 1.5.3 Symptômes graves

- difficultés respiratoires ou essoufflement.
- perte de la parole, difficultés à se déplacer ou confusion.

- douleurs thoraciques [8].

## 1.6 La pneumonie

### 1.6.1 Définition de la pneumonie

La pneumonie est une inflammation des sacs aériens dans les poumons en réponse à une blessure, comme une infection. Lorsque les voies respiratoires sont également atteintes, on parle alors de bronchopneumonie. La pneumonie peut être dans une zone d'un poumon ou être dans plusieurs zones (pneumonie « double » ou « multilobaire »). Beaucoup de choses peuvent causer une pneumonie, bien que le plus souvent elles soient infectieuses. Le diagnostic de la pneumonie est presque comme le diagnostic de la COVID [9].

**Les symptômes de la pneumonie bactérienne comprennent :**

- Toux et fièvre.
- Essoufflement qui s'aggrave avec l'activité.
- Fatigue extrême.
- Transpiration abondante [10].

### 1.6.2 Les types de la pneumonie

- **Pneumonie virale** : Ce type est causé par divers virus, dont la grippe (influenza), et est responsable d'environ un tiers de tous les cas de pneumonie. Les gens peuvent être plus susceptibles de contracter une pneumonie bactérienne s'ils ont d'abord une pneumonie virale [10].

- **Pneumonie bactérienne** : Ce type est causé par diverses bactéries. Le plus commun est *Streptococcus pneumoniae*. Cela se produit généralement lorsque le corps est affaibli d'une manière ou d'une autre, par exemple par une maladie, une mauvaise alimentation, la vieillesse ou une immunité altérée, et que les bactéries sont capables de se frayer un chemin dans les poumons [10].

- **Pneumonie à mycoplasme** : Aussi connu sous le nom de pneumonie de la marche, ce type est considéré comme atypique car ses symptômes et ses signes physiques peuvent varier. En outre, il provoque une maladie plus bénigne. Il survient le plus souvent chez les enfants plus âgés et les jeunes adultes [11].

## 1.7 Diagnostic de la COVID-19

Le personnel médical a prélevé des échantillons de frottis nasopharyngés, de crachats, de sécrétions des voies respiratoires inférieures, de sang et d'excréments de patients et a détecté l'ARN de nouveaux coronavirus dans les échantillons par réaction en chaîne par polymérase en transcription inverse (RT-PCR). Le séquençage génétique peut également être effectué sur des échantillons à diagnostiquer en comparant la séquence de nouvelles séquences connues du génome du coronavirus.

La détection des acides nucléiques et la détection des gènes viraux peuvent également retracer l'origine du virus dans la recherche scientifique et guider la prévention et le contrôle de nouveaux agents pathogènes à l'avenir [12].

### 1.7.1 Les tests par PCR

La PCR ou RT-PCR SARS-CoV 2 (désignation équivalente pour Reverse Transcriptase-Polymerase Chain reaction) est une technique de laboratoire permettant de détecter le matériel génétique d'un virus dans un échantillon. Il nécessite un prélèvement aux endroits les plus accessibles où la concentration de virus est la plus élevée, c'est-à-dire le nasopharynx, derrière les conduits nasaux [13].



FIGURE 1.2 – test PCR [13].

### 1.7.2 Les tests Sérologiques

Le test sérologique COVID-19 contient un test sanguin et sa fonction est de tester les anticorps contre le virus. Ces anticorps apparaissent quelques jours ou semaines après l'infection par le SRAS-CoV-2. Il existe deux classes d'anticorps : les IgM d'apparition précoce, et les IgG, d'apparition un peu plus tardive [13].



FIGURE 1.3 – test sérologie [13].

### 1.7.3 Les tests antigéniques

Le test rapide d'antigène (TRA) recherche la présence de protéines du virus Sars-Cov-2 dans l'échantillon et nécessite le même prélèvement nasopharyngé que le test PCR mais nécessite une technique d'analyse simplifiée.

Le test antigénique consiste en un test manuel, avec une lecture directement sur une petite boîte en plastique, à la manière des tests de grossesse vendus en pharmacie. Le résultat s'affiche dans un délai maximum de 30 minutes sous forme de barres qui apparaissent sur la boîte [13].



FIGURE 1.4 – test antigénique [13]

#### 1.7.4 Imagerie médicale

L'imagerie médicale est la pratique qui consiste à examiner l'intérieur du corps humain sans chirurgie. L'imagerie médicale est utilisée à des fins cliniques pour le diagnostic de nombreuses maladies telles que le cancer et l'épilepsie [14], grâce à des images médicales issues des champs magnétiques, de la radioactivité, des ultrasons et des rayons lumineux [15], et elle permet également de visualiser l'infection causée par la COVID-19 de deux techniques : la radiographie thoracique et le scanner thoracique (tomodensitométrie).

##### **Radiographie thoraciques (CHEST X-RAY)**

La radiographie thoracique montre les poumons, le cœur, les os et les vaisseaux sanguins. La radiographie consiste à projeter des rayons X à travers le corps pour créer des images sur un tableau de capture numérique. La quantité de rayonnement absorbée varie selon les différentes parties du corps. Ceux qui absorbent le plus de rayonnement, comme les os, apparaissent blancs sur les rayons X. Les parties qui absorbent moins de rayonnement (comme le cœur) apparaissent en différentes nuances de gris. Les zones qui contiennent de l'air, comme les poumons, sont représentées en noir. Les images radiographiques peuvent révéler une hypertrophie du cœur (augmentation du volume du cœur) et un œdème pulmonaire (liquide dans les poumons).

Face aux difficultés de réaliser des tests PCR en nombre et d'obtenir des résultats rapidement, les radiographies thoraciques sans injection se sont révélées être une solution alternative fiable pour poser le diagnostic du COVID-19 [17].



FIGURE 1.5 – La radiographie thoracique (Chest X-ray) [16].

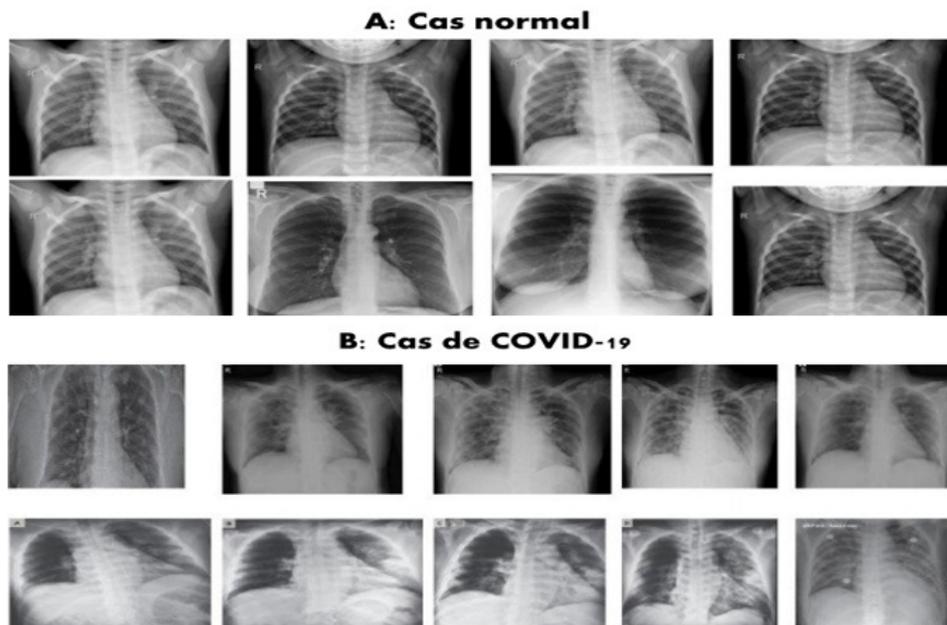


FIGURE 1.6 – Radiographie thoracique (Une personne A : cas normal, Une personne B : cas de COVID) [18].

## Scanner thoracique tomodensitométrie - TDM (CT SCAN)

Une tomodensitométrie (TDM) est un appareil à rayons X couplé à un ordinateur. Il permet d'obtenir des images en coupe de différentes parties du corps ainsi que de visualiser des tissus de différentes densités tels que les poumons, les os, les tissus mous ou les vaisseaux sanguins. Il est utilisé pour diagnostiquer les cancers, les traumatismes, les maladies cardiovasculaires, infectieuses et l'arthrose [19].

Le scanner thoracique est plus sensible pour le diagnostic précoce de la COVID-19 que la radiographie thoracique car :

- La radiographie thoracique peut rester normale pendant 4 à 5 jours après le début des symptômes.
- La scanner tomodensitométrie montre un schéma typique d'opacité et de consolidation du verre dépoli.

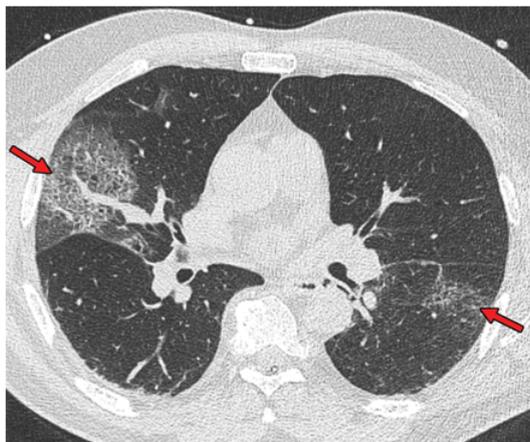


FIGURE 1.7 – La tomodensitométrie (CT Scan) cas de COVID-19 [20].

## 1.8 Conclusion

Dans ce chapitre nous avons défini la COVID-19 comme maladie qui affecte les individus de différentes manières. La plupart des personnes infectées développent une forme légère à modérée de la maladie et guérissent sans hospitalisation.

Face aux difficultés pour réaliser des tests PCR et confirmer le diagnostic du COVID-19 et d'obtenir rapidement les résultats, les examens radiologiques sont considérés comme so-

lution alternative fiable pour établir un diagnostic COVID-19, en complément de l'examen clinique, néanmoins le diagnostic de cette maladie reste toujours difficile par manque de moyens de diagnostic et par absences de signes spécifiques, de ce fait il est nécessaire d'automatiser le diagnostic de ce virus afin d'éliminer les faux négatifs, d'obtenir un diagnostic de positif et de réduire les dommages causés par la pandémie.

Dans le chapitre suivant, nous allons présenter l'apprentissage automatique plus spécifiquement l'apprentissage profond ainsi que ses différentes méthodes d'apprentissage et quelques travaux connexes.

# Deep Learning et Travaux Connexes

## 2.1 Introduction

L'apprentissage profond (deep learning) est un type d'intelligence artificielle dérivé de l'apprentissage automatique, où la machine est capable d'apprendre par elle-même, contrairement à la programmation classique où il est garanti que les règles prédéfinies seront mises en œuvre littéralement. Dans ce chapitre, nous étudierons les bases et les principes des réseaux de neurones convolutifs (CNN), et nous nous concentrerons sur :

- l'apprentissage profond et ces différentes applications.
- Les différentes couches qui composent un réseau CNN.
- Les techniques utilisées pour améliorer la performance des CNN.
- Les travaux connexes pour le diagnostic de COVID-19.

## 2.2 Intelligence Artificielle (IA)

L'intelligence artificielle vise à simuler la façon dont l'esprit humain prend des décisions. L'intelligence artificielle implique la mise en œuvre d'un certain nombre de technologies qui permettent aux machines de simuler une certaine forme d'intelligence réelle. L'intelligence artificielle se compose de plusieurs domaines, dont les plus connus sont l'apprentissage automatique (Machine Learning) et l'apprentissage profond (Deep Learning).

## 2.3 Apprentissage automatique (Machine Learning)

L'apprentissage automatique est une catégorie d'intelligence artificielle dont l'idée est de faire en sorte que les ordinateurs s'améliorent automatiquement grâce à l'expérience. [21], En d'autres termes, une technique visant à utiliser des algorithmes qui permettent aux ordinateurs de modifier leurs actions afin d'améliorer les procédures pour atteindre une plus grande précision [22].

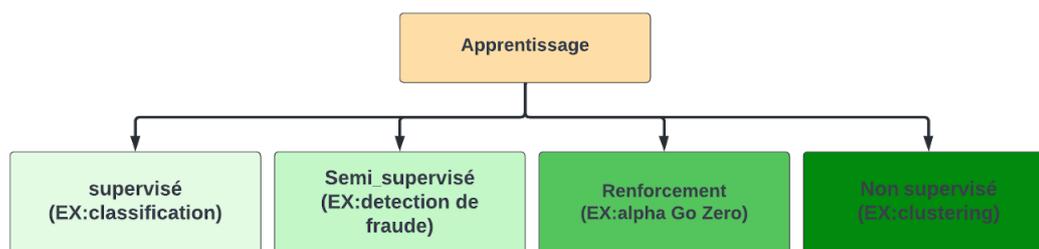


FIGURE 2.1 – Différents types de Machine Learning

### 2.3.1 Les types de l'apprentissage automatique

#### Apprentissage Supervisé

L'apprentissage supervisé est une approche de la Machine Learning qui se définit par l'utilisation d'ensembles de données étiquetées. Ces ensembles de données sont conçus pour former les algorithmes afin qu'ils classent les données ou prédisent les résultats avec précision. En utilisant des entrées et des sorties étiquetées, le modèle peut mesurer sa précision et apprendre au fil du temps [23].

L'apprentissage supervisé peut être séparé en deux types de problèmes lors de l'exploitation de données :

**A. Classification :** Dans l'apprentissage automatique, la classification est le concept d'apprentissage supervisé qui catégorise essentiellement un ensemble de données en catégories. Les problèmes de classification qui surviennent lorsqu'une variable de sortie est une classe sont utilisés pour classer un algorithme afin de mapper précisément les données de test à des classes spécifiques [23].

Parmi les problèmes de classification les plus courants, nous pouvons citer : la reconnaissance de l'écriture manuscrite, la classification des documents et des images, la classification des spams, etc.

**B. Régression** : est un autre type de méthode d'apprentissage supervisé, basé sur la compréhension de la relation entre les variables dépendantes et indépendantes, car il joue un rôle dans la prédiction des valeurs numériques basées sur différents points de données. Il est divisé en plusieurs types : régression linéaire, régression logistique et régression polynomiale [23].

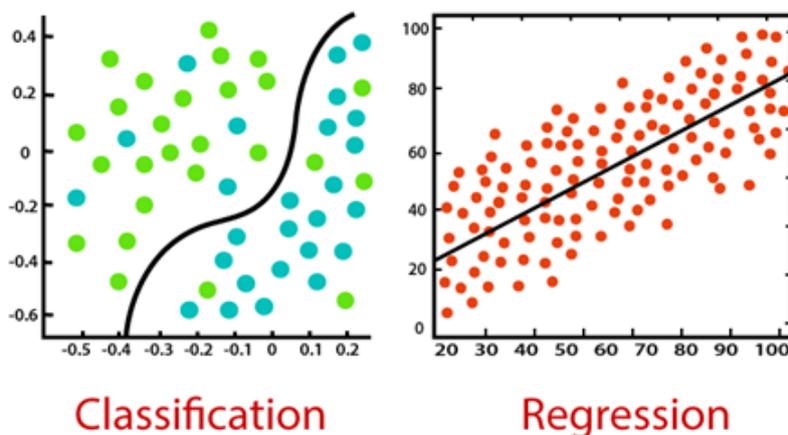


FIGURE 2.2 – La classification et la régression [24]

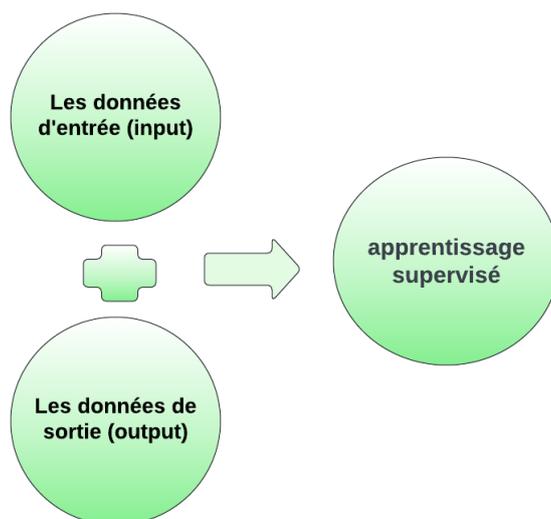


FIGURE 2.3 – Apprentissage supervisé.

## Apprentissage Non supervisé

L'apprentissage non supervisé utilise des algorithmes d'apprentissage automatique (Machine Learning) pour analyser et agréger des ensembles de données non étiquetés. Ces algorithmes révèlent des modèles cachés dans les données sans nécessiter d'intervention humaine [23].

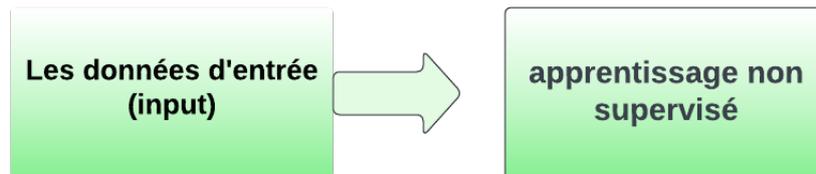


FIGURE 2.4 – Apprentissage non supervisé.

## Apprentissage Semi-supervisé

L'apprentissage semi-supervisé est une approche d'apprentissage automatique (Machine Learning) qui consiste en un petit ensemble de données étiquetées et un grand ensemble de données non étiquetées pendant la formation. Alors que l'apprentissage semi-supervisé se situe entre l'apprentissage supervisé qui ne contient que des données d'entraînement étiquetées et l'apprentissage non supervisé qui ne contient pas de données d'entraînement étiquetées. Il s'agit d'un cas particulier de faible supervision [25].

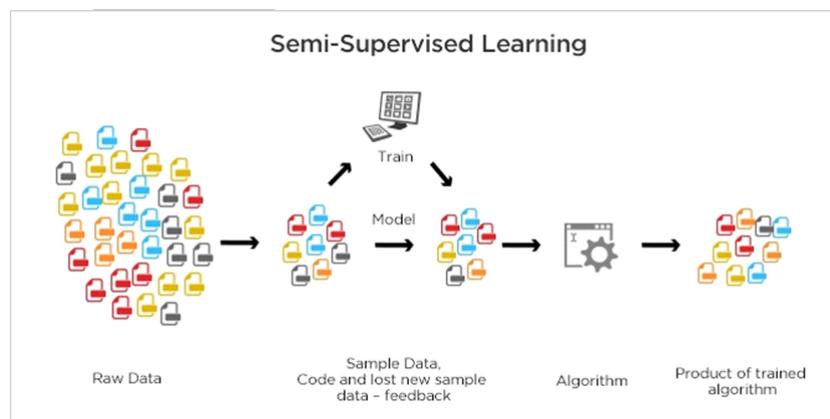


FIGURE 2.5 – Apprentissage semi supervisé [26].

## Apprentissage par renforcement

L'apprentissage par renforcement désigne une classe de problèmes de la Machine Learning, où il vise à apprendre, à partir d'expériences précédentes, ce qu'il faut faire pour trouver la meilleure solution [27].

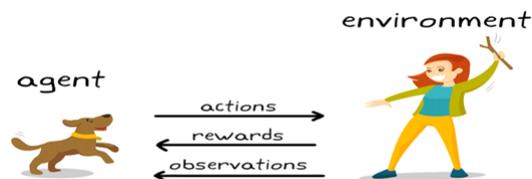


FIGURE 2.6 – Apprentissage par renforcement [28].

