

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DE 8 MAI 1945 – GUELMA -
FACULTÉ DES MATHÉMATIQUES, D'INFORMATIQUE ET DES SCIENCES DE LA
MATIÈRE

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème _____

**La segmentation d'organes à partir d'images
médicales**

Encadré Par :

DR. FAROU Brahim

Présenté par :

BOULEFRAKH Ines

Juin 2022

Remerciements

Alhamdoulillah qui m'a facilité mon périple, et qui m'a fait sortir des ténèbres de l'ignorance à la lumière de la science et de la connaissance, merci mon dieu de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel, Tout le mérite revient à Dieu avant tout.

Le travail présenté dans cette mémoire a été réalisé au département d'informatique de la faculté MISM de l'Université de 08 mai 1945 Guelma, sous la direction Dr. FAROU BRAHIM, je voudrais saisir cette occasion et exprimer ma plus grande gratitude pour mon encadreur pour sa disponibilité et la confiance qu'il m'a accordée.

Je tiens également à remercier les membres du jury de m'avoir fait l'honneur de juger mon travail.

Je remercie mes très chers parents pour l'encouragement et toute l'aide qu'ils m'ont apportée durant mes études et pour les sacrifices qu'ils ont consentis pour mon instruction et mon bien-être, Je tiens à remercier également mes deux petits frères ABDERRAHMAN et GHILES .

Remerciement spécial à mon ami KH.BILAL pour son encouragement et son soutien.

Un grand merci pour IMANE BOUACIDA pour son aide, ses inestimables conseils, ses explications claires et ses orientations précises.

Enfin, je remercie mes chères amis BADJI AHMED AMINE, NOUAR IKRAM et HAMICI LOUBNA pour l'amitié et le soutien qu'ils m'ont apportés durant la période de nos études et de préparation de ce travail.

RÉSUMÉ

La propagation de la maladie infectieuse "Covid-19" a bouleversé la santé et l'économie mondiale. Il existe plusieurs méthodes pour la détection et le diagnostic de cette maladie; cependant, ces derniers nécessitent du temps, non disponible dans toutes les structures et coûte cher pour le citoyen de classe moyenne. Pour cela, les chercheurs essayent de trouver une bonne alternative à ces derniers.

Dans ce contexte, la tomodensitométrie a permis d'ouvrir une voie à la détection de COVID-19 par traitement des images médicales numériques en utilisant les techniques de l'intelligence artificielle.

Dans ce projet nous avons adopté les techniques de l'apprentissage profond et de l'apprentissage autosupervisé dans le but de construire un système de détection de la maladie Covid-19 dans les images CT d'un patient suspect et de localiser les régions infectées par le virus si ce dernier est malade. Ce système nous permettra de faciliter la tâche de détection et de dépistage de cette maladie et de les mettre à la disposition des médecins ou des radiologues sans faire aucun effort ni perdre de temps.

Mots-clés : Images CT, Tomodensitométrie, Covid-19, Apprentissage en profondeur, Apprentissage auto-supervisé, Segmentation, Classification.

ABSTRACT

The spread of the infectious disease "Covid-19" has disrupted health and the global economy. There are several methods for the detection and diagnosis of this disease; however, these are time-consuming, not available in all facilities, and expensive for the middle-class citizen. For this, researchers are trying to find a good alternative to these methods.

In this context, computed tomography has paved the way for the detection of COVID-19 by processing digital medical images using artificial intelligence techniques.

In this project we have adopted the techniques of deep learning and self-supervised learning with the aim of building a system for detecting Covid-19 disease in CT images of a suspected patient and locating the regions infected with the virus if the latter and sick. This system will allow us to facilitate the task of detecting and screening for this disease and make them available to doctors or radiologists without making any effort or wasting time.

Keywords : CT images, Computed tomography, Covid-19, Deep learning, Self-supervised learning, Segmentation, Classification.

ملخص

انتشار مرض كورونا المعدي ادى الى تدهور الوضع الصحي و الاقصادي العالميين, للكشف عن هذا المرض وتشخيصه يعتمد الاطباء على تقنيات التصوير الطبي وخاصة التصوير المقطعي المدعم بالحاسوب اكثر من التقنيات البيولوجية وهذا بسبب مشاكلها المتعلقة بالتوافر, الوقت و التكلفة... الخ

التصوير المقطعي المدعم بالحاسوب مهد الطريق لمعالجة الصور الطبية الرقمية باستخدام تقنيات الذكاء الاصطناعي مثل التعلم العميق في هذا المشروع اعتمدنا على التعلم العميق و التعلم الذاتي بهدف انشاء نظام بكشف عن مرض كورونا في الصور المقطعي المدعمة بالحاسوب لمريض مشكوك فيه وتحديد المناطق المصابة بالفيروس اذا كان هذا الاخير مصابا. هذا النظام يمكن من تسهيل عملية الكشف وفحص هذا المرض وجعلها متاحة للاطباء و اخصائيي الاشعة دون اي جهد او مضیعة للوقت

الكلمات المفتاحية : صور التصوير المقطعي المحوسب ، كوفيد-19 ، التعلم العميق ، التعلم تحت الإشراف الذاتي ، التقسيم ، التصنيف

TABLE DES MATIÈRES

Liste des figures	x
Liste des tableaux	xi
Introduction générale	1
1 Généralité sur L'IA et l'apprentissage Automatique	3
1.1 Introduction	3
1.2 L'Intelligence Artificielle « IA »	4
1.3 Apprentissage automatique (Machine Learning)	5
1.4 Apprentissage profond (Deep Learning)(DL)	5
1.4.1 Pourquoi DL plutôt que ML	6
1.4.2 Les modèles basiques de DL les plus appliqués en sciences médicales	6
1.5 Les types d'apprentissage	9
1.5.1 Apprentissage supervisé	9
1.5.2 Apprentissage non supervisé	10
1.5.3 Apprentissage semi-supervisé	11
1.5.3.1 L'apprentissage inductif	11
1.5.3.2 L'apprentissage transductif	12

1.5.4	L'apprentissage auto-Supervisé (Self Supervised Learning SSL)	12
1.6	Les neurones	13
1.6.1	Les neurones biologiques	13
1.6.2	Les réseaux de neurones artificiels (ANN)	13
1.6.3	Les types de réseaux de neurones artificiels	14
1.7	La classification	15
1.8	La segmentation d'images	16
1.9	Conclusion	17
2	Détection de COVID-19 par image CT et segmentation des régions infectées	18
2.1	Introduction	18
2.2	Covid-19	18
2.2.1	Les symptômes de la maladie Covid-19	19
2.3	L'imagerie par la tomodensitométrie TDM (CT scan)	20
2.4	Diagnostic de Covid-19 par tomodensitométrie	21
2.5	Les travaux connexes	22
2.6	Architecture du système proposé	24
2.6.1	Description du modèle utilisé pour la classification	26
2.6.1.1	ResNet	26
2.6.1.2	ResNet50	27
2.6.1.3	Les paramètres utilisé pour le modèle	28
2.6.2	Description du modèle utilisé pour la segmentation	29
2.6.2.1	U-Net	29
2.6.2.2	Les paramètres utilisés pour le modèle	32
2.7	Conclusion	33
3	Implémentation, tests et résultats	34
3.1	Introduction	34
3.2	Environnement de développement	34
3.2.1	Google Colab	34

3.2.2	Pourquoi les GPU?	35
3.2.3	Python	35
3.3	Description des bibliothèques utilisées	36
3.3.1	Pandas	36
3.3.2	NumPy	37
3.3.3	Matplotlib	37
3.3.4	OpenCV	37
3.3.5	TensorFlow	37
3.3.6	Keras	38
3.4	Les bases de données utilisées	38
3.4.1	Dataset	38
3.4.2	Dataset++	39
3.4.3	CT Data Segmentation	40
3.5	Résultats du modèle de classification	41
3.5.1	Entraînement	41
3.5.2	Classification	43
3.6	Résultat du modèle de segmentation	45
3.6.1	Préparation des données	45
3.6.2	Entraînement	46
3.6.3	Évaluation du modèle	46
3.6.4	Prédiction	46
3.7	Conclusion	51
	Conclusion générale	52
	Bibliographie	53