## 2.4 Apprentissage profond (Deep Learning)

### 2.4.1 Définition

Il s'agit d'un sous-domaine plus avancé de l'intelligence artificielle dérivé de l'apprentissage automatique, un système hiérarchique, Il consiste à permettre aux machines d'apprendre de leurs expériences au fur et à mesure que les données d'entrée passent par le premier niveau où des informations simples sont extraites, passées au niveau suivant, de manière hiérarchique et séquentielle, et les informations extraites deviennent plus complexes jusqu'à ce que le résultat final soit L'apprentissage en profondeur peut être considéré comme un moyen d'automatiser l'analyse prédictive [29].

### 2.4.2 Domaine d'applications

L'apprentissage profond est utilisé dans de nombreux aspects et devrait s'étendre à de nombreux domaines de notre vie quotidienne à l'avenir. Parmi ces domaines on retrouve la reconnaissance faciale, la reconnaissance vocale, le diagnostic médical, la recommandation personnalisée et la modération automatique des réseaux sociaux. . .

### 2.4.3 Historique

Année	Contributeur	Contribution
1873	Alexander Bain	Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones.
1943	McCulloch and Pitts	Introduction du McCulloch–Pitts (MCP) modèle considéré comme L’ancêtre des réseaux de neurones artificielles (RNA).
1949	Donald Hebb	Considérer comme le père des RNA, il introduit la règle d’apprentissage de Hebb qui servira de fondation pour les RNA modernes.
1958	Frank Rosenblatt	Introduction du premier perceptron.
1974	Paul Werbos	Introduction de la retro propagation.
1980	Teuvo Kohonen	Introduction des cartes auto organisatrices.
1980	Kunihiko Fukushima	Introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif.
1982	John Hopfield	Introduction des réseaux de Hopfield.
1985	Hilton and Sejnowski	Introduction des machines de Boltzmann.
1986	Paul Smolensky	Introduction d’Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes.
1986	Michael I. Jordan	Définition et introduction des RN récurrent.
1990	Yann LeCun	Introduction de LeNet et montra la capacité des RN profonds.
1997	Schuster and Paliwal	Introduction des RN récurrent bidirectionnelles.
1997	Hochreiter and Schmidhuber	Introduction des LSTM, qui ont résolu le problème du vanishing gradient dans les RN récurrent.
2006	Geoffrey Hinton	Introduction des Deep belief Network.
2009	Salakhutdinov and Hinton	Introduction des Deep Boltzmann Machines.
2012	Geoffery Hinton	Introduction dropout, un moyen efficace d’entraîner les réseaux de neurones.

TABLE 2.1 – L’historique de l’apprentissage profond [30] .

#### 2.4.4 Différence entre l'apprentissage automatique et l'apprentissage profond

Historiquement, l'apprentissage automatique est une technologie plus ancienne et plus simple. Il fonctionne avec un algorithme qui s'adapte lorsqu'il reçoit des commentaires humains. L'une des conditions requises pour tirer parti de cette technologie est la disponibilité des données conservées.

Alors qu'avec l'apprentissage profond (deep learning), les données extraites ne sont plus nécessaires. C'est pourquoi le système peut également traiter des données non structurées.

Dans l'apprentissage en profondeur, le système trouve lui-même les propriétés de différenciation appropriées dans les fichiers, sans avoir besoin d'une classification externe. En d'autres termes : la formation par le développeur n'est pas nécessaire. Alors que l'apprentissage automatique peut en effet fonctionner avec un ensemble gérable de données, l'apprentissage en profondeur nécessite plus de données. Pour que le système produise des résultats fiables.

La technologie d'apprentissage en profondeur nécessite également plus de ressources à mettre en œuvre. Cela nécessite plus de ressources informatiques et coûte beaucoup plus cher que l'apprentissage automatique, ce qui signifie qu'il n'est - pour le moment du moins - pas une option pour les entreprises grand public.

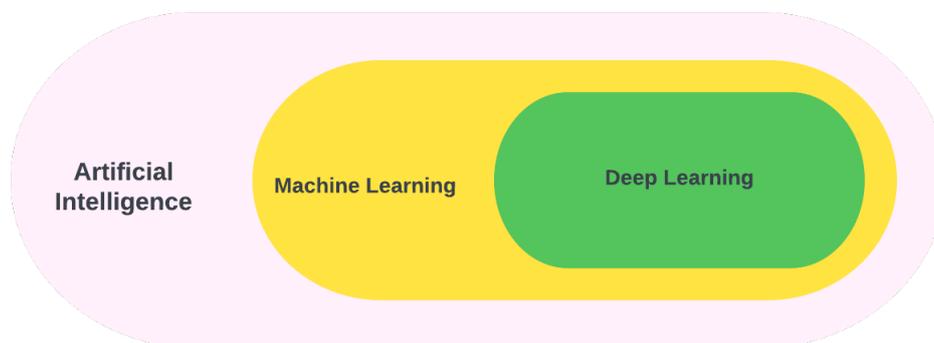


FIGURE 2.7 – la relation entre IA, ML et DL.

Dans le tableau ci-dessous, nous allons voir les points les plus différents entre eux :

	<b>l'apprentissage automatique</b>	<b>l'apprentissage profond</b>
<b>Matérielles</b>	CPU	GPU
<b>Bibliothèque Python</b>	Scipy , Panda , Scikit-learn	Tensorflow , Keras , Pytorch
<b>Temps d'exécution</b>	Nécessite relativement peu de temps	Nécessite un temps d'entraînement assez long,
<b>Domaines d'applications principaux</b>	Classification et Régression données tabulaires	Classification et Régression comme traitement de NLP et Computer Vision, données tabulaires,
<b>Sortie</b>	La sortie est une valeur numérique, telle qu'une note ou une classification.	La sortie peut avoir plusieurs formats : texte, un son, un score.

TABLE 2.2 – La différence Entre l'apprentissage automatique et l'apprentissage profond.

### 2.4.5 Les réseaux de neurones convolutifs

L'apprentissage en profondeur est un domaine en pleine croissance et de nouvelles architectures alternatives pour les algorithmes émergent constamment. Dans notre travail, nous nous intéressons aux modèles de classification des images qui définissent une sous-catégorie de réseaux de neurones.

#### Définition

Dans le domaine de la vision par ordinateur, les réseaux de neurones convolutifs (CNN pour Convolutional Neural Networks) est l'algorithme le plus connu et le plus utilisé. Le principal avantage de CNN par rapport à ses prédécesseurs est que l'extraction des caractéristiques pertinentes se fait de manière automatique à partir des données en entrée.

Les CNN ont été largement appliqués dans une gamme de domaines différents, y compris la vision par ordinateur, le traitement de la parole, la reconnaissance faciale [31].

De plus on peut les utiliser dans tous les problèmes ayant en entrée une matrice. Par exemple, Gehring a utilisé une matrice de texte dans une tâche de traduction automatique de langue. Le réseau de neurones convolutif est spécialisé dans le traitement des données matricielles et du signal.

### Les couches de réseaux de neurones convolutifs

La structure CNN est constituée d'un ensemble de couches, nous allons essayer d'expliquer chacune séparément.



FIGURE 2.8 – Les différentes couches de CNN.

**A. Couche d'entrée :** Qui permet d'introduire (envoyer) vos données dans votre réseau de neurones. Il s'agit généralement de la matrice de pixels d'une image. Cette matrice a trois dimensions : longueur, largeur, profondeur.

**B. Couche de convolution :** Est la couche de base pour la construction d'un réseau neuronal convolutif, La convolution est une opération arithmétique permettant de combiner deux ensembles d'informations Une image passe à travers une succession de filtres créant de nouvelles images appelées cartes de convolutions. c'est carte concaténées mises à plat et pour produire un vecteur de caractéristiques [32].

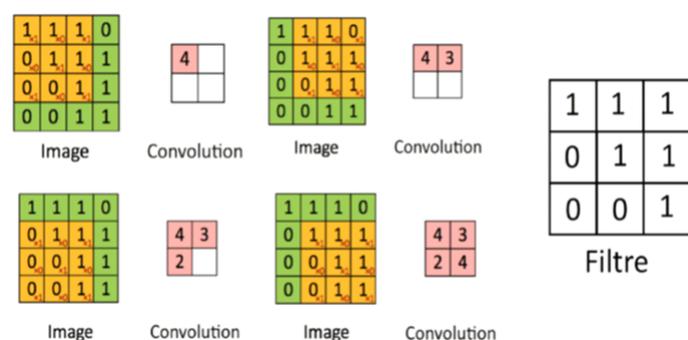


FIGURE 2.9 – La couche de convolution [33].

- **Les filtres** : Les filtres sont très importants dans le processus d'extraction des attributs d'image dans l'étape de convolution. Et vous pouvez appliquer à toutes les images, car ces filtres prennent la forme d'un carré avec quelques côtés et sont appelés noyaux.
- **Les Pas (Stride)** : Représente le nombre de transitions de carré à carré sur toute la matrice d'entrée lors de la numérisation de l'image lorsque le pas est de 1, nous déplaçons les filtres d'un pixel à la fois.

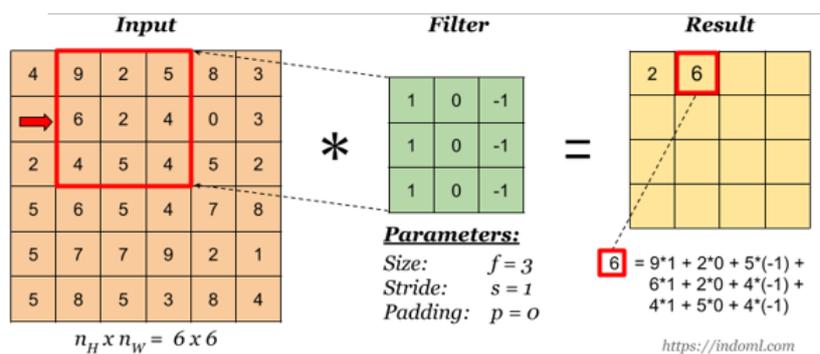


FIGURE 2.10 – Exemple d'application de pas "stride" [34].

- **Rembourrage (padding)** : Nous utilisons le rembourrage lorsque nous voulons augmenter la taille de la sortie et enregistrer les informations fournies dans les coins en ajoutant des lignes et des colonnes supplémentaires sur la dimension extérieure des images afin de rendre la taille des données d'entrée similaire aux données de sortie. Cela permet également au filtre de couvrir plus d'espace dans l'image [35].

Quelques terminologies de rembourrage :

- Rembourrage « valid ».
- Rembourrage « same ».
- Rembourrage « full ».

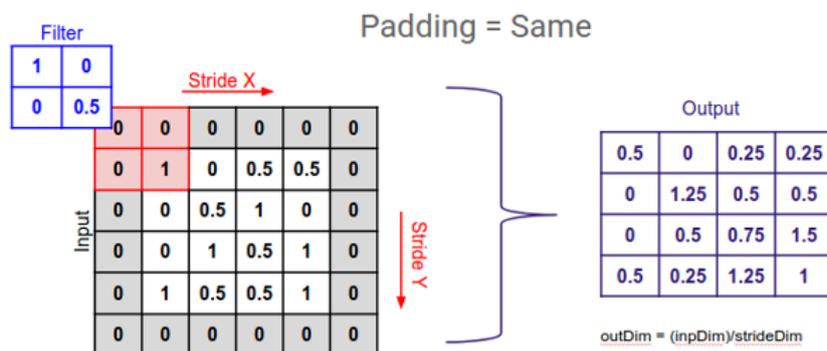


FIGURE 2.11 – Exemple d’application de rembourrage [36].

**C. Couche de correction/activation (ReLU) :** ReLU est la fonction d’activation la meilleure et la plus avancée à l’heure actuelle par rapport à la sigmoïde et TanH, car tous les défauts tels que le problème de gradient de fuite sont complètement supprimés car il remplace les nombres négatifs d’images filtrées par des zéros peut être considéré comme le plus rapide par rapport aux autres fonctions d’activation [37].

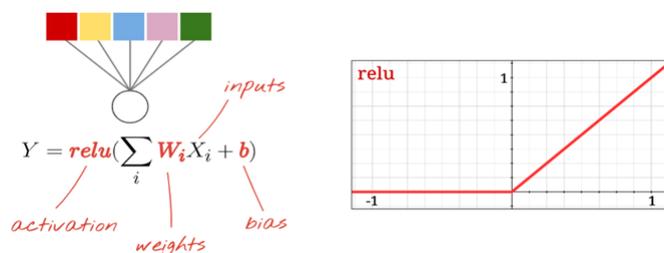


FIGURE 2.12 – La couche ReLU [32].

**D. Couche pooling :** La tâche principale de la couche de regroupement est le sous-échantillonnage des cartes d’entités. Ces cartes sont générées en suivant les opérations convolutionnelles. En d’autres termes, cette approche réduit les cartes d’entités de grande taille pour créer des cartes d’entités plus petites. Parallèlement, il conserve la majorité des informations (ou caractéristiques) dominantes à chaque étape de l’étape de mise en commun. D’une manière similaire à l’opération convolutive, la foulée et le noyau sont initialement attribués en taille avant l’exécution de l’opération de regroupement. Les méthodes de mise en commun les plus connues et les plus fréquemment utilisées sont la mise en

commun max pooling et average pooling [31], le tableau 2 illustre ces deux opérations de pooling :

Type	Max pooling	Average pooling
But	Sélection la valeur maximale de la surface.	Sélection la valeur moyenne de la surface.
Illustration		
Commentaires	Garder les caractéristiques détectées Plus communément utilisé.	Sous-échantillonne la feature map Utilisé dans LeNet.

TABLE 2.3 – Les types de pooling [38].

- **Aplatissement (Flattening)** : L'étape d'aplatissement consiste à prendre la carte de caractéristiques assemblée créée lors de l'étape d'agrégation et à la transformer en un vecteur unidimensionnel dans le but d'alimenter ce vecteur dans un réseau de neurones artificiels. Cela peut être considéré comme une étape simple et rafraîchissante impliquée dans la construction d'une convolution réseau neuronal [39].

Voici une représentation visuelle de ce à quoi ressemble ce processus :

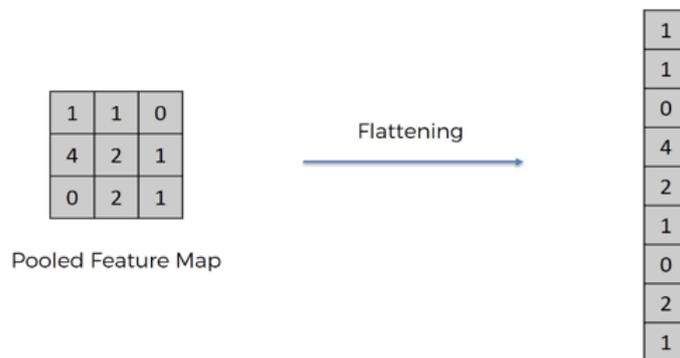


FIGURE 2.13 – Types de Pooling [39].

**E. Couche entièrement connectée (Fully Connected) :** La couche entièrement connectée (FC) généralement située à la fin de chaque architecture CNN est simplement une alimentation directe des réseaux de neurones, l'entrée de la couche entièrement connectée est la sortie de l'assemblage final ou de la couche convolutive, cette entrée est sous forme vectorielle, elle est générée à partir de cartes d'entités, et la sortie de la couche finale et convolutive est une matrice 3D [40].

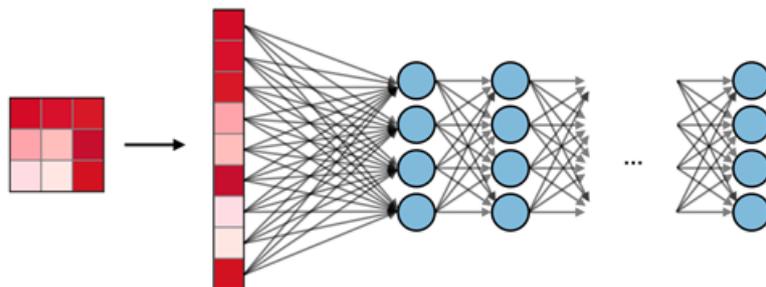


FIGURE 2.14 – Couche entièrement connectée (FC) [38].

**F. Couche logistique (softmax) :** L'étape softmax peut être vue comme une généralisation de la fonction logistique qui prend comme argument un vecteur de scores  $x \in \mathbb{R}^n$  et qui renvoie un vecteur de probabilités  $p \in \mathbb{R}^n$  à travers une fonction softmax à la fin de l'architecture. Elle est définie de la manière suivante [38] :

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{où} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

**G. Couche de sortie :** La couche de sortie contient une étiquette qui indique le résultat attendu.

## Exemples d'architectures CNN

Il existe plusieurs architectures dans le domaine des réseaux convolutifs qui ont un nom. Les plus courants sont :

Architecture	Année	Description
<b>AlexNet</b> [41]	2012	Le premier travail qui a popularisé les réseaux convolutifs en vision par ordinateur est AlexNet, développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton. AlexNet a été soumis au défi ImageNet LSVRC en 2012 et a largement surpassé le deuxième finaliste. Le réseau avait une architecture très similaire à LeNet, mais était plus profond, plus grand et comportait des couches convolutionnelles empilées les unes sur les autres (auparavant, il était courant de n'avoir qu'une seule couche CONV toujours immédiatement suivie d'une couche POOL)
<b>DenseNet</b> [42]	2017	Un DenseNet est un type de réseau neuronal convolutif qui utilise des connexions denses entre les couches, via des blocs denses, où nous connectons toutes les couches (avec des tailles de carte de fonctionnalités correspondantes) directement les unes avec les autres. Pour préserver la nature prédictive, chaque couche obtient des entrées supplémentaires de toutes les couches précédentes et transmet ses propres cartes d'entités à toutes les couches suivantes.
<b>ResNet</b> [43]	2015	Les modèles ResNet ont énormément contribué à l'utilisation des réseaux de neurones très profonds, en limitant la perte de gradient dans les couches les plus profondes de celui-ci par l'ajout d'une liaison résiduelle entre chaque couche de convolution.

<b>GoogLeNet</b> [44]	2014	GoogLeNet est un type de réseau neuronal convolutif basé sur l'architecture Inception. Il utilise des modules Inception, qui permettent au réseau de choisir entre plusieurs tailles de filtre convolutionnel dans chaque bloc. Un réseau Inception empile ces modules les uns sur les autres, avec des couches de mise en commun maximale occasionnelles avec la foulée 2 pour réduire de moitié la résolution de la grille.
<b>VGG</b> [45]	2014	En effet, le moyen le plus simple d'améliorer les performances des réseaux de neurones profonds consiste à augmenter leur taille (Szegedy et. al). Les gens de Visual Geometry Group (VGG) ont inventé le VGG-16 qui a 13 couches convolutionnelles et 3 entièrement connectées, emportant avec eux la tradition ReLU d'AlexNet. Ce réseau empile plus de couches sur AlexNet et utilise des filtres de plus petite taille ( $2 \times 2$ et $3 \times 3$ ). Il se compose de 138M de paramètres et occupe environ 500Mo d'espace de stockage. Ils ont également conçu une variante plus profonde, VGG-19.
<b>Xception</b> [46]	2016	le modèle Xception implique une convolution en profondeur et ponctuelle. Le travail de Chollet,F montre l'amélioration de Xception par rapport aux modèles précédents. Le modèle Xception est un CNN profond à 71 couches, inspiré du modèle Inception de Google, et il est basé sur une interprétation extrême du modèle Inception. Son architecture est empilée avec des couches convolutives séparables en profondeur. La version pré-formée du modèle est formée à l'aide de millions d'images du réseau Imagenet ...

TABLE 2.4 – Les architectures très connues de CNN.

## L'entraînement (training)

Nous pouvons utiliser des algorithmes d'optimisation et des algorithmes de rétropropagation pour entraîner des réseaux de neurones. Le calcul des gradients se fait par rétropropagation pour chaque variable du modèle, ce qui est un préalable à l'optimisation.

**A. Rétro propagation :** est un moyen efficace de calculer les gradients. Il s'agit d'une méthode couramment utilisée pour entraîner les réseaux de neurones frontaux, qui dépend de l'ajustement des poids du réseau de neurones en fonction du taux d'erreur que nous avons obtenu lors des itérations précédentes.

**B. Optimisation de l'apprentissage profond :** Les optimiseurs sont des algorithmes ou des méthodes utilisés pour minimiser une fonction d'erreur (fonction de perte) ou pour maximiser l'efficacité de la production. Les optimiseurs sont des fonctions mathématiques qui dépendent des paramètres apprenables du modèle, c'est-à-dire des poids et des biais. Les optimiseurs aident à savoir comment modifier les poids et le taux d'apprentissage du réseau de neurones pour réduire les pertes [47].

Voici une liste de certains algorithmes d'optimisation

N	Méthode	Année
1	Adam	2014
2	SGD	1951
3	ADMM	2000
4	RMSProp	2013
5	SGD with Momentum	1999
6	AdaGrad	2011
7	TTUR	2017
8	Gradient Clipping	2000
9	LAMB	2019
10	AMSGrad	2019

TABLE 2.5 – Méthodes d'optimisation les plus utilisées [48].

## 2.4.6 Le problème du sur-apprentissage (OVERFITTING)

### Définition :

Les réseaux de neurones profonds (apprentissage en profondeur) ne sont que des réseaux de neurones artificiels avec de nombreuses couches entre les entrées et les sorties (prédiction), ce qui augmente la complexité du réseau.

Lorsque la base de données d'entraînement contient très peu d'exemples, il est probable qu'il y ait sur apprentissage (OVERFITTING), c'est-à-dire lorsque le réseau est capable de prédire avec précision les échantillons des données d'entraînement mais a des performances médiocres et ne peut pas bien généraliser sur la validation et les données de test [49].

### Fonctionnement du sur-apprentissage :

Au fur et à mesure que le modèle est formé sur plusieurs époques, il commence à comprendre les modèles de l'ensemble de données. Ainsi, lorsque le modèle est su rajusté, la précision des données de formation serait très élevée et celle de la validation très faible, car le modèle a tout appris. les modèles dans les données de formation, mais ne peut pas généraliser et faire de bonnes prédictions sur des données qu'il n'a pas vues auparavant, ce qui va à l'encontre de l'objectif de la formation du modèle, qui est de faire des prédictions à l'avenir sur des données qu'il n'a pas vues auparavant.

Ce que nous voulons réaliser, c'est réduire le sur apprentissage, une situation où la précision de notre formation et de notre validation est proche et également élevée, ce qui suggère que le modèle peut faire de meilleures prédictions sur des données qu'il n'a jamais vues auparavant [49].

### Technique pour éviter le sur-apprentissage :

**A. L'abandon (dropout) :** est une technique qui supprime les neurones du réseau de neurones ou les "ignore" pendant l'entraînement, en d'autres termes, différents neurones sont temporairement retirés du réseau. Pendant l'entraînement, le décrochage modifie l'idée d'apprendre tous les poids du réseau pour apprendre une petite partie des poids du réseau.

D'après la figure ci-dessous, on peut voir que pendant la phase d'entraînement standard, tous les neurones participent et pendant le dropout, seuls quelques neurones sélectionnés participent avec le reste "off". Ainsi après chaque itération, différents groupes de neurones sont activés, pour empêcher certains neurones de prendre le relais du processus.

Cela nous aide donc à réduire le risque de sur ajustement et permet l'émergence d'architectures de réseau plus profondes et plus larges qui peuvent faire de bonnes prédictions sur les données que le réseau n'a jamais vues auparavant [49].

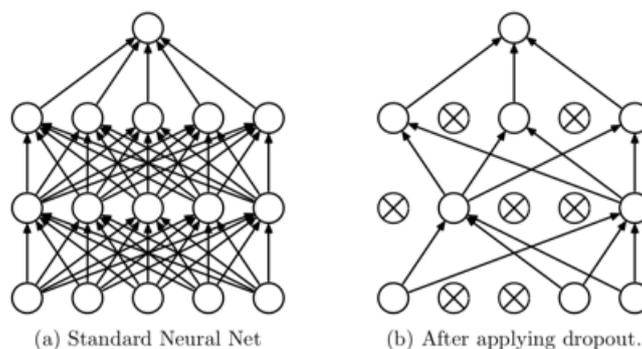


FIGURE 2.15 – Description du L'abandon [49].

**B. Apprentissage par transfert (Transfer learning) :** Il s'agit d'une méthode d'apprentissage profond qui utilise cette technique pour travailler sur un modèle qui a été formé et développé comme point de départ d'un modèle sur une deuxième tâche. Il vise à réduire le temps d'entraînement. Son rôle principal est de réduire le problème de l'insuffisance des données d'entraînement, qui est un problème majeur de sur-apprentissage [50].

## 2.5 Base de données (DataSet)

Une base de données est un ensemble d'informations organisées de manière à faciliter la recherche, la gestion et la mise à jour, qu'il s'agisse de fichiers vidéo, d'images, de textes, de sons ou encore de statistiques, leurs combinaisons forment un ensemble complet. Ils sont accessibles individuellement ou en groupe. Il existe différents modèles, tels que l'ensemble de test et l'ensemble de données d'apprentissage.

— **Exemples de bases de données utilisées pour le diagnostic du COVID-19 :**

Des ensembles de données contenant de nombreuses images médicales de personnes normales et de patients COVID-19 ont été créés afin de faciliter la détection des maladies COVID. Parmi ces ensembles de données figurent les suivants :

N°	nom de Dataset	Nbr total des images	Nbr d'images normal	Nbr d'images Covid	Ref
1	COVID-19 Radiography Database	21200	10192	3616	[51]
2	COVID-19 X-ray images	357	0	357	[52]
3	Covid-19 Image Dataset	317	90	137	[53]
4	COVIDx CXR-2	16400	0	16400	[54]
5	COVID-19 Patients Lungs X Ray images 10000	98	28	70	[55]
6	Chest X-ray (Covid-19 & Pneumonia)	6432	1583	576	[56]
7	COVID-19 Chest X-ray Image Dataset	94	25	69	[57]
8	Chest Xray for covid-19 detection	371	176	174	[58]
9	CoronaHack -Chest X- Ray- Datas	5935	0	5935	[59]
10	COVID-19 X rays	97	0	97	[60]

TABLE 2.6 – Exemples de bases de données utilisées pour le diagnostic du COVID-19.

## 2.6 Les travaux connexes pour le diagnostic de COVID-19

L'utilisation des rayons X pour identifier automatiquement les maladies thoraciques et pulmonaires grâce à l'intelligence artificielle est devenue une source d'inspiration pour de nombreux scientifiques, en particulier ces derniers temps avec l'émergence de la maladie

## COVID-19.

- En 2020, **Apostolopoulos et al [61]** ont utilisé deux ensembles de données d'apprentissage par transfert, et ont utilisé une base de données de 1427 images radiographiques (X-ray) qui contient 224 images COVID-19 et le reste entre pneumonie et normal cas (n'est infecté ni par l'un ni par l'autre) cette recherche a atteint une précision de 96,78 %.
- Dans l'article de la référence numéro [62], **Naren et al** ont utilisé l'apprentissage par transfert pour une classification à quatre classes, cette recherche a nécessité l'utilisation de cinq modèles pré-entraînés sur des réseaux de neurones convolutifs, le modèle préentraîné Resnet50 fournit les meilleurs résultats dans cette catégorie avec une précision de 96.1%.
- En 2020 aussi, **Maghdid et al [64]** ont construit un réseau et apporter des modifications à un modèle AlexNet déjà formé avec un ensemble de données comprenant des rayons X et des tomodensitogrammes, cette approche a enregistré une précision de 98 %.
- En 2021 **sarath Pathari [65]** : Dans cette étude, un ensemble de données d'images radiographiques de patients atteints de pneumonie et de la maladie COVID-19 confirmée a été utilisé pour la détection, cette recherche a nécessité l'utilisation de modèle mobilenetv3 avec une base de données qui contient 24000 radiographiques (X-ray), cette méthode a donné une précision de 95,58 %.
- En 2020, **Abdolkarim Saeedi et al [66]**, ont introduit un service Web de diagnostic assisté par ordinateur, l'une des plus grandes bases de données publiques sur les tomodensitogrammes thoraciques, contenant 746 images Tomodensitométrie (CT Scan) COVID-19, ils ont utilisé une combinaison de DenseNet afin de réduire les dimensions de l'image et Nu-SVM comme goulot d'étranglement anti-overfitting. Ils ont obtenu un résultat de 90,61% de précision.
- Dans la même année, **Heidari M et al [67]** ont développé et testé un nouveau schéma de diagnostic assisté par ordinateur (CAO). Il comprend plusieurs algorithmes qui génèrent trois images d'entrée, puis se connectent à un réseau neuronal convolutif basé sur la classification d'images radiographiques thoraciques en trois catégories, à l'aide d'une base de données de 8 474 images, Ils trouvent que la meilleure précision pour cette méthode est de 94 %.

- En 2022 **Taban Majeed et al** [68], ils ont utilisé l'apprentissage par transfert avec 12 architectures CNN pour réaliser une classification Multi-classes, pour la base de données ils ont utilisé 3 BDD de 111 images, 6292 images et 73 images successivement. Cette approche a enregistré une Sensibilité (Sensitivity) de 94,52%.
- En 2020, **Linda Wang, Alexander Wong** [69] présentent un réseau de neurones à convolution conçu pour détecter les cas de COVID-19 à partir d'images de radiographie pulmonaire en utilisant l'architecture COVID-Net, et ils utilisent une base de données COVIDX qui contient 13870 radiographiques (X-ray), le modèle a donné une précision de 93.3%.
- Durant la même année **Hemdan et al** [70] ont utilisé l'apprentissage par transfert à l'aide de 7 architectures CNN avec une méthode qui s'appelle CovidX-net qui contient une base de données de 50 radiographiques (X-ray). Cette méthode nous a donné une précision de 90%.

Le tableau suivant résume les travaux cités précédemment. Avec différentes base de données (dataset) :

Travaux et années	Dataset	Modalité	Méthodes	Nb classes	Résultats
[61] 2020	1427 X-ray images : -224 COVID-19 -700 pneumonia -504 normal	X-ray	Transfert learning	classification binaire	Précision= 96,78% Sensibilité= 98,66% Spécificité= 96,46%
[62] 2021	341 X-ray images COVID-19 2800 normal	X-ray	ResNet50 ResNet101 ResNet152 InceptionV3 Inception-ResNetV2	Classification binaire	Précision = 96.1%

<b>[63]</b> <b>2020</b>	170 X-ray images COVID-19 361 CT scan images COVID-19	X-ray CT scan	AlexNet	Classification binaire	Précision = 98 % Sensibilité = 100% Spécificité = 96%
<b>[64]</b> <b>2021</b>	24000 X-ray images : -6000 covid- 19 -6000 pneumo- nia -6000 normal	X-ray	Transfert learning MobileNetV3	4 classes	Précision = 95,58 % Sensibilité = 97,52% Spécificité = 95,14%
<b>[65]</b> <b>2020</b>	746 CT scan images COVID-19	CT scan	DenseNet121 InceptionV3 ResNet50 MobileNet	Classification binaire	Précision = 90,61 %
<b>[66]</b> <b>2020</b>	8474 X-ray images : -415 COVID-19 -5179 pneu- monia -2880 normal	X-ray	VGG16	3 classes	Précision = 94%
<b>[67]</b> <b>2020</b>	3 dataset : -111 X- ray images COVID-19 -6292 X- ray images -73 images COVID-19	X-ray	12 archi- tectures CNN	Multi classes	Sensibilité = 94,52%

[68] 2020	COVIDX 13870 X-ray images	X-ray	Covid-Net	Multi classes	Précision = 93,3% Sensibilité = 91%
[69] 2020	50 X-ray images -25 COVID-19 -25 normal	X-ray	7 architec- tures CNN	Classification binaire	Précision = 90%

TABLE 2.7 – Comparaison des travaux connexes pour le diagnostic de COVID-19

## 2.7 Conclusion

Dans ce chapitre, nous avons présenté l'apprentissage automatique dans ses différents types (supervisé, non supervisé, semi-supervisé et par renforcement). Nous avons notamment exploré la technologie de l'apprentissage profond et essayé de clarifier les concepts qui nous intéressent dans ce domaine, et donné la différence entre l'apprentissage en profondeur et l'apprentissage automatique.

Ensuite nous avons donné un aperçu sur l'état de l'art des systèmes automatiques utilisés pour la détection de cette maladie basée sur les réseaux de neurones convolutifs (CNN).

Enfin, nous avons présenté des travaux connexes qui ont été utilisés pour la classification de COVID-19 à partir d'images radiographiques. Dans le prochain chapitre, nous présentons l'ensemble de données que nous utiliserons. Ensuite, nous traiterons des détails de conception et de mise en œuvre pour parvenir à notre solution.

## Approches et solutions proposées

### 3.1 Introduction

Le développement de l'intelligence artificielle a eu de grands effets dans le domaine médical, en particulier dans l'imagerie médicale, où l'apprentissage en profondeur est considéré comme l'une des technologies les plus intéressantes dans ce domaine.

Dans ce chapitre, nous analyserons et décrirons d'abord l'ensemble de données que nous utiliserons dans notre travail, puis fournirons des détails sur notre solution proposée.

### 3.2 Vue globale de notre système

La figure 3.1 représente la conception globale de notre système, où nous allons d'abord utiliser une base de données qui représente des images radiographiques du thorax. Ensuite, nous passons à l'étape de pré-traitement afin de clarifier l'image et d'améliorer sa qualité, après quoi nous la diviserons la base de données en trois parties (données d'entraînement, données de validation et données de test), puis nous créerons un modèle qui entrera les données traitées et s'entraîne sur une partie des données d'entraînement. Le système introduit une image de la radiographie thoracique et la classe en trois classes (soit un cas de COVID, un cas normal ou un cas de pneumonie virale).

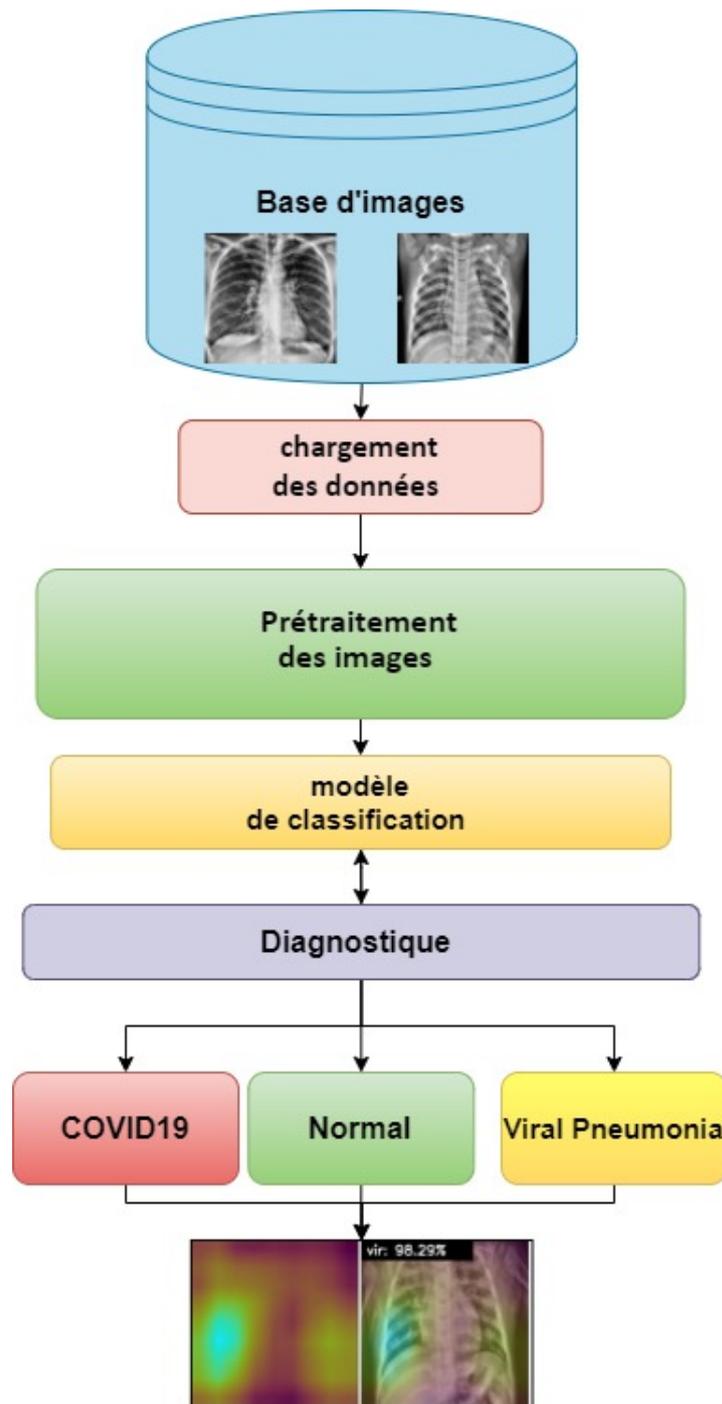


FIGURE 3.1 – Représentation de l'architecture globale de notre système.

### 3.3 Présentation détaillée de notre système

#### 3.3.1 Base de données

La raison pour laquelle nous avons choisi d'utiliser les radiographies thoraciques (X ray), et non la tomodensitométrie (CT scan), était que les médecins les préféraient pour

diagnostiquer d'autres pneumonies. Et il est intéressant d'explorer cette voie car il est plus facile de faire une radiographie thoracique qu'un scanner tomodensitométrie (les scanners ne sont pas omniprésents et chers), surtout si l'appareil peut attirer l'attention du médecin sur des endroits de l'image qu'il n'a pas remarqués (Grad-CAM). C'est pourquoi nous avons choisi cette base de données.

Notre base de données se compose de quatre classes mais nous n'avons utilisé que trois classes dont nous avons besoin dans notre travail, et nous avons changé le nombre pour rendre toutes les classes égales, la première classe est les images médicales thoraciques des patients COVID (1345 images), la deuxième est les images médicales thoraciques des patients normaux (1345 images) et la troisième classe est les images médicales thoraciques des patients atteints de pneumonie virale (1345 images) qui ont été extraites de cette base de données [51].

Les images de cette base de données sont de taille (299\*299 pixels) sous format (PNG).

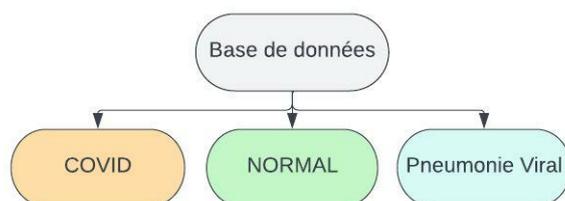


FIGURE 3.2 – Diagramme global de notre base de données.

Nous avons divisé cette base de données en trois sous-ensembles, une pour l'entraînement, une pour la validation et une pour le test (train/val/test).

- **Train** : un échantillon de l'ensemble de données utilisé pendant la phase d'entraînement du modèle.
- **Test** : un échantillon de l'ensemble de données utilisé pour tester et évaluer le modèle final.
- **Validation** : un échantillon de l'ensemble de données utilisé pendant la phase de validation.