TABLE DES FIGURES

1.1	sous-domaines de l'IA inclus les uns dans les autres[3]	5
1.2	schéma d'apprentissage supervisé	10
1.3	schéma d'apprentissage non supervisé	11
1.4	Neurones biologiques[21]	13
1.5	Schéma des réseaux de neurones artificiels[23]	15
1.6	Exemple de segmentation d'une image médicale[27]	16
1.7	Exemple de régions infectées par COVID-19 (B) en coupe axiale CT (A)[28]	17
2.1	Appareil scanner CT multi-détecteur sur fond de radiographie thora- cique pour le diagnostic Covid-19[35]	20
2.2	Les étapes de développement du système de classification avec DL . .	25
2.3	Les étapes de développement du système de classification avec SSL . .	25
2.4	Les étapes de développement du système de segmentation	26
2.5	l'architecture d'un bloc résiduel[55]	27
2.6	Le modèle ResNet50	29
2.7	Architecture U-Net (exemple pour 32x32 pixels dans la résolution la plus basse)[57]	31
2.8	La fonction IoU avec la compilation du modèle	32

2.9	Les paramètres d'entraînement du modèle	33
3.1	Le logo de Python[70]	36
3.2	(A) : Les bibliothèques utilisées pour la classification, (B) : Les bibliothèques utilisées pour la segmentation	36
3.3	Architecture de la base de données "Dataset"	38
3.4	La visualisation des images de "Dataset" dans Colab	39
3.5	Architecture de la base de données "Dataset++"	39
3.6	Architectutre de la base de données "CT Data Segmentation"	40
3.7	Évaluation du modèle de classsification DL	41
3.8	Évaluation du modèle	41
3.9	L'apprentissage auto-Supervisé du modèle avec une seule époque . . .	42
3.10	Résultat de l'apprentissage auto-Supervisé	42
3.11	Affichage des résultats de prédiction avec le modèle auto-suprvisé et les valeurs originaux	43
3.12	Résultats de test de la classification du système basé sur DL	44
3.13	Résultats de test de la classification du système basé l'apprentissage auto supervisé	44
3.14	Division d'images pour l'entraînement et l'évaluation du modèle . . .	45
3.15	Nombres des images dans chaque variable	45
3.16	Résultat avec 89 époques	46
3.17	Évaluation du modèle	46
3.18	Fonction de prédiction du modèle de segmentation	47
3.19	L'image CT	47
3.20	(A) : Mask originale de l'image, (B) : Mask donné par le système . . .	47
3.21	(A) :Image 1 , (B) : Mask par prédiction 1	48
3.22	(A) :Image 2, (B) : Mask par prédiction 2	48
3.23	(A) :Image 3, (B) : Mask par prédiction 3	49
3.24	(A) :Image 4, (B) : Mask par prédiction 4	49

3.25 (A) :Image 5, (B) : Mask par prédiction 5	50
3.26 (A) :Image 6, (B) : Mask par prédiction 6	51

LISTE DES TABLEAUX

1.1	DL les plus appliqués en sciences médicales[3][6]	9
2.1	Quelques travaux effectués avec des images CT	22
3.1	La distribution des images sur les trois parties de la base	39

INTRODUCTION GÉNÉRALE

COVID-19 est une maladie provoquée par un virus de la famille des « Coronaviridae » signalé par l'OMS comme une pandémie et un problème d'urgence mondiale à la fin de 2019[1]

Il existe plusieurs outils pour détecter ce virus dans le corps humain, l'un des outils les plus faciles pour la détection de cette maladie est la tomographie par densitométrie (TDM).

Les images par TDM fournissent des informations précieuses sur la façon dont cette nouvelle maladie affecte les poumons des patients, donc elles sont utilisées efficacement pour compléter d'autres méthodes de diagnostic à cause des problèmes de coût, de temps et de disponibilité.

La TDM a permis d'ouvrir la voie au traitement des images médicales numériques avec les techniques de l'intelligence artificielle et de l'apprentissage profond avec une meilleure résolution.

L'objectif principal de ce projet est de construire un système qui permet de détecter COVID-19 dans les images CT d'un patient suspect et de localiser les régions infectées par le virus si ce dernier est malade en se basant sur des techniques de l'apprentissage profond et de l'apprentissage auto-supervisé afin d'aider les radiologues à prendre des décisions, et aussi de tester la performance de l'apprentissage auto-supervisé qui peut être considérée comme une solution efficace pour la rareté des données annotées.