Où nous l'avons divisé comme suit : 80% pour l'entraînement (train), 10% pour le test et 10% pour la validation.

Ensemble de données	COVID-19	Normal	Pneumonie Viral	Totales
Train	1076	1076	1076	3228
Test	134	134	134	402
Validation	135	135	135	405
Totales	1345	1345	1345	4035

TABLE 3.1 – représentation de données.

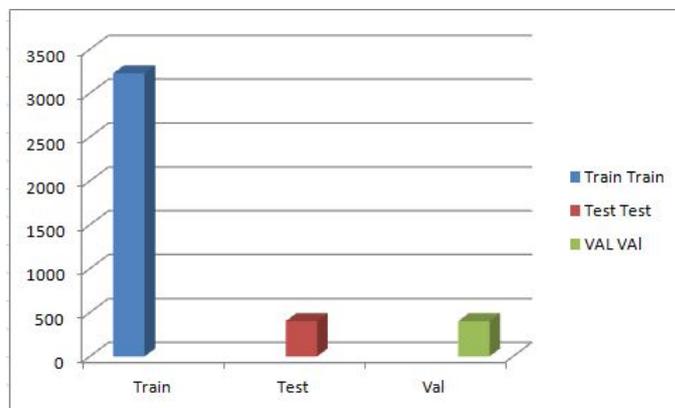
La division de ces données a été effectuée à l'aide du code présenté ci-dessous :

```
import splitfolders

database=('C:/Users/maiss/OneDrive/Desktop/DATA')

splitfolders.ratio(database,output="C:/Users/maiss/Downloads/Newdata",seed=1227,ratio=(0.8,0.1,0.1))
```

FIGURE 3.3 – code python pour la division des données.



### 3.3.2 Prétraitement de la base de données

L'étape de modification et de traitement de l'image est une étape très importante dans l'analyse des données, afin de donner au modèle la capacité de mieux extraire les caractéristiques pertinentes, car le négliger peut conduire à un sur-ajustement (surapprentissage) et le modèle peut nous donner résultats erronés.

Pour notre ensemble de données, nous l'avons converti au format PNG en modifiant la

taille du résultat en (299 \* 299) afin que toutes les données correspondent les unes aux autres.

Nous avons utilisé deux algorithmes afin de clarifier les images et d'améliorer leur qualité et de bien mettre en valeur leur contenu, où nous avons utilisé n-clahe et aussi gamma pour l'étape de pré-traitement.

- **N-CLAHE** : L'approche N-CLAHE est la combinaison de la fonction de normalisation et de la méthode CLAHE (Contrast Limited Adaptive HistogramEqualization). La technique proposée consiste en deux étapes principales. Premièrement, la correction d'intensité globale de l'image brute est normalisée par la fonction log avec l'image de balayage vierge. Deuxièmement, le calcul de l'amélioration supplémentaire du contraste local est opéré par l'approche CLAHE [70].

- **GAMMA** : La correction gamma consiste à combler la différence entre les représentations linéaires des intensités lumineuses et la réponse non linéaire de l'œil humain, qui est plus sensible aux variations de faible luminosité qu'à une variation égale de la lumière vive. La correction gamma est une opération de traitement d'image qui compense les comportements inhérents de reproduction de tonalité d'un appareil de capture ou pour préparer une image pour la sortie sur un moniteur ou une imprimante, qui peut également être calibrée en termes non linéaires. Pour un exemple intéressant de bricolage non linéaire [71].

```
def gamma_trans(img, gamma):
    gamma_table = [np.power(x / 255.0, gamma) * 255.0 for x in range(256)]
    gamma_table = np.round(np.array(gamma_table)).astype(np.uint8)
    return cv2.LUT(img, gamma_table)

target_path = 'C:/Users/maiss/Downloads/1300 Data set Nclah/test/COVID'
for file_name in glob(os.path.join('C:/Users/maiss/OneDrive/Desktop/1300 data set/test/COVID/*')):
    src_img = cv2.imread(file_name, 0)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(16, 16))
    img_clahe = clahe.apply(src_img)
    mean = np.mean(img_clahe)
    gamma_val = math.log10(0.5)/math.log10(mean/255)
    image_gamma_correct = gamma_trans(img_clahe, gamma_val)
    image_gamma_correct = cv2.cvtColor(image_gamma_correct, cv2.COLOR_GRAY2RGB)
    #image_resize = cv2.resize(image_gamma_correct, (299, 299))
    cv2.imwrite(os.path.join(target_path, file_name.split('\\')[-1]), image_gamma_correct)
```

FIGURE 3.4 – code python utilisé pour le traitement (n-clahe et gamma) .



FIGURE 3.5 – exemple de traitement d'image avec N-CLAHE ET GAMMA.

**Augmentation de la base de données :** La possibilité que les images médicales soient verticales dans un grand pourcentage ou toujours claires n'est pas toujours certaine. Pour faire face à ce problème, nous faisons un ensemble d'ajustements, ou ce qu'on appelle aussi l'incrément, qui sert à augmenter la taille de l'ensemble de données. Ces augmentations incluent la rotation à gauche et à droite en plus de la réflexion zoom avant et arrière horizontal de l'image et la normalisation par la division sur 255.

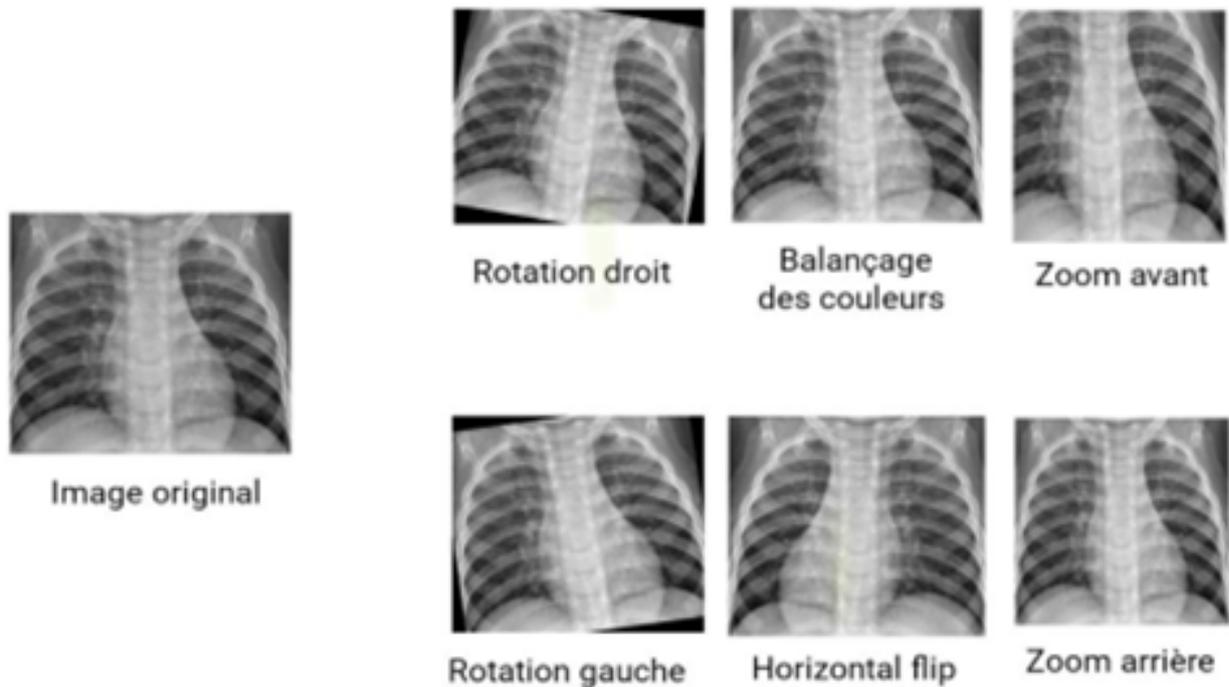


FIGURE 3.6 – augmentation de données.

### 3.3.3 Modèle de classification

L'objectif de notre travail dépend notamment du processus de classification. A cette étape, nous allons créer des modèles qui classent les images médicales en 3 classes. Dans ce travail, nous utiliserons la technique d'apprentissage par transfert, qui est une solution à de nombreux problèmes dans réseaux de neurones, et la raison de son succès est sa capacité à utiliser les connaissances acquises lors de l'exécution d'une tâche pour résoudre un problème différent mais similaire. Nous diviserons notre travail en 2 approches, où la première approche est divisée en deux étapes, dans la première étape nous utiliserons l'apprentissage par transfert en faisant des changements basés sur les modèles (ResNet, DenseNet et Xception) et nous l'utiliserons aussi pour concaténer deux modèles ensemble (ResNet et Xception) et nous avons utilisé Grad-CAM qui est une technique populaire pour visualiser où un modèle de réseau neuronal convolutif regarde. Grad-CAM est spécifique à une classe, ce qui signifie que lorsque la prédiction de l'image permet de savoir à quelle classe elle appartient, elle produit une visualisation distincte sous la forme d'un heatmap approximatif qui met en évidence les régions d'intérêt de cette image pour le concept de prédiction [72].

Dans la deuxième étape nous utiliserons toujours l'apprentissage par transfert pour analyser l'impact de la taille de la base (base d'images non médicales) sur les performances de la classification, où nous l'entraîné avec une énorme base de données, (ImageNet1k et ImageNet21k), et dans la deuxième approche, nous utiliserons la technique Vision Transformer.

Dans ce qui suit, nous présenterons comment créer ces modèles.

## Approche 1 :

### Étape 1 :

**L'apprentissage par transfert (transfer learning) :** C'est une technique pour éviter le surapprentissage qui est défini précédemment dans le deuxième chapitre. Nous l'avons utilisé après avoir créé 4 modèles qui ils sont était entraînés sur une base d'images naturelle (non médicales) qu'il s'appelle ImageNet [73] et les avons entraînés sur notre base

de données après avoir changé les couches de chaque modèle pour le rendre cohérent avec notre travail pour donner des résultats plus précis.

**Modèle ResNet50 :** Nous avons créé un modèle basé sur CNN basé sur un modèle ResNet50 pré-entraîné. Où nous avons utilisé l'apprentissage par transfert (transfer learning) dans cette architecture et apporté quelques modifications aux dernières couches selon les besoins.

Tout d'abord, la couche d'entrée du modèle reçoit une image ( $299 * 299 * 3$ ) et les caractéristiques est extraite au niveau ResNet50.

La sortie de la couche d'extraction des caractéristiques est transmise à la couche MaxPooling2D pour prendre le maximum parti de chaque carte de statistiques, suivi par FLATTEN qui modifie la forme des données en une entité unidimensionnelle. Suivi à la fois de la couche BatchNormalization pour améliorer l'entraînement et d'une couche de Dropout afin d'éviter l'overfitting.

Quant à l'étape de classification, nous avons inséré une couches Dense constituée de 128 neurones, et le résultat final est donné sous la forme d'une couche DENSE constituée de 3 neurones avec une fonction d'activation Softmax, qui classe les images résultantes de 3 catégories (Covid , normal et pneumonie).

Nous avons également utilisé Grad-CAM afin de clarifier les zones importantes de l'image sur lesquelles le modèle s'est appuyé pour la prédiction.

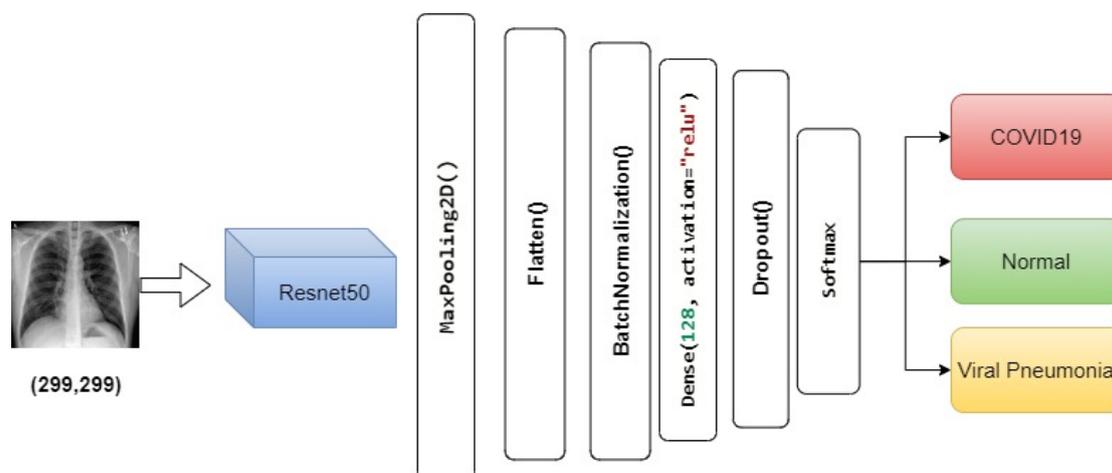


FIGURE 3.7 – Notre architecture basé sur ResNet50.

**Modèle DenseNet121** : Le processus de cette architecture est comme suit : Le modèle DenseNet121 reçoit en entrée une image (299, 299, 3) de notre jeu de données, l'extraction des caractéristiques se fait en utilisant plusieurs caractéristiques extraites par ce réseau.

Ces cartes de caractéristiques sont transmises à une couche GlobalAveragePooling2D pour prendre la moyenne de chaque carte de caractéristiques et à une couche Flatten qui modifie la forme des données en une entité unidimensionnelle.

La partie classification se compose de trois couches Dense la première avec 256 neurones, la deuxième avec 64 neurones et la troisième couche avec trois neurones avec une fonction d'activation Softmax qui est responsable de la classification finale.

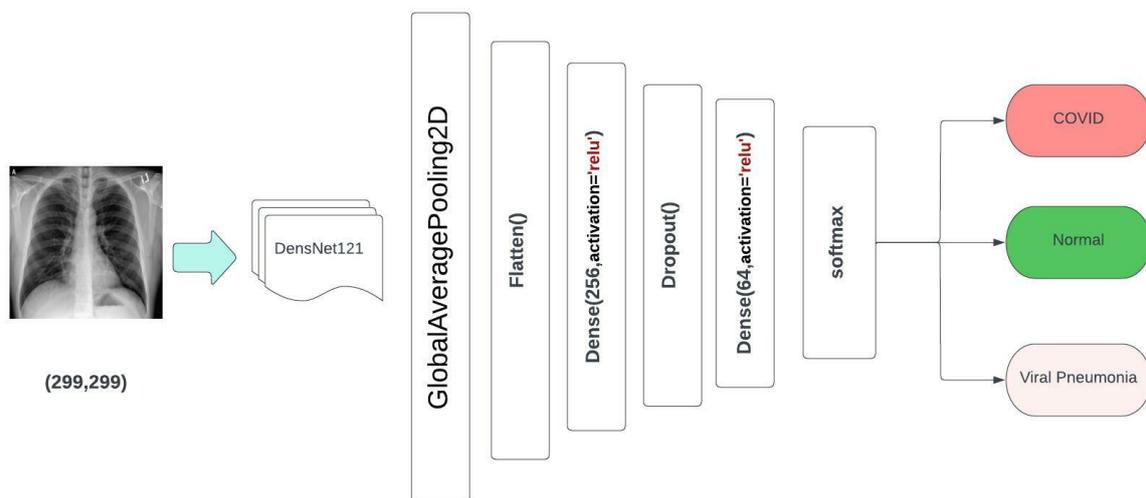


FIGURE 3.8 – Notre architecture basé sur DenseNet121.

**Modèle Xception** : Nous avons construit un modèle basé sur CNN basé sur un modèle Xception pré-entraîné .Tout d'abord, la couche d'entrée du modèle reçoit une image (299\*299\*3) et les caractéristiques sont extraites au niveau Xception, La sortie de cette couche est transmise à la couche AveragePooling2D Suivi par la couche Flatten qui modifie la forme des données en une entité unidimensionnelle. Quant à l'étape de classification, elle est composée d'une couche de Dense constituée de 128 neurones avec une couche de dropout afin d'éviter l'overfitting et d'une couche de sortie de résultat final Dense avec une fonction d'activation Softmax, qui classe les images résultantes de 3 catégories (Covid, normal et pneumonie).

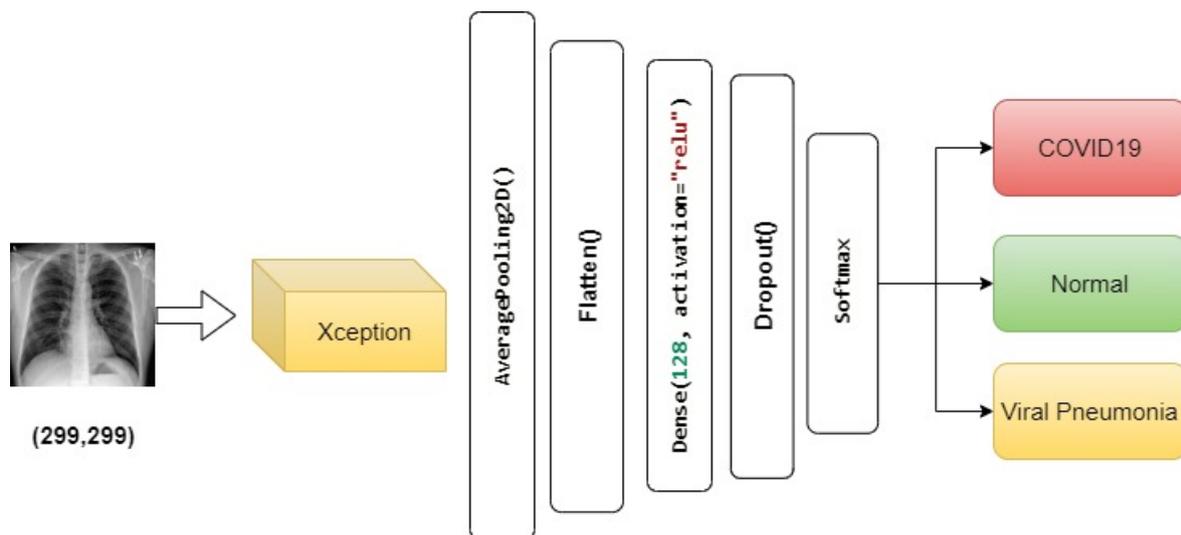


FIGURE 3.9 – Notre architecture basé sur Xception.

### Modèle de concaténation entre deux architectures (Xception et ResNet50V2) :

Dans ce modèle, nous avons concaténé les deux architectures Xception et ResNet50V2, où chaque modèle crée une carte de caractéristiques dans la dernière couche pour extraire les caractéristiques de l'image d'entrée de taille (299 \* 299 \* 3),

Ensuite, la sortie de ce modèle est envoyée à deux couches Convolution afin de mieux identifier les caractéristiques extraites des deux architectures concaténées. Chaque couche est suivie d'une couche MaxPooling2D pour prendre le maximum parti de chaque carte de statistiques, suivi par 2 couches Dropout et une couche Flatten.

Pour la classification, nous avons appliqué une couche Dense contenant 124 neurones et une couche Dense contenant 3 neurones extraits avec une fonction d'activation Softmax.

Et pour clarifier les zones importantes de l'image sur lesquelles le modèle s'est appuyé dans le processus de prédiction, nous avons utilisé Grad-CAM. Nous avons également utilisé dans la prédiction le Grad-CAM pour clarifier les zones importantes de l'image sur lesquels le modèle s'est appuyé pour donner le résultat.

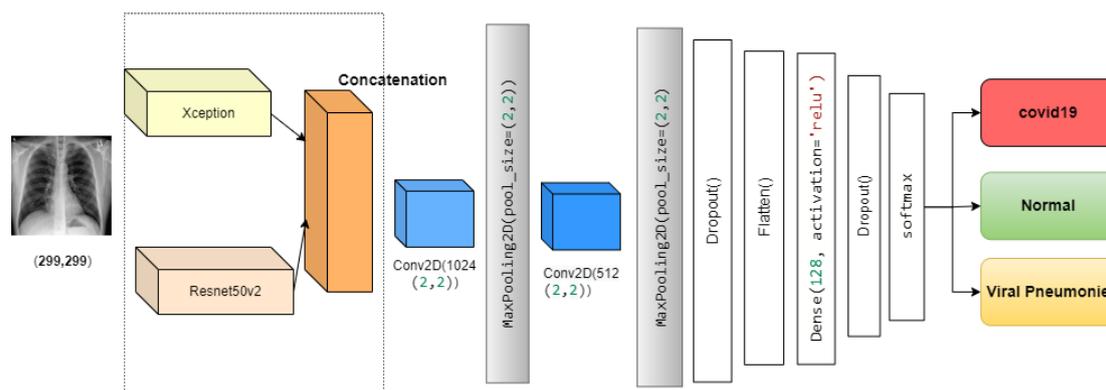


FIGURE 3.10 – Notre architecture basé sur Xception concatiné avec ResNet50V2.

**Entraînement de nos modèles :** Puisque nous travaillons sur un ensemble de données contenant une petite quantité d'images thoraciques (X ray) et que notre objectif est évité le surapprentissage et avoir bonne accuracy, cela nous a incités à utiliser l'apprentissage par transfert (transfer learning), que nous avons déjà présenté au chapitre 2.

Lors de l'entraînement de nos modèles, nous avons utilisé de nombreux paramètres tels que la fonction de perte et l'optimiseur ... où la fonction de perte est un élément essentiel de l'entraînement des modèles, et la fonction que nous avons utilisée est 'categorical\_crossentropy', qui est une fonction de perte utilisée dans tâches de classification multi-classes [74], Quant à la fonction d'optimisation, qui est utilisée pour réduire les erreurs et augmenter l'efficacité de la production, nous avons utilisé les deux fonctions Adam et SGD, où la fonction Adam est une extension de la descente de gradient stochastique, qui est l'abréviation pour « estimation du moment adaptatif », et il ajoute des fractions des gradients précédents au gradient actuel [75], et le SGD est Stochastic Gradient, signifie un système qui est connecté ou lié avec une probabilité aléatoire[76].

Dans notre travail, nous avons également utilisé Early Stopping, qui est une méthode qui vous permet de spécifier un nombre arbitrairement grand d'époques d'entraînement et d'arrêter l'entraînement dès que les performances du modèle cessent de s'améliorer dans l'ensemble de données de validation. Enfin, nous avons utilisé ReduceLRonPlateau, qui est un callback pour réduire le taux d'apprentissage lorsqu'une métrique a cessé de s'améliorer.

## Étape 2 :

**Big Transfer :** Big Transfer (BiT) est un modèle qui entraîne sur un grand ensemble de données source supervisé et affine les pondérations sur la tâche cible. Les modèles sont entraînés sur le jeu de données JFT-300M. Les modèles affinés contenus dans cette collection sont affinés sur ImageNet [77].

**Modèle Big Transfer :** dans cette étape nous avons utilisé l'apprentissage par transfert avec une grande base de données. Il a été entraîné deux fois, une sur ImageNet1k et une sur ImageNet21k.

Le modèle reçoit une image de taille (299 299,3) en entrée, puis passe l'image sur les couches du modèle r50x3h dont le travail consiste à pré-entraîner le modèle disponible pour publication sur ImageNet qui comprend des images de différentes classes.

Et puis nous ajoutons une couche BatchNormalization qui améliore et accélère l'entraînement en plus de la couche Flatten et de la couche Dropout afin de reconfigurer les données et d'éviter de forcer l'overfitting , respectivement. Ensuite, il passe à travers la couche Dense contenant 256 neurones avec la fonction d'activation ReLU pour nous donner le résultat final via la couche Dense avec 3 neurones avec la fonction d'activation Softmax en sortie du processus de classification.

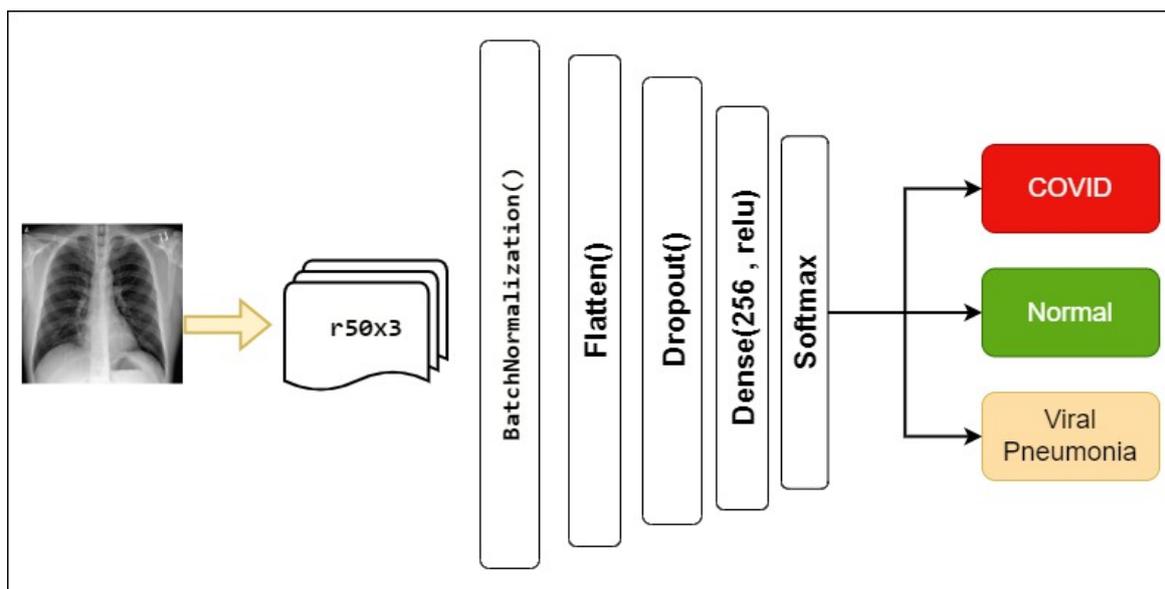


FIGURE 3.11 – Notre architecture basé sur Big Transfer.

## Approche 2 :

**Vision Transformer :** Le transformateur de vision (ViT) est un transformateur utilisé dans le domaine de la vision par ordinateur qui fonctionne sur la base de la nature fonctionnelle des transformateurs utilisés dans le domaine du traitement du langage naturel. En interne, le transformateur apprend en mesurant la relation entre les paires de jetons d'entrée. En vision par ordinateur, nous pouvons utiliser les patches d'images comme jeton [78].

**Modèle Vision Transformer :** Dans cette étape, nous avons utilisé le modèle Vision transformer ViT. Premièrement la couche d'entrée du modèle reçoit une image  $(299 \times 299 \times 3)$ , Nous passons l'image dans deux blocs de convolution pour réduire les dimensions de l'image.

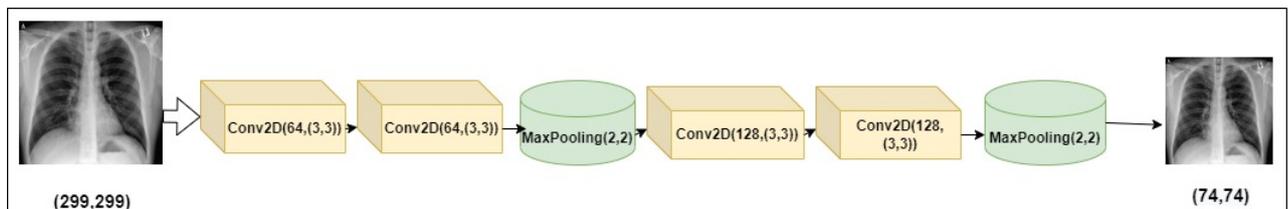


FIGURE 3.12 – la méthode de redimensionnement d'image en fonction de CNN.

Après avoir terminé la première étape, nous divisons l'image avec les dimensions  $(74, 74)$  en un ensemble de patches.

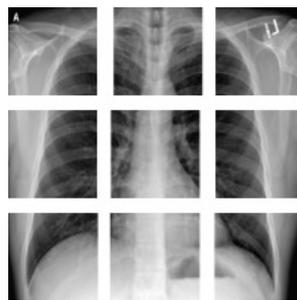
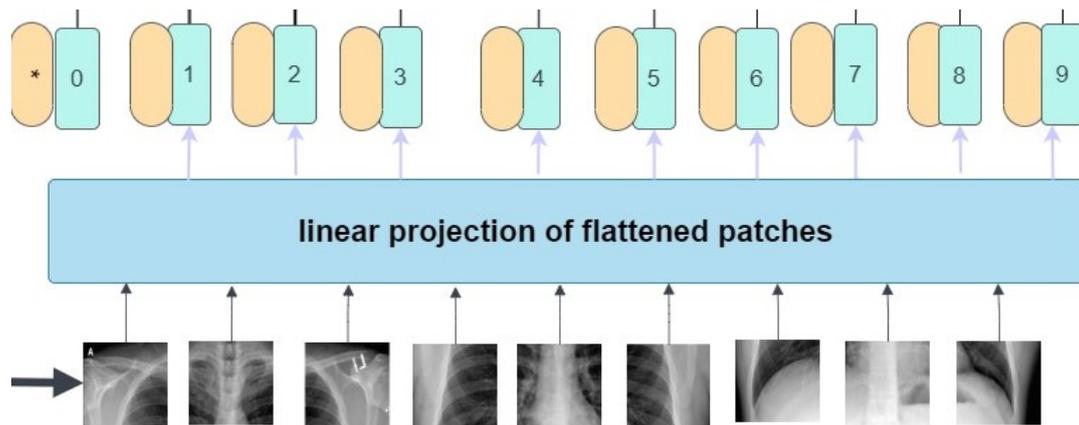


FIGURE 3.13 – la division d'image en patches.

Ensuite, nous passons les patches en projection linéaire, ce qui nous donne un vecteur des patches avec l'indice de chaque patche.



Un encodeur de transformateur le reçoit et le fait passer à travers un ensemble de couches La couche BatchNormalization reçoit la correction avec son index puis la passe à Attention multi-tête, puis on saute la connexion, puis vient une autre couche de BatchNormalization, suivi de Bloc perceptron multicouche.

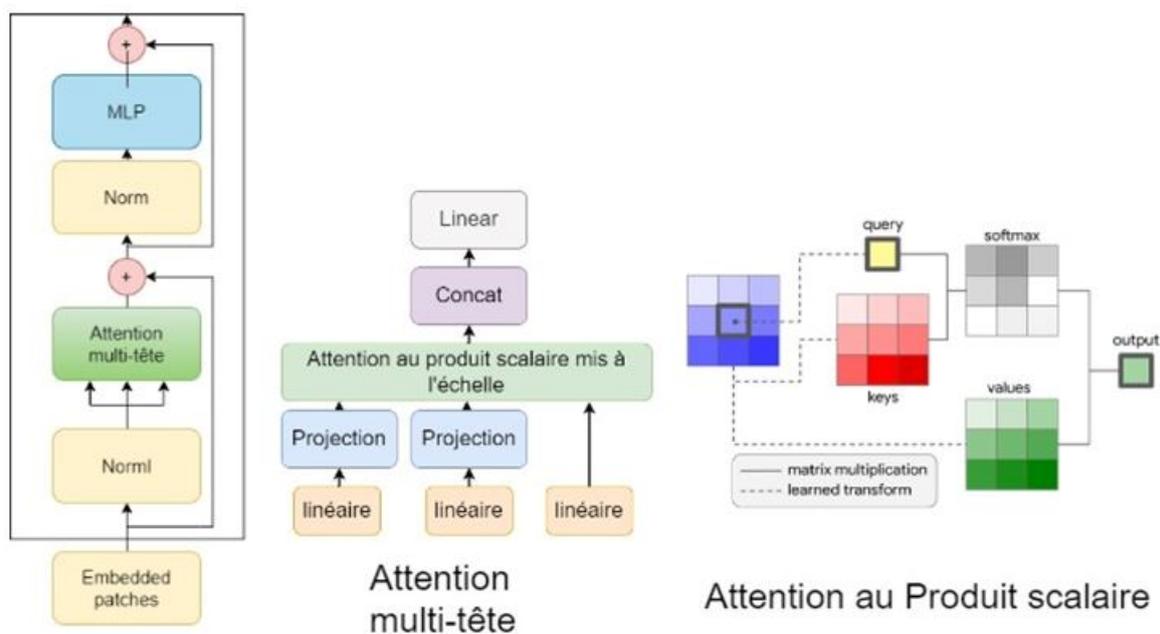


FIGURE 3.14 – L’architecture de l’encodeur du transformateur.

La sortie de transformer encoder transmette à une couche BatchNormalization pour optimiser les paramètres du modèle et améliorer la vitesse de l’entraînement. Suivre par une couche Flatten pour convertir la forme des données en un vecteur unidimensionnel, suivre d’une couche Dropout en suit la partie classification est un bloc MLP qui contient une couche Dense avec 100 neurones suivre par dropout. La sortie finale est produite à

partir d'une couche Dense avec 3 neurones qui classe l'image de sortie dans l'une des classes COVID-19, pneumonie virale ou normal.

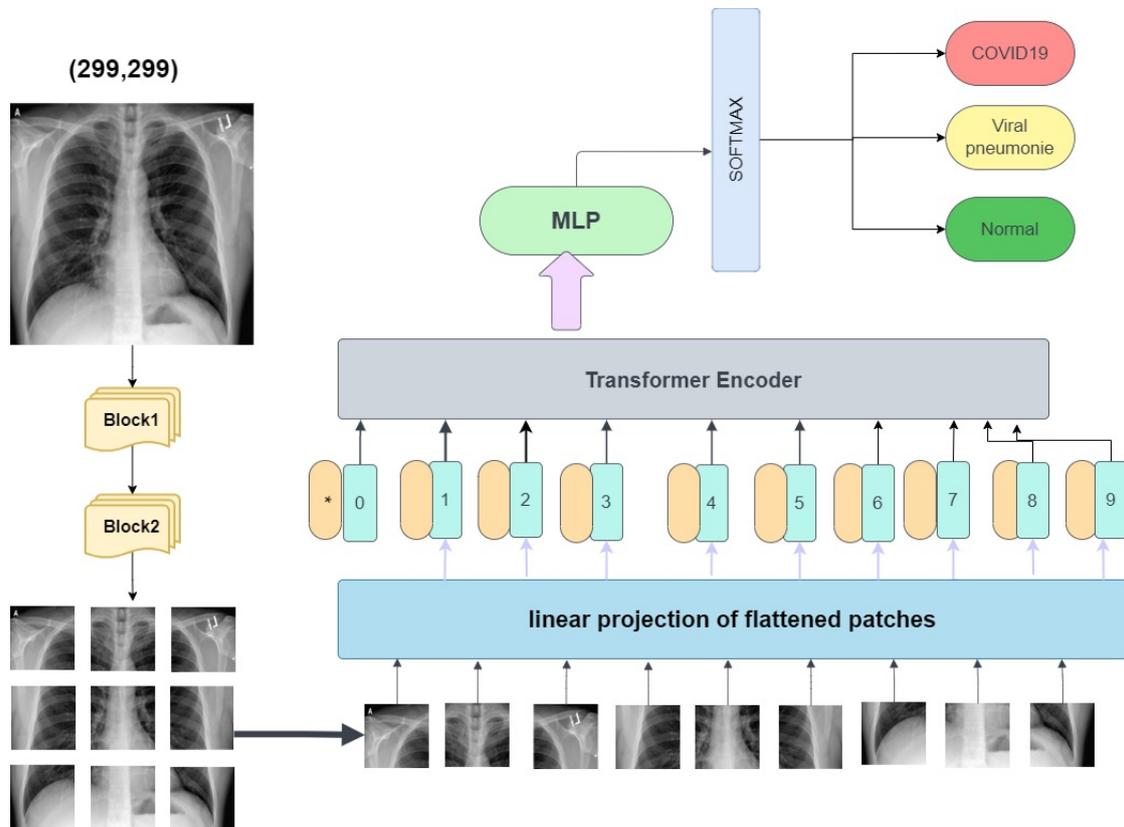


FIGURE 3.15 – l'architecture globale de le Vision transformer (ViT).

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté l'ensemble de données utilisé dans notre travail et détaillé la solution proposée.

Dans le chapitre suivant. Nous montrerons d'abord les différents outils de programmation utilisés dans notre travail, puis nous présenterons les résultats de l'implémentation.

## Résultats et discussions

### 4.1 Introduction

Après avoir présenté l'approche que nous avons proposée dans le chapitre précédent, nous consacrerons cette partie à l'implémentation et à la discussion des méthodes proposées précédemment.

Dans ce chapitre, nous parlerons de tout ce qui concerne la réalisation de notre travail depuis l'environnement de développement ainsi que les bibliothèques utilisées pour implémenter notre modèle, enfin, nous discuterons des résultats obtenus.

### 4.2 Environnement de travail

Dans cette partie, nous allons vous présenter les langages de programmation avec les programmes et bibliothèques utilisés dans notre travail.

#### 4.2.1 Le langage de programmation

**Python** Dans ce travail nous avons utilisé le langage de programmation Python qui est un langage de programmation informatique souvent utilisé pour créer des sites Web et des logiciels, automatiser des tâches et effectuer des analyses de données. Python est un langage à usage général, ce qui signifie qu'il peut être utilisé pour créer une variété de programmes différents et qu'il n'est pas spécialisé pour des problèmes spécifiques. Il s'agit d'un langage de programmation interprété, il n'a donc pas besoin d'être compilé

pour fonctionner, interactif et orienté objet avec une sémantique dynamique.

Pouvez-vous faire avec python Certaines choses incluent :

- Analyse de données et apprentissage automatique.
- Développement web.
- Automatisation ou script.
- Test et prototypage de logiciels.
- Tâches quotidiennes.

## 4.2.2 Les bibliothèques utilisées

**TensorFlow** : TensorFlow est une bibliothèque open source destinée à l'informatique numérique rapide. Il supporte aussi l'apprentissage automatique traditionnel. TensorFlow a été initialement mis au point pour les grands calculs numériques sans tenir compte de l'apprentissage profond. Toutefois, il s'est aussi révélé très utile pour le développement du l'apprentissage profond, et Google l'a donc mis en open source.

TensorFlow accepte les données sous la forme de tableaux multidimensionnels de dimensions supérieures appelés tensors. Les tableaux multidimensionnels sont très pratiques pour gérer de grandes quantités de données.

Il peut fonctionner sur des CPU, des GPU ainsi que des appareils mobiles et les systèmes distribués.

**Keras** : Keras est une API (Application programming interface) de réseau neuronal de haut niveau conçue pour les humains, pas pour les machines. Est considéré comme une puissante de la bibliothèque Python, facile à utiliser pour développer et évaluer des modèles d'apprentissage profond. Keras suit les meilleures pratiques pour réduire la charge cognitive : il fournit des API cohérentes et simples, réduit le nombre d'actions de l'utilisateur requises pour les cas d'utilisation courants et fournit des messages d'erreur clairs et exploitables. Il comprend également une documentation complète et des guides de développement.

**Matplotlib** : Matplotlib est une bibliothèque multiplateforme de visualisation de données et de traçage graphique pour Python. Les développeurs peuvent également uti-

liser les interfaces de programmation d'applications (API) matplotlib pour intégrer les tracés dans les applications graphiques.