Tous ces éléments sont traités dans ce projet à travers trois chapitres principaux :

- **Chapitre 1 : Généralité sur l'IA et l'apprentissage automatique.** Dans ce chapitre nous présentons les principes de base de l'intelligence artificielle, et quelques types d'apprentissage.
- **Chapitre 2 : Détection de COVID-19 par image CT et segmentation des régions infectées.** Dans ce chapitre, nous apprenons plus sur la maladie Covid-19, ainsi que l'imagerie par la tomographie TDM (CT scan) et son rôle dans le diagnostic de cette maladie. Nous présentons ensuite la conception détaillée de notre système de détection et de classification.
- **Chapitre 3 : Implémentation, tests et résultats .** Dans ce chapitre, nous abordons les aspects techniques relatifs à l'implémentation, les tests et les résultats de ce projet et de tous les moyens qui ont été adoptés pour la réalisation de notre système de détection Covid-19.

CHAPITRE 1

GÉNÉRALITÉ SUR L'IA ET L'APPRENTISSAGE AUTOMATIQUE

1.1 Introduction

Les années 1950 ont vu naître l'ambition de créer des systèmes ou des machines capables de simuler l'intelligence humaine. L'intelligence Artificielle (IA) vise à générer des systèmes informatiques qui ont la capacité de simplifier voire de remplacer l'intervention humaine. Ces derniers pouvant comprendre, percevoir et souvent prendre des décisions.

Depuis plus d'un demi-siècle, la recherche dans le domaine de l'IA s'est considérablement développée. Ce domaine de recherche a de plus en plus attiré l'attention de nombreux disciplines et chercheurs d'horizons différents et est devenu un sujet largement interdisciplinaire. Avec le développement technologique, l'intelligence artificielle s'est rapidement répandue et développée, et elle a favorisé le développement d'autres disciplines.

Ces dernières années, les ordinateurs modernes ont fait de grands progrès tant sur le plan matériel que logiciel, et la recherche sur l'IA a également joué un rôle important. Bien que l'intelligence artificielle soit confrontée à de nombreux défis dans

le processus de développement, les défis coexistent toujours avec des opportunités afin de développer des systèmes d'aide à la décision et d'améliorer la vie humaine. Ce chapitre présente une définition de l'intelligence artificielle ainsi qu'une description des techniques avancées d'apprentissage automatique, en mettant l'accent sur l'apprentissage profond « réseaux de neurones convolutionnels » et ses différentes couches.

1.2 L'Intelligence Artificielle « IA »

L'IA est la science et l'ingénierie de la création de machines intelligentes qui peuvent penser par elles-mêmes et prendre leurs propres décisions afin d'imiter le comportement (intelligence) humain[2] , Les modèles d'intelligence artificielle (IA) jouent un rôle croissant dans la recherche biomédicale et les services de santé. L'intelligence artificielle présente de nombreux avantages :

- Grâce à sa précision, le principal avantage de l'intégration de l'intelligence artificielle est de réduire le risque d'erreur (moins d'erreurs). De plus, l'IA peut initier une discussion collaborative avec les humains pour les aider à prendre des décisions (les décisions peuvent être prises très rapidement).
- L'intelligence artificielle fournira des pistes, des indicateurs, des données statistiques et des possibilités qui simplifieront le processus de prise de décision, ce qui réduira le risque d'erreur.
- L'intelligence artificielle fournit des outils infatigables. Contrairement aux humains, les machines ne nécessitent pas de pauses ou de rafraîchissements fréquents. Elles sont programmées pour de longues heures et peuvent fonctionner sans interruption.

L'intelligence artificielle est devenue un sujet populaire ces dernières années, en grande partie grâce aux progrès récents de l'apprentissage automatique et en particulier aux réalisations obtenues grâce à l'apprentissage profond « deep learning ».