Il existe de nombreuses bibliothèques présentes dans Python. Et Matplotlib est l'une des bibliothèques les plus réussies et les plus utilisées, qui fournit divers outils de visualisation de données en Python.

C'est l'une des bibliothèques de traçage les plus puissantes de Python. Il s'agit d'une bibliothèque multiplateforme qui fournit divers outils pour créer des tracés 2D à partir des données dans des listes ou des tableaux en python.

**NumPy** : NumPy est une forme abrégée de Numerical Python, qui est appliquée à la programmation scientifique en Python, en particulier pour les nombres. Il comprend des objets multidimensionnels dans des tableaux et un ensemble d'outils d'intégration pour l'implémentation de Python. Il s'agit essentiellement d'un mélange de C et de Python, où les données sous forme de chiffres sont traitées comme des tableaux pour les fonctions multidimensionnelles et les opérations de réarrangement.

**Pandas** : Pandas est une bibliothèque Python pour l'analyse de données. Initié par le besoin d'un outil d'analyse quantitative puissant et flexible, pandas est devenu l'une des bibliothèques Python les plus populaires. Il a une communauté de contributeurs très active.

**Scikit-learn** : Scikit-learn est une bibliothèque clé d'analyse de données open source et la référence en matière d'apprentissage automatique (ML). Pour le langage de programmation Python.

Scikit-learn se concentre sur les outils d'apprentissage automatique, y compris les algorithmes mathématiques, statistiques.

### 4.2.3 La plateforme et environnements de développement

**Google Colaboratory notebook** : Google Colaboratory a été développé par Google pour fournir un accès gratuit aux GPU et aux TPU à toute personne qui en a besoin pour créer un modèle d'apprentissage automatique ou d'apprentissage profond. Google Colab peut être défini comme un environnement de bloc-notes hébergé par Jupyter Notebook.

Vous pouvez dire que Google Colab est une version améliorée de jupyter.

Il vous permet de programmer dans le langage de programmation Python ou de développer des applications d'apprentissage profond à l'aide de bibliothèques populaires telles que Keras, TensorFlow, PyTorch et OpenCV.

## 4.3 Indicateurs de performances de la classification

Afin que les performances du modèle soient correctement analysées. Nous créons une matrice de confusion.

### 4.3.1 Matrice de confusion

Une matrice de confusion est un résumé tabulaire du nombre de prédictions correctes et incorrectes faites par un classifieur. Il est utilisé pour mesurer les performances d'un modèle de classification. Afin que les performances du modèle soient correctement analysées. Nous créons une matrice de confusion.. Il peut être utilisé pour évaluer les performances d'un modèle de classification grâce au calcul de métriques de performance.

Elle représente les différentes combinaisons de valeurs réelles et prédites.

**True positive (TP)** : les valeurs qui étaient réellement positives et qui ont été prédites positives.

**False Positive (FP)** : Les valeurs qui étaient en fait négatives mais faussement prédites comme positives. Aussi connu sous le nom d'erreur de type I.

**False Négative (FN)** : Les valeurs qui étaient réellement positives mais faussement prédites comme négatives. Aussi connu sous le nom d'erreur de type II.

**True Négative (TN)** : Les valeurs qui étaient réellement négatives et prédites négatives.

		Valeurs de prédictions	
		Positive	Négative
Valeurs réelles	Positive	TP	FN
	Négative	FP	TN

FIGURE 4.1 – Matrice de confusion.

A partir de cette matrice, nous avons les calculs des indicateurs suivants :

Métriques	Description	Formule
Précision	La précision vérifie combien de résultats sont réellement des résultats positifs sur le total des résultats prédits positivement.	$precision = \frac{TP}{TP+FP}$
sensibilité ou rappel (recall)	Le rappel est la mesure permettant de vérifier correctement les résultats prédits positifs sur le nombre total de résultats positifs.	$recall = \frac{TP}{TP+FN}$
moyenne harmonique (F1-score)	F1-score est la moyenne harmonique de la précision et le rappel.	$F1 = 2 \frac{precision * recall}{precision + recall}$
Accuracy	indique le pourcentage de bonnes prédictions, Il est calculé en divisant le nombre total de prédictions correctes par toutes les prédictions.	$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

TABLE 4.1 – Métrique d'évaluation de la classification.

## 4.4 Résultats et discussion

La performance des modèles de formation a été évaluée sur la base de plusieurs paramètres importants, qui sont la précision de l'entraînement (training accuracy), la précision de la validation (validation accuracy), la perte de l'entraînement (training loss) et la perte de validation (validation loss) sur plusieurs époques.

Nous présenterons les résultats de la précision de l'entraînement et des valeurs de perte pour tous les modèles dans la figure suivante :

<b>Resnet50</b>	
Epoch 1/10	100/100 [=====] - 83s 797ms/step - loss: 0.4385 - accuracy: 0.8453 - val_loss: 0.9794 - val_accuracy: 0.6042
Epoch 10/10	100/100 [=====] - 76s 763ms/step - loss: 0.1893 - accuracy: 0.9319 - val_loss: 0.2517 - val_accuracy: 0.9349
<b>DenseNet121</b>	
Epoch 1/10	100/100 [=====] - 863s 8s/step - loss: 0.6318 - accuracy: 0.7557 - val_loss: 0.3860 - val_accuracy: 0.8542
Epoch 10/10	100/100 [=====] - 858s 9s/step - loss: 0.1971 - accuracy: 0.9330 - val_loss: 0.2146 - val_accuracy: 0.9193
<b>Xception</b>	
Epoch 1/10	100/100 [=====] - 1003s 10s/step - loss: 0.3712 - accuracy: 0.8509 - val_loss: 0.2511 - val_accuracy: 0.9167
Epoch 10/10	100/100 [=====] - 1001s 10s/step - loss: 0.1543 - accuracy: 0.9404 - val_loss: 0.1828 - val_accuracy: 0.9245
<b>Concatenation</b>	
Epoch 1/10	32/32 [=====] - 1001s 31s/step - loss: 0.7092 - accuracy: 0.6609 - val_loss: 0.5720 - val_accuracy: 0.8906 - lr: 1.0000e-04
Epoch 10: ReduceLROnPlateau reducing learning rate to 8.99999972648877e-06.	
32/32 [=====] - 972s 31s/step - loss: 0.0936 - accuracy: 0.9678 - val_loss: 0.1735 - val_accuracy: 0.9349 - lr: 3.0000e-05	
<b>Big-transfer avec imagenet1k</b>	
Epoch 1/10	100/100 [=====] - 451s 4s/step - loss: 0.5652 - accuracy: 0.7784 - val_loss: 0.4846 - val_accuracy: 0.8906
Epoch 10/10	100/100 [=====] - 116s 1s/step - loss: 0.2273 - accuracy: 0.9281 - val_loss: 0.2644 - val_accuracy: 0.9167
<b>Big-transfer avec imagenet21k</b>	
Epoch 1/10	100/100 [=====] - 665s 6s/step - loss: 0.3798 - accuracy: 0.8625 - val_loss: 0.3845 - val_accuracy: 0.8594
Epoch 10/10	100/100 [=====] - 107s 1s/step - loss: 0.1499 - accuracy: 0.9549 - val_loss: 0.2015 - val_accuracy: 0.9297
<b>Vision Transformer</b>	
Epoch 1/10	(None, None, 192)
(None, None, 192)	
100/100 [=====] - ETA: 0s - loss: 0.7680 - accuracy: 0.6603	(None, None, 192)
100/100 [=====] - 140s 812ms/step - loss: 0.7680 - accuracy: 0.6603 - val_loss: 2.3131 - val_accuracy: 0.3047 - lr: 1.0000e-04	
Epoch 10/10	
100/100 [=====] - 82s 820ms/step - loss: 0.2726 - accuracy: 0.9085 - val_loss: 0.3446 - val_accuracy: 0.8594 - lr: 9.0000e-06	

FIGURE 4.2 – Aperçu sur les résultats initiaux et finals de chaque modèle.

Afin de montrer les résultats obtenus pour notre modèle, on illustre dans ce qui suit les résultats en termes de précision et d'erreur de chaque modèle. Ensuite, nous montrerons la matrice de confusion de chacun de nos modèles proposés en terme d'évaluation des performances de classification qui nous donnera les résultats de précision, de rappel et de F1-score, puis nous comparerons tous les modèles que nous avons créés.

## Approche 1 :

### Étape 1 :

Modèle de ResNet50 :

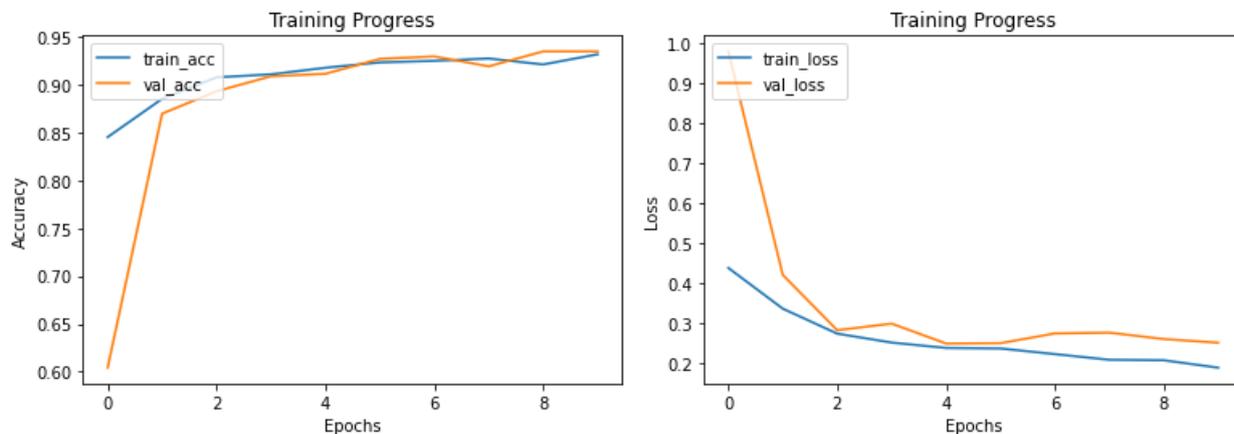


FIGURE 4.3 – Courbes accuracy et loss obtenus pour le modèle ResNet50.

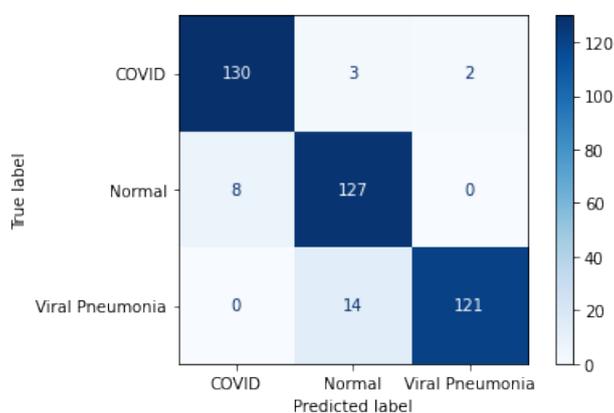


FIGURE 4.4 – Matrice de confusion pour le modèle ResNet50.

classe	précision	recall	F1-score	accuracy
COVID-19	94%	96%	95%	92.29%
Normal	88%	94%	91%	92.29%
Pneumonie Viral	98%	90%	94%	92.29%

TABLE 4.2 – Mesure de performance du modèle ResNet50.

**Modèle de Densenet121 :**

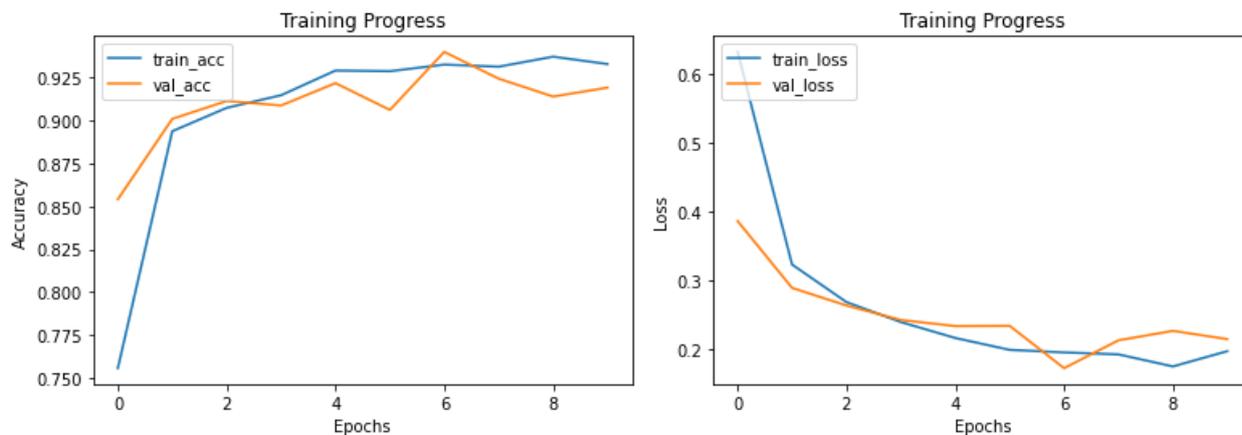


FIGURE 4.5 – Courbes accuracy et loss obtenus pour le modèle DenseNet121.

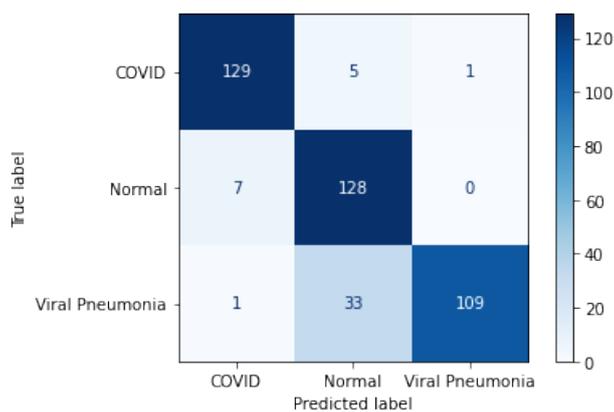


FIGURE 4.6 – Matrice de confusion pour le modèle DenseNet121.

classe	précision	recall	F1-score	accuracy
COVID-19	94%	96%	95%	93.22%
Normal	77%	95%	85%	93.22%
Pneumonie Viral	99%	76%	86%	93.22%

TABLE 4.3 – Mesure de performance du modèle DenseNet121.

Modèle de Xception :

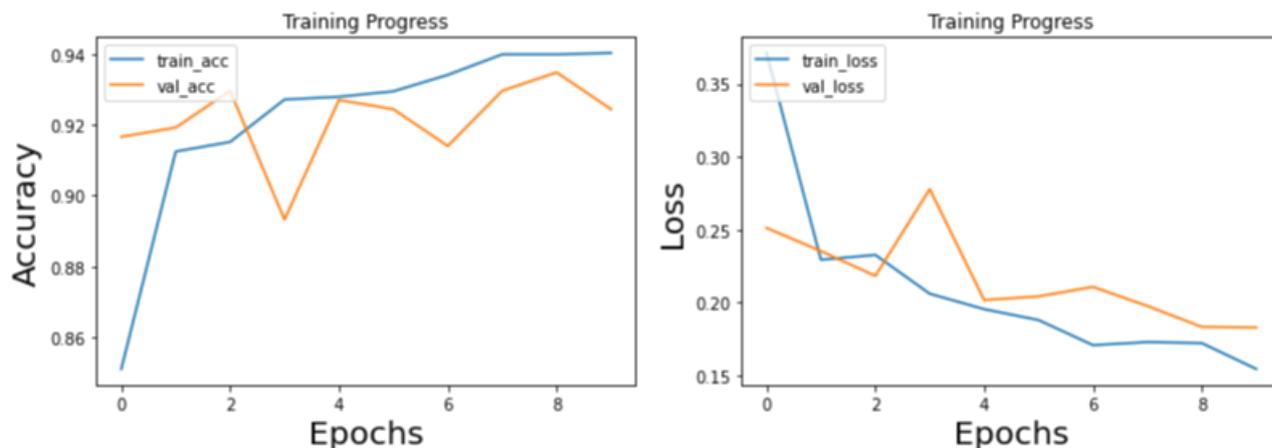


FIGURE 4.7 – Courbes accuracy et loss obtenus pour le modèle Xception.

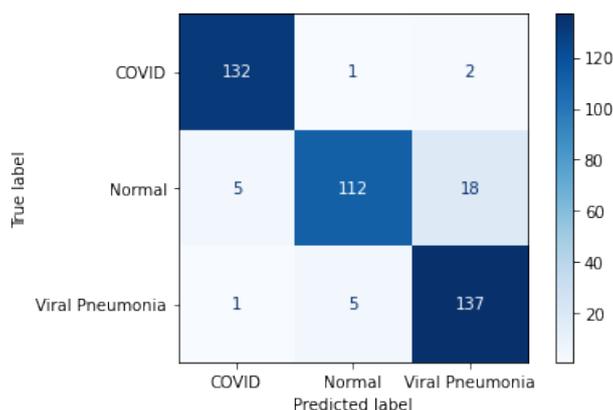


FIGURE 4.8 – Matrice de confusion pour le modèle Xception.

classe	précision	recall	F1-score	accuracy
COVID-19	96%	98%	97%	93.22%
Normal	97%	83%	89%	93.22%
Pneumonie Viral	85%	96%	91%	93.22%

TABLE 4.4 – Mesure de performance du modèle Xception.

Modèle de concaténation (Resnet50 + Xception) :

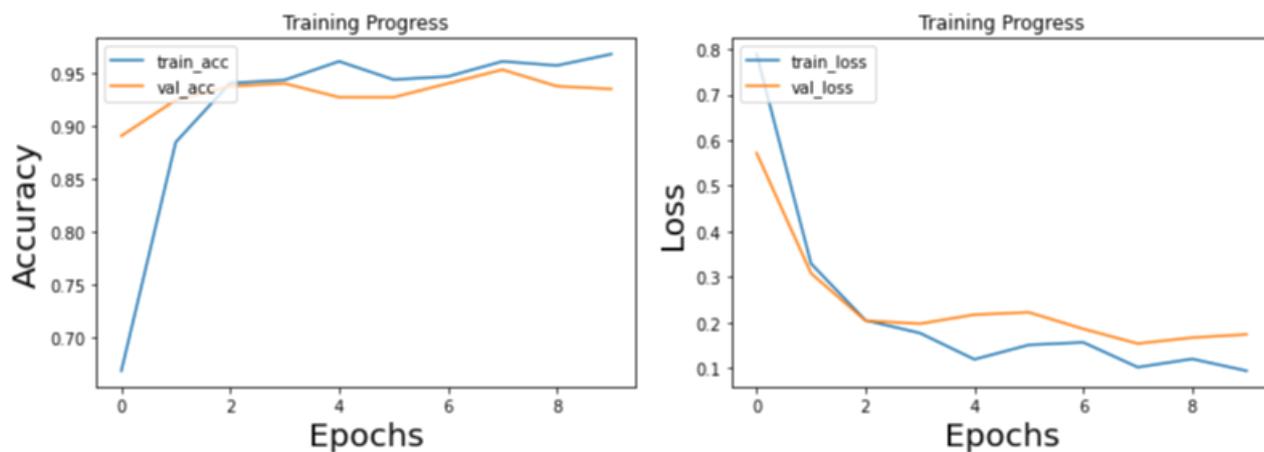


FIGURE 4.9 – Courbes accuracy et loss obtenus pour le modèle concaténation (Resnet50 + Xception).