L'IA contient deux sous-domaines majeurs, L'apprentissage machine (Machine Learning ML) et l'apprentissage profond (Deep Learning DL)

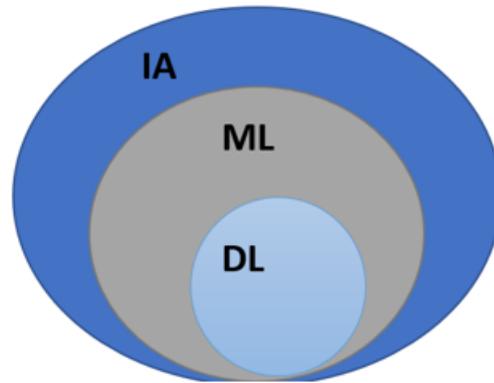


FIGURE 1.1 – sous-domaines de l'IA inclus les uns dans les autres[3]

1.3 Apprentissage automatique (Machine Learning)

L'apprentissage automatique est un sous-domaine majeur de l'intelligence artificielle qui vise à étudier, développer, comprendre et évaluer des algorithmes et des techniques qui permettent aux ordinateurs / machines d'apprendre[2]. Cet apprentissage, qui se fait toujours, est basé sur une sorte d'expérience illustrée par / reposant sur des données d'entrée ou des instructions connues. L'apprentissage automatique consiste à faire en sorte que les ordinateurs modifient ou adaptent leurs actions afin qu'elles deviennent plus précises[4].

1.4 Apprentissage profond (Deep Learning)(DL)

DL une fonction de l'IA qui imite le fonctionnement du cerveau humain[3]. C'est l'une des formes de l'apprentissage automatique (Machine Learning) dans laquelle on développe des réseaux neurones artificiels[3]. Dans nos jours, les méthodes basées DL ont données

un nouveau souffle aux domaines de recherche et elles sont utilisé partout, par exemple : Aider à détecter la fraude ou le blanchiment d'argent, le traitement des données pour la détection d'objets, la reconnaissance de la parole, la traduction des langues et la prise de décisions, etc.

Dans le domaine médical, la prédiction ou la détection précise de maladies ou d'anomalies est une tâche difficile, par conséquent, les techniques d'apprentissage profond jouent un rôle important dans ce domaine afin que les experts médicaux puissent prendre leur décision ou préparer un calendrier de traitement précis[3].

1.4.1 Pourquoi DL plutôt que ML

L'apprentissage profond est capable d'extraire, de traiter et d'apprendre des représentations de données dans leur forme originale(non modifiées). Ce même processus est très limité dans l'apprentissage machine et parfois impossible, car il nécessite de sélectionner soigneusement les caractéristiques des données avec une grande compétence en ingénierie pour les représentations[3].

L'avantage de DL sur ML est qu'il est capable de comprendre les caractéristiques des données brutes sans les extraire manuellement comme dans ML.

Cela permet d'économiser du temps et des efforts, en particulier pour les données très volumineuses[4]. En outre, les résultats obtenus par l'apprentissage profond sont meilleurs, plus précis et plus perfectionnés que ceux de l'apprentissage machine[5]

1.4.2 Les modèles basiques de DL les plus appliqués en sciences médicales

Les modèles de DL	Description
Réseau de neurones profond (Deep Neural Network)	<ul style="list-style-type: none">— Cadre generalisé du modèle d'apprentissage profond Largement applicable pour le modèle de classification et de régression— Composé de plus de 2 canapés cachés— Largement utilisé avec un taux de succès élevé dans différents domaines— Le processus d'apprentissage prend beaucoup de temps
Réseau de croyances profondes (Deep Belief Network)	<ul style="list-style-type: none">— Combinaison de machines à usage limité Boltzman (RBM)— La couche cachée de chaque réseau visible à la couche suivante— Il permet un apprentissage supervisé et non-supervisé— Utilisé une approche d'apprentissage gourmande pour initialiser le réseau— Maximiser les inférences de vraisemblance directement— le traitement peut être coûteux en raison de l'initialisation et du processus d'échantillonnage

La suite dans la page suivante...