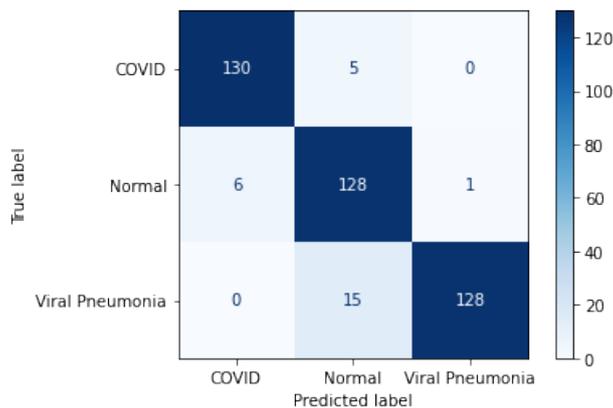


FIGURE 4.10 – Matrice de confusion pour le modèle concaténation (Resnet50 + Xception).

classe	précision	recall	F1-score	accuracy
COVID-19	96%	96%	96%	94.43%
Normal	86%	95%	90%	94.43%
Pneumonie Viral	99%	90%	94%	94.43%

TABLE 4.5 – Mesure de performance du modèle concaténation (Resnet50 + Xception).

### Comparaison entre les modèles d'étape 1 :

Modèles	précision	recall	F1-score	accuracy
ResNet50	94%	93%	93%	92.29%
DenseNet121	90%	89%	89%	93.22%
Xception	93%	92%	92%	93.22%
Concaténation (ResNet50 et Xception)	94%	94%	94%	94.43%

TABLE 4.6 – comparaison entre les modèles d'étape 1.

**Discussion :** Après avoir comparé les modèles de la première étape, il a été constaté que le modèle de concaténation entre ResNet et Xception donnait les meilleurs résultats en termes d'accuracy, de rappel, de précision et de f1-score. D'ailleurs, la concaténation a été choisie entre ResNet et Xception, et non entre Xception et DenseNet, par exemple, car ResNet a un meilleur f1-score et un meilleur rappel, et puisque le rappel est important dans le domaine médical, où dans notre cas la technique de concaténation permet de s'affranchir du problème de contrôle des hyperparamètres. Il permet également de tirer partie des avantages de chaque modèle.

### Étape 2 :

#### Modèle de BiT + ImageNet1k :

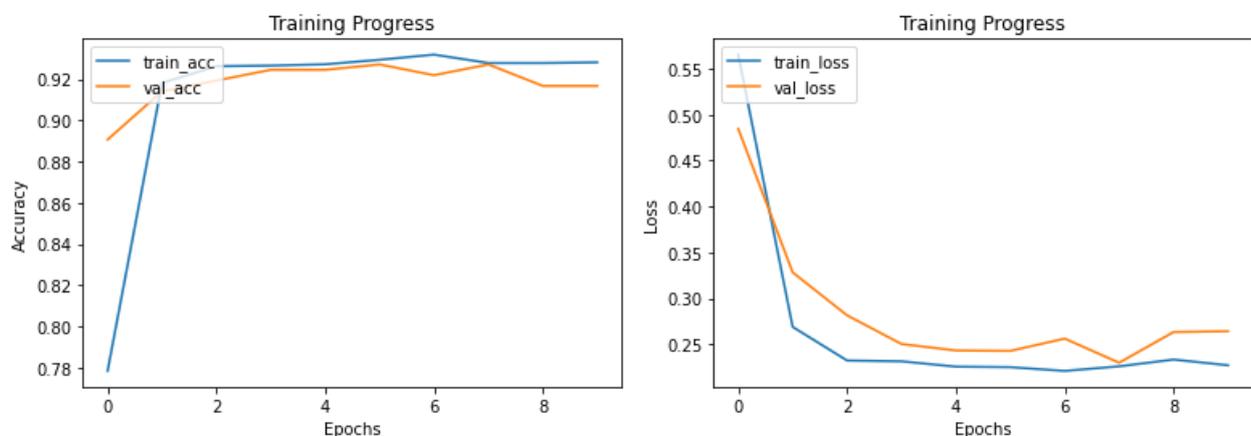


FIGURE 4.11 – Courbes accuracy et loss obtenus pour le modèle BiT avec ImageNet1k.

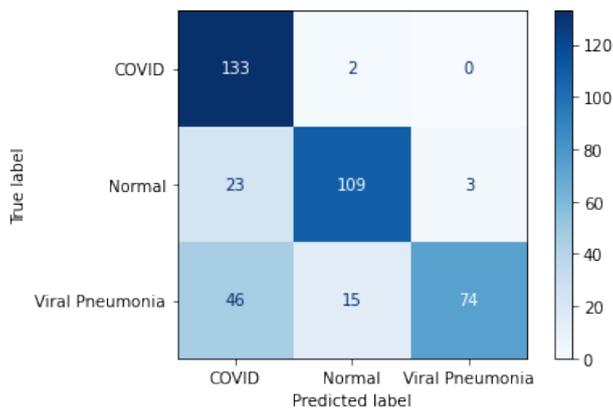


FIGURE 4.12 – Matrice de confusion pour le modèle BiT avec ImageNet1k.

classe	précision	recall	F1-score	accuracy
COVID-19	66%	99%	79%	92.10%
Normal	87%	81%	84%	92.10%
Pneumonie Viral	96%	55%	70%	92.10%

TABLE 4.7 – Mesure de performance du modèle BiT avec ImageNet1k.

**Modèle de BiT + ImageNet21k :**

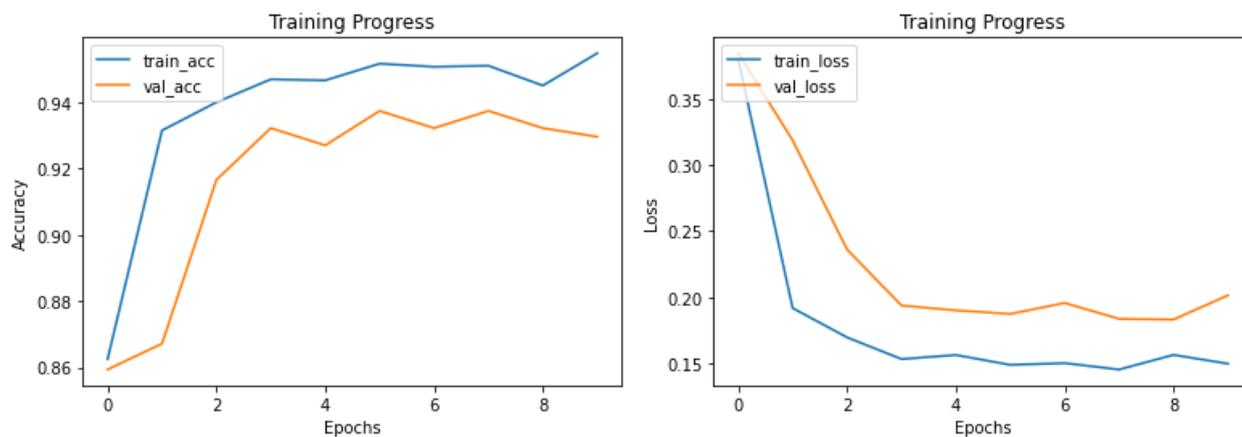


FIGURE 4.13 – Courbes accuracy et loss obtenus pour le modèle BiT avec ImageNet21k.

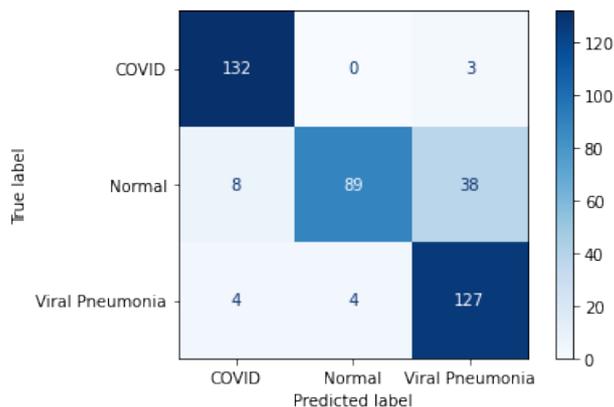


FIGURE 4.14 – Matrice de confusion pour le modèle BiT avec ImageNet21k.

classe	précision	recall	F1-score	accuracy
COVID-19	92%	98%	95%	93.33%
Normal	96%	66%	78%	93.33%
Pneumonie Viral	76%	94%	84%	93.33%

TABLE 4.8 – Mesure de performance du modèle BiT avec ImageNet21k.

#### Comparaison entre les modèles d'étape 2 :

Modèles	précision	recall	F1-score	accuracy
BiT avec ImageNet1k	83%	78%	77%	92.10%
BiT avec ImageNet21k	88%	86%	86%	93.33%

TABLE 4.9 – comparaison entre les modèles d'étape 2.

**Discussion :** Nous remarquons des résultats de ces deux modèles que l'augmentation de la taille de la base de données permet d'améliorer l'accuracy, le rappel, la précision et le f1-score, même si le pré-entraînement ne s'est pas effectué sur une base médicale.

Quant à la comparaison avec la première étape, le modèle de concaténation ( ResNet50 et Xception) a enregistré de meilleurs résultats que le BiT, et après avoir vu les résultats, nous n'avons pas utilisé le modèle concaténé pour étudier l'impact de la taille de la base de données car nous n'avons pas assez de ressources pour entraîner notre réseau sur ImageNet21k.

## Approche 2 :

### Modèle ViT

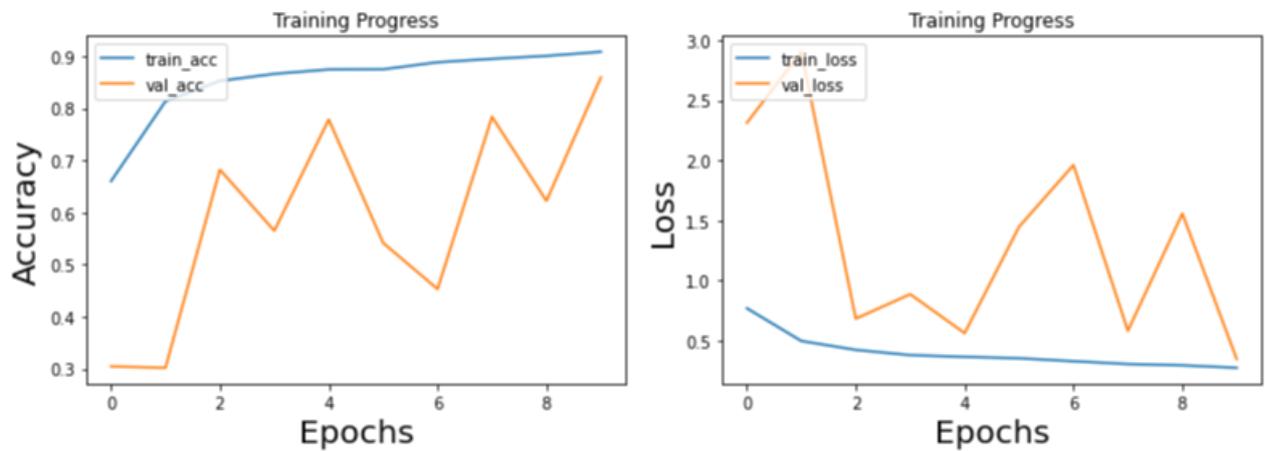


FIGURE 4.15 – Courbes accuracy et loss obtenus pour le modèle ViT.

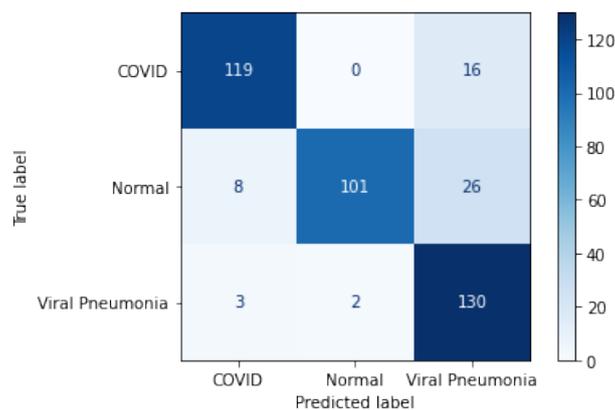


FIGURE 4.16 – Matrice de confusion pour le modèle ViT.

classe	précision	recall	F1-score	accuracy
COVID-19	92%	88%	90%	86.57%
Normal	98%	75%	85%	86.57%
Pneumonie Viral	76%	96%	85%	86.57%

TABLE 4.10 – Mesure de performance du modèle ViT.

**Discussion :** Quant à la troisième approche (ViT), elle donne des résultats acceptables, mais elle est inférieure aux autres modèles en termes d'accuracy, puisqu'elle nous a

donné un taux de 86,57 %. Cela est dû au fait que cette technique n'est pas courante dans le monde. Domaine de la classification des images médicales, qui pourra être amélioré à l'avenir.

## Comparaison entre nos modèles

à travers les tableaux et les figures ci-dessus, nous remarquons que le pourcentage d'accuracy, de rappel, de précision et de F1-score est élevé, bien que nous ayons utilisé un ensemble de données plutôt petit, c'est-à-dire que le taux d'erreur sera quelque peu réduit.

Modèles	précision	recall	F1-score	accuracy
<b>ResNet50</b>	94%	93%	93%	92.29%
<b>DenseNet121</b>	90%	89%	89%	93.22%
<b>Xception</b>	93%	92%	92%	93.22%
<b>Concaténation (ResNet50 et Xception)</b>	94%	94%	94%	94.43%
<b>BiT avec ImageNet1k</b>	83%	78%	77%	92.10%
<b>BiT avec ImageNet21k</b>	88%	86%	86%	93.33%
<b>ViT</b>	88%	86%	86%	96.57%

TABLE 4.11 – comparaison entre non modèles.

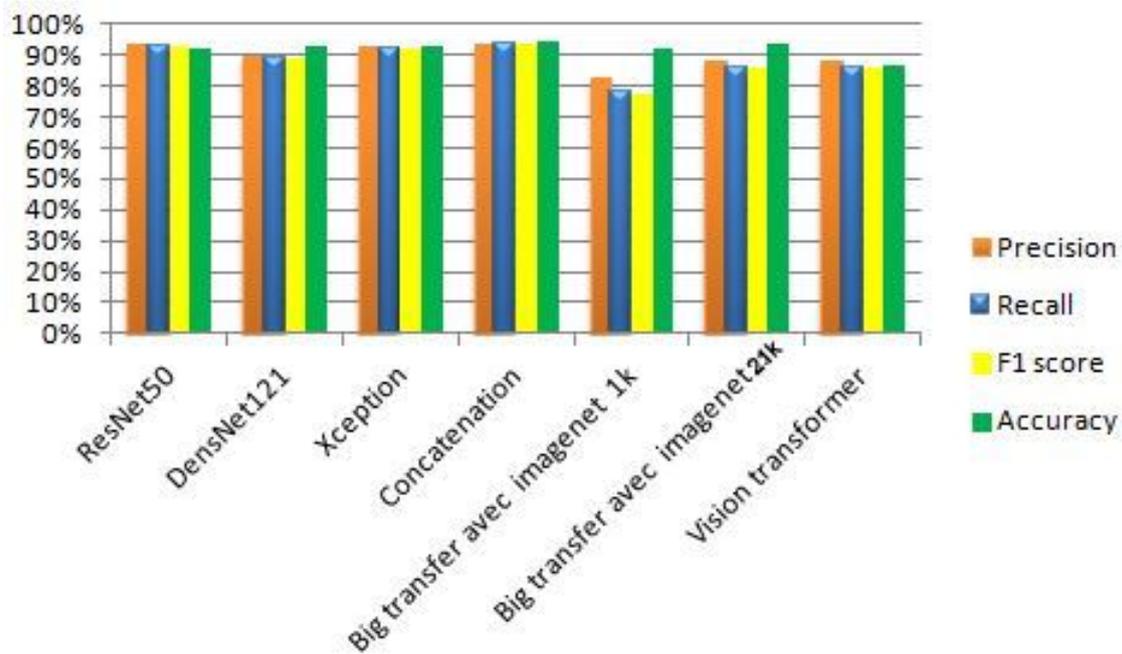


FIGURE 4.17 – comparaison entre non modèles.

En comparant nos modèles entre eux, nous remarquons qu'ils ont tous atteint un taux de précision élevé, et c'est le but de notre travail. Il s'avère que le modèle de concaténation (ResNet50 et Xception) est le plus précis en précision, suivi par le reste des modèles avec des pourcentages très proches. À peu près tous les ratios de tous les modèles sont proches. Le modèle de concaténation est le meilleur modèle dans notre travail car il nous permet de profiter des avantages de deux modèles dans un seul modèle, c'est-à-dire la combinaison, nous permet de mieux décrire nos images. Nous aurions pu également combiner le ViT avec un autre modèle mais on n'a pas assez de temps pour le faire.

#### 4.4.1 comparaison de performances entre notre modèle et autres travaux connexes

Étant donné que le covid-19 est une maladie moderne et qu'il y a une pénurie dans le nombre de travaux qui l'ont classé en (covid, normal, pneumonie virale). Nous avons recherché des travaux connexes qui travaillaient sur la même base de données pour faire une comparaison des résultats obtenus. Nous avons choisi le modèle de concaténation entre Xception ET ResNet50 car c'est notre meilleur modèle en matière d'accuracy.

recherche	précision	recall	F1-score	accuracy
Modèle basé sur MobilNetV1 [79]	93.24%	pas mentionner	pas mentionner	83.7%
Modèle basé sur MobilNetV2 [79]	pas mentionner	pas mentionner	pas mentionner	87.5%
Notre modèle (concaténation entre Xception et ResNet50)	94%	94%	94%	94.43%

TABLE 4.12 – Comparaison entre notre modèle proposé (DenseNet121 + Xception) et un travail de littérature.

Pour comparer notre travail avec d'autres travaux connexes, nous avons essayé de trouver des travaux qui travaillent sur la même base de données sur laquelle nous avons travaillé, mais l'absence de nom pour cette base nous a rendu difficile pour nous de trouver un travail qui travaille sur cette base, nous n'avons trouvé qu'un travail qui s'appuie sur MobileNetV1 et MobileNetV2, et nos résultats sont meilleurs qu'en termes de précision.

## 4.5 Prédiction

Dans l'étape de prédiction, nous avons utilisé la technique Grad-CAM afin de connaître les caractéristiques sur lesquelles les modèles se sont appuyés pour classer les images par catégories, et nous présenterons dans la figure ci-dessous quelques résultats après avoir sélectionné les images à classer aléatoirement.

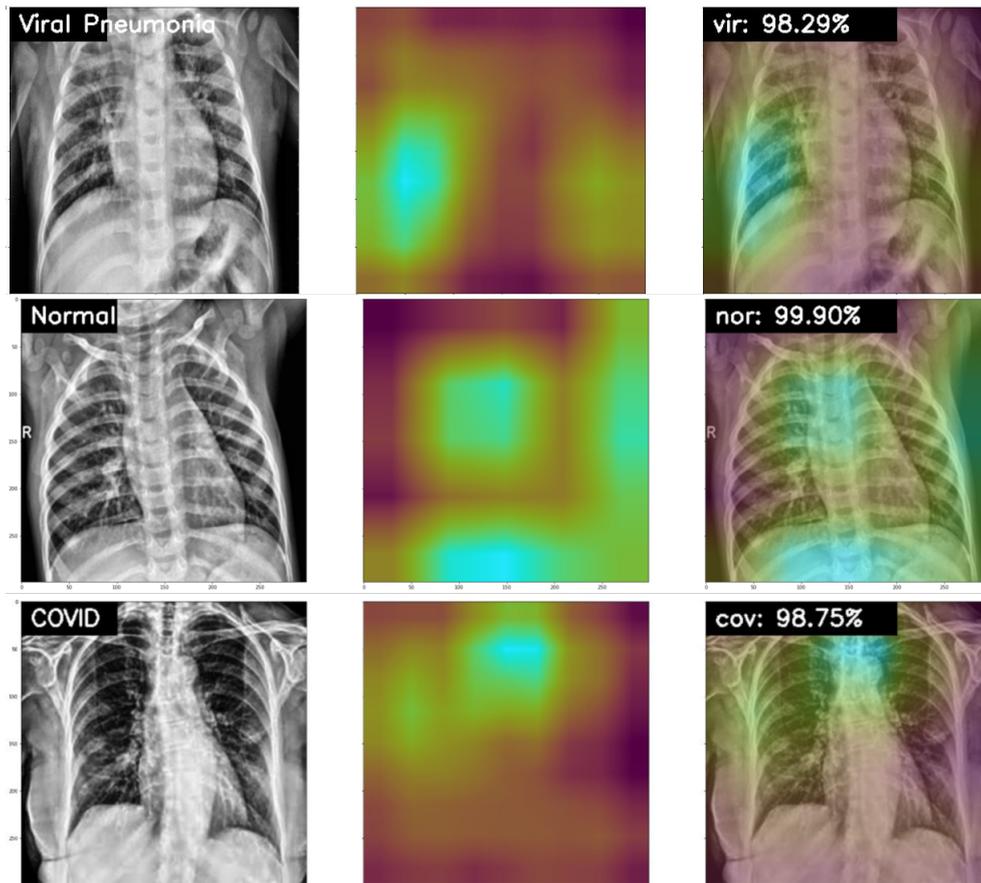


FIGURE 4.18 – Résultat de teste de notre modèle sur quelques exemple de teste avec l'utilisation de Grad-CAM.

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté les outils de programmation et couvert l'implémentation de nos modèles, et nous avons discuté également des résultats obtenus, où les choses se sont bien déroulées et nous avons obtenu des résultats encourageants. c'est notre première tentative dans ce domaine et nous espérons améliorer les performances de nos modèles et les rendre plus précis à l'avenir.

# Conclusion générale et perspectives

**Conclusion générale :** La classification des images est un aspect important de la vision par ordinateur.

Depuis que le monde a vécu une période difficile en raison de la pandémie de Corona, qui a connu une augmentation significative du nombre de personnes infectées par le virus et le problème du manque de kits de test et de la lenteur dans l'analyse des résultats de diagnostic, les chercheurs ont essayé de trouver des solutions simples et rapides pour faciliter la tâche des médecins.

Dans ce travail, nous avons proposé deux approches qui classent automatiquement le COVID-19 et la pneumonie à l'aide de réseaux neuronaux convolutifs d'images radiographiques thoraciques. Notre travail s'appuie sur un apprentissage profond pour classer les images médicales en 3 classes (COVID-19, Pneumonie virale, cas normaux).

Afin d'atteindre cet objectif, nous avons créé plusieurs modèles en utilisant l'apprentissage par transfert sur des modèles pré-entraînés et également en concaténant entre eux, puis en expérimentant l'apprentissage par transfert avec une grande base de données et en expérimentant la technique de Vision Transformer. Notre objectif était de proposer une nouvelle méthode pour le diagnostic de COVID, afin d'obtenir des bons résultats.

Nos modèles ont obtenu des résultats de précision significatifs et proches les uns des autres. Le modèle de la concaténation entre ResNet50 et Xception basé sur le transfert d'apprentissage est considéré comme le modèle le plus précis avec 94.43%, suivi par le reste des modèles avec des pourcentages proches. Et dans le modèle Big Transfer nous avons constaté une amélioration des résultats avec l'augmentation de la taille de la base de pré-entraînement, quant à le troisième modèle, il a donné des résultats encourageant. Cela confirme l'efficacité de notre solution.

**perspectives** Le diagnostic automatique est considéré comme un grand défi pour les chercheurs afin d'obtenir des résultats très précis et sans erreur, car on sait qu'il n'y a pas de travail complet.

Nous n'avons travaillé cela que dans sa version initiale, et il est toujours ouvert pour des améliorations et des travaux de comparaison avec d'autres méthodes de classification et essais expérimentaux à l'aide d'images de tomodensitométrie. Et nous allons essayer de combiner le Vision Transformer avec notre meilleur modèle (Xception + ResNet50V2). À l'avenir, nous travaillerons à l'améliorer en augmentant la taille de la base de données et en ajoutant davantage de sous-catégories de types de maladies COVID-19. Et en expérimentant plus de modèles en utilisant des techniques jamais essayées auparavant dans le domaine médical. Où nous espérons déployer nos modèles et les rendre applicables dans le monde réel dans l'espoir de les rendre plus faciles pour les médecins.

# Bibliographie

- [1] <https://www.worldometers.info/coronavirus/> , consulté le 12/02/2022.
- [2] Shi, Z. (2021). Du SRAS et du MERS à la COVID-19 : un voyage pour comprendre les coronavirus des chauves-souris. *Bulletin de l'Académie Nationale de Médecine*, 205(7), 732-736.
- [3] Dr. Larinouna, (janvier 2021) Livre 6eme année médecine module épidémiologie, université youcef benkhedda - alger 1, consulté le 11/02/2022.
- [4] <https://www.academie-sciences.fr/fr/Rapports-ouvrages-avis-et-recommandations-de-l-Academie/covid-19-maladie-virus-traitement.html> , consulté le 02/02/2022.
- [5] Adnet, J., & Dina, J. (2021). Virus grippaux et Sars-CoV-2, sommes-nous prêts pour le futur ? *Influenza viruses and SARS-CoV-2, are we ready for the future ?*. *Actualites Pharmaceutiques*, 60(608), 28-35.
- [6] Rozado, J., Ayesta, A., Morís, C., & Avanzas, P. (2020). Fisiopatología de la enfermedad cardiovascular en pacientes con COVID-19. Isquemia, trombosis y disfunción cardiaca. *Revista Española de Cardiología Suplementos*, 20, 2-8.
- [7] M. Lotfi, M.R. Hamblin, N. Rezaei, COVID-19 : Transmission, prevention, and potential therapeutic opportunities. *Clin Chim Acta*. 2020 Sep ;508 :254-266. doi : 10.1016/j.cca.2020.05.044. Epub 2020 May 29. PMID : 32474009 ; PMCID : PMC7256510.
- [8] [https://www.who.int/fr/health-topics/coronavirus#tab=tab\\_3](https://www.who.int/fr/health-topics/coronavirus#tab=tab_3) ,consulté le 03/02/2022.

- [9] B. Jones, N. Dean, R. Wunderink, M. Sockrider , What Is Pneumonia? , American Thoracic Society, <https://www.thoracic.org/patients/patient-resources/resources/what-is-pneumonia.pdf> , consulté le 12/02/2022.
- [10] Jain V, Vashisht R, Yilmaz G, et al. (2022 Jan). Pneumonia Pathology. Treasure Island (FL) : StatPearls Publishing; Available from : <https://www.ncbi.nlm.nih.gov/books/NBK526116/>.
- [11] Saraya T.(2017 Apr 17) Mycoplasma pneumoniae infection : Basics. J Gen Fam Med.;18(3) :118-125. doi : 10.1002/jgf2.15. PMID : 29264006; PMCID : PMC5689399.
- [12] Chinese Center for Disease Control and Prevention, Partie du diagnostic et traitement - COVID-19 prévention <https://www.chinacdc.cn/en/COVID19/202002/P020200219281884905880.pdf> , consulté le 03/02/2022.
- [13] <https://www.cerballiance.fr/fr/services/tous-les-tests-covid-19> , consulté le 03/02/2022.
- [14] François Chung;(2012). L'imagerie médicale : Un domaine d'ingénierie et de recherche au service de la société; L'Ing., 17, pp. 10-12,
- [15] Treyvaud, Geneviève. Francus, Pierre. Des Roches, Mathieu. (2020) La tomodensitométrie en archéologie. Les Nouvelles de l'archéologie. DOI : 10.4000/nda.9542.
- [16] <https://www.ottawaheart.ca/fr/examen-intervention/radiographie-thoracique> , consulté le 03/02/2022.
- [17] Ou X, Chen X, Xu X, Xie L, Chen X, Hong Z, Bai H, Liu X, Chen Q, Li L, Yang H. (2021 Dec 26). Recent Development in X-Ray Imaging Technology : Future and Challenges. Research (Wash D C);2021 :9892152. doi : 10.34133/2021/9892152. PMID : 35028585; PMCID : PMC8724686.
- [18] <https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images> , consulté le 12/02/2022.
- [19] Gutiérrez Y, Ott D, Töpperwien M, Salditt T, Scherber C. (2018 Jul 6). X-ray computed tomography and its potential in ecological research : A review of studies and optimization of specimen preparation. Ecol Evol;8(15) :7717-7732. doi : 10.1002/ece3.4149. PMID : 30151184; PMCID : PMC6106166.

- 
- [20] Thomas C. Kwee and Robert M. Kwee, (2020). Chest CT in COVID-19 : What the Radiologist Needs to Know, <https://pubs.rsna.org/doi/full/10.1148/rg.2020200159>.
- [21] J. Alzubi, A. Nayyar and A. Kumar, (2018). Machine Learning from Theory to Algorithms : An Overview *iopscience*, <https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/meta>.
- [22] M. I. Jordan and T. M. Mitchell, (2015). Machine learning : Trends, perspectives, and prospects *science DOI*, <https://pubmed.ncbi.nlm.nih.gov/26185243/>.
- [23] Tian, X., Héroult, R., Gasso, G., Canu, S., & EA4108, L. I. T. I. S. (2010, January). Pré-apprentissage supervisé pour les réseaux profonds. In *Proceedings of Rfia*.
- [24] <https://www.javatpoint.com/regression-vs-classification-in-machine-learning> , consulté le 15/02/2022.
- [25] O. Chapelle, B. Schölkopf, A. Zien, (2006). *Apprentissage semi-supervisé*, Cambridge, Massachusetts : MIT Press. ISBN 978-0-262-03358-9.
- [26] R. Thourani, Types of Semi Supervised Machine Learning Algorithms, <https://www.finsliqblog.com/ai-and-machine-learning/types-of-semi-supervised-algorithms/> , consulté le 15/02/2022.
- [27] Zimmer, M. (2018). *Apprentissage par renforcement développemental* (Doctoral dissertation, Université de Lorraine).
- [28] D.S. Estay, *reinforcement-learning use in supplychains*, <https://medium.com/analytics-vidhya/reinforcement-learning-use-in-supply-chains-d6d7a337e926> , consulté le 15/02/2022.
- [29] LeCun, Yann , Bengio, Y. , Hinton, Geoffrey. (2015). Deep Learning. *Nature*. 521. 436-44. [10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [30] H.Wang, B. Raj, E. P. Xing, (2017). On the origin of deep learning, <https://arxiv.org/abs/1702.07800>.
- [31] L. Alzubaidi, J. Zhang, A.J. Humaidi et al. (2021). Review of deep learning : concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8, 53 <https://doi.org/10.1186/s40537-021-00444-8>.
- [32] McGreivy, Nick. Hakim, Ammar. (2022). Convolutional Layers Are Not Translation Equivariant. [10.48550/arXiv.2206.04979](https://arxiv.org/abs/2206.04979) , consulté le 21/02/2022.

- 
- [33] E. Djenaihi, deep learning, <https://slideplayer.fr/slide/13975436/> , consulté le 21/02/2022.
- [34] V. Kumar, (2020). convolutional neural networks, <https://towardsdatascience.com/convolutional-neural-networks-f62dd896a856> , consulté le 23/02/2022.
- [35] Liu, J. (2021). Improved Padding in CNNs for Quantitative Susceptibility Mapping. arXiv preprint arXiv :2106.15331.
- [36] A. Perera,(2018). what is padding in convolutional neural network'sCNN's padding, <https://ayeshmanthaperera.medium.com/what-is-padding-in-cnns-71b21fb0dd7> , consulté le 23/02/2022.
- [37] Dubey, Shiv Ram. Chakraborty, Soumendu. (2021). Average Biased ReLU Based CNN Descriptor for Improved Face Retrieval. Multimedia Tools and Applications. 80. 10.1007/s11042-020-10269-x.
- [38] A. Amidi, S. Amidi, Deep Learning, <https://stanford.edu/shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels> , consulté le 21/02/2022.
- [39] N. McCullum, <https://nickmccullum.com/python-deep-learning/flattening-full-connection/> , consulté le 23/02/2022.
- [40] Buffoni, Lorenzo. Civitelli, Enrico. Giambagli, Lorenzo. Chicchi, Lorenzo. Fanelli, Duccio. (2022). Spectral pruning of fully connected layers. Scientific Reports. 12. 11201. 10.1038/s41598-022-14805-7. , consulté le 23/02/2022.
- [41] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
- [42] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, (2016). Densely Connected Convolutional Networks, <https://arxiv.org/abs/1608.06993v5>.
- [43] K. He, X. Zhang, S. Ren, et al. (2016). Deep residual learning for image recognition. In : Proceedings of the IEEE conference on computer vision and pattern recognition. p. 770-778.

- 
- [44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, (2014). Going Deeper with Convolutions, <https://arxiv.org/abs/1409.4842v1>.
- [45] K. Simonyan, A. Zisserman ,(2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, <https://doi.org/10.48550/arXiv.1409.1556>.
- [46] K. Srinivasan, L. Garg, D. Datta, A. A. Alaboudi, N. Z. Jhanjhi et al, (2021). "Performance comparison of deep cnn models for detecting driver's distraction," *Computers, Materials & Continua*, vol. 68, no.3, pp. 4109–4124.
- [47] Leo Souquet. (2019). Design of optimization algorithms for large scale continuous problems : application on deep learning. *Machine Learning [cs.LG]*. Université Paris-Est, (NNT : 2019PESC0089).
- [48] R. Aouassa, (2020). CNN Based Deep Face Recognition, <https://dspace.univ-guelma.dz/xmlui/handle/123456789/10136> , consulté le 28/02/2022.
- [49] H. Ampadou ,(2021). Understanding Dropouts in Deep Learning to reduce overfitting, <https://ai-pool.com/a/s/dropout-in-deep-learning> , consulté le 04/04/2022.
- [50] Q. Yang, Y. Zhang, W. Dai, S. Pan, (2020). *Transfer Learning*. Cambridge : Cambridge University Press. doi :10.1017/97811390617730
- [51] T. Rahman, M. Chowdhury, A. Khandakar, COVID-19 Radiography Database <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> , consulté le 02/03/2022.
- [52] Bachir, COVID-19 chest xray, <https://www.kaggle.com/bachrr/covid-chest-xray?select=images> , consulté le 02/03/2022.
- [53] P. Raikote, covid19 image dataset, <https://www.kaggle.com/pranavraikokte/covid19-image-dataset> , consulté le 02/03/2022
- [54] A. Zhao and 5 collaborators, COVIDx CXR-2, [https://www.kaggle.com/andyczhao/covidx-cxr2?select=competition\\_test](https://www.kaggle.com/andyczhao/covidx-cxr2?select=competition_test) , consulté le 02/03/2022.
- [55] N. Elsajid, <https://www.kaggle.com/datasets/nabeelsajid917/covid-19-x-ray-10000-images> , consulté le 02/03/2022.

- 
- [56] P. Patel, chest xray (covid19 & pneumonia), <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>, consulté le 02/03/2022.
- [57] A. Rahman, covid19 chest xray image dataset, <https://www.kaggle.com/alifrahman/covid19-chest-xray-image-dataset>, consulté le 02/03/2022
- [58] F. Fenta, chest xray for covid-19 detection, <https://www.kaggle.com/fusicfenta/chest-xray-for-covid19-detection>, consulté le 02/03/2022.
- [59] Praveen, coronahack –chest-xray-dataset, <https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset>, consulté le 02/03/2022.
- [60] Larxel, covid-19 X rays, <https://www.kaggle.com/andrewmvd/convid19-x-rays?select=X+rays>, consulté le 02/03/2022
- [61] I.D. Apostolopoulos, T.A. Mpesiana, (2020). Covid-19 : automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 43, 635–640 <https://doi.org/10.1007/s13246-020-00865-4>.
- [62] A. Narin, C. Kaya, Z. Pamuk, (2020). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, <https://arxiv.org/abs/2003.10849>.
- [63] H.S. Maghdid, A.T. Asaad, K.Z. Ghafoor, A.S. Sadiq, M.K. Khan, (2020). Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms, <https://arxiv.org/abs/2004.00038>.
- [64] S. Pathari, (2021). Automatic Detection of COVID-19 and Pneumonia from Chest X-Ray using Deep Learning, <https://arxiv.org/abs/2110.09384>.
- [65] A. Saeedi, M. Saeedi, A. Maghsoudi, (2020). A novel and reliable deep learning web-based tool to detect COVID-19 infection form chest CT-scan, <https://arxiv.org/ftp/arxiv/papers/2006/2006.14419.pdf>.
- [66] M. Heidari, S. Mirniaharikandehi, A.Z. Khuzani, G. Danala, Y. Qiu, B. Zheng, (2020). Improving performance of CNN to predict likelihood of COVID-19 using chest X-ray images with preprocessing algorithms, <https://arxiv.org/abs/2006.12229>.

- [67] M. Taban, R. Rasber, A. Dashti, A. Aras, (2020). Issues associated with deploying CNN transfer learning to detect COVID-19 from chest X-rays, <https://pubmed.ncbi.nlm.nih.gov/33025386/>.
- [68] L. Wang, A. Wong, (2020). COVID-Net : A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-ray Images, <https://arxiv.org/abs/2003.09871>.
- [69] E.E. Hemdan, M.A. Shouman, M.E. Karar, (2020). Covidx-net : a framework of deep learning classifiers to diagnose covid-19 in X-ray images, <https://arxiv.org/abs/2003.11055>.
- [70] K. Koonanit, S. Thongvigitmanee , N. Pongnapang, P. Thajchayapong, (2017). Image enhancement on digital x-ray images using N-CLAHE, 1-4. 10.1109/BMEi-CON.2017.8229130.
- [71] J. Liao, Y. Wang, D. Zhu, Y. Zou, S. Zhang, H. Zhou, (2020). Automatic Segmentation of Crop/Background Based on Luminance Partition Correction and Adaptive Threshold. *IEEE Access*. 8. 202611-202622. 10.1109/ACCESS.2020.3036278.
- [72] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-cam : Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
- [73] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F. Li, (2009). ImageNet : a Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248-255. 10.1109/CVPR.2009.5206848.
- [74] Rusiecki, A. (2019). Trimmed categorical cross-entropy for deep learning with label noise. *Electronics Letters*, 55(6), 319-320.
- [75] Kingma, D. P., & Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.
- [76] J. Duda, (2019). SGD momentum optimizer with step estimation by online parabola model, *arXiv preprint arXiv :1907.07063v3*.
- [77] A. Kolesnikov, L. Beyer , X. Zhai , J. Puigcerver, J. Yung,S. Gelly, N. Houlsby, (2020). Big Transfer(BiT) :General Visual Representation Learning, <https://arxiv.org/abs/1912.11370>.

- [78] K. Han et al., (2020). "A Survey on Vision Transformer," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi : 10.1109/TPAMI.2022.3152247.
- [79] F. Arifin, H. Artanto, Nurhasanah, T. S. Gunawan , (2021). Fast COVID-19 Detection of Chest X-Ray Images Using Single Shot Detection MobileNet Convolutional Neural Networks, <https://doi.org/10.35741/issn.0258-2724.56.2.19>.