<p>Réseau neuronal récurrent (Recurrent Neural Network)'RNN'</p>	<ul style="list-style-type: none">— Un réseau neuronal capable d'analyser le flux de données— Partager le même poids pour chaque étape— Très utile quand la production dépend des résultats précédentes— Peut mémoriser des événements séquentiels— Problèmes d'apprentissage dus au problème de gradient
<p>Réseau neuronal convolutif (Convolutional Neural Network)'CNN'</p>	<ul style="list-style-type: none">— Ce modèle est adapté aux données 2D -Conçu par la fonctionnalité du comportement des neurones biologiques— Chaque filtre de couche cachée transforme son entrée en volume de sortie 3D pour l'activation des neurones— Les CNNs surpassent tous les autres algorithmes ML classiques et faits un grand succès dans les tâches de traitement de vision par ordinateur (Computer vision Tasks)— Ils ont des larges applications dans le traitement d'images et vidéo et le traitement du langage naturel (NLP)— Il nécessite un grand ensemble de données étiquetées

La suite dans la page suivante...

Réseau antagoniste génératif(Generative Adversarial Network)'GAN'	<ul style="list-style-type: none"> — Les GANs d'origines ont été proposés en 2014. — C'est une technique émergente pour l'apprentissage semi-supervisé et non supervisé — Se compose de deux agents un générateur G et un discrimination D — Le GAN peut éviter certaines lacunes dans l'application pratique de certains modèles de génération traditionnels — Il peut optimiser subtilement certaines fonctions de perte qui sont à peine à traiter par l'apprentissage contradictoire — Il ne peut pas atteindre la performance idéale — La difficulté de contrôler la synchronisation des deux réseaux antagonistes
---	--

TABLE 1.1 – DL les plus appliqués en sciences médicales[3][6]

1.5 Les types d'apprentissage

Les approches d'apprentissage DEEP ont obtenu un succès remarquable dans le traitement des images médicales, exigeant généralement une grande quantité de données étiquetées pour l'apprentissage du modèle.

1.5.1 Apprentissage supervisé

C'est une technique d'apprentissage qui utilise des données étiquetées, les ensembles de données sont dits étiquetés lorsqu'ils comportent à la fois des variables d'entrée (X) et une variable de sortie (Y); les données ont donc déjà été associées à la

bonne réponse et nous utilisons un algorithme pour apprendre la fonction de correspondance entre l'entrée et la sortie[7]

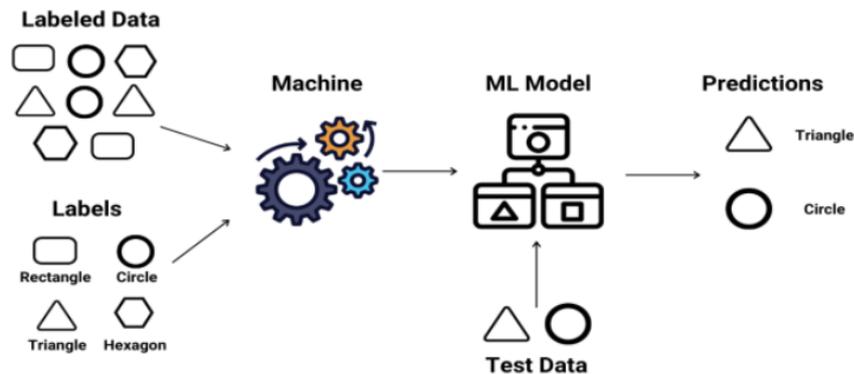


FIGURE 1.2 – schéma d'apprentissage supervisé

Les problèmes de ce type d'apprentissage peuvent être regroupés en problèmes de **classification** (lorsque le paramètre de sortie est une catégorie, comme rouge ou bleu) et de **régression** (lorsque le paramètre de sortie est une valeur réelle, comme les poids et les âges) et d'autres qui sont basés sur ces deux types de problèmes

1.5.2 Apprentissage non supervisé

Par opposition à l'apprentissage supervisé, c'est un apprentissage qui se fait sans la présence de données étiquetées donc nous ne fournissons pas de données d'apprentissage au réseau, mais simplement des données d'entrée (X) sans sortie associée, et le réseau doit conclure de lui-même en se basant sur la structure des données pour obtenir davantage d'informations. On dit apprentissage non supervisé, car il n'y a pas de données étiquetées à partir desquelles les modèles peuvent apprendre, il est donc un peu plus difficile à mettre en œuvre, car ils sont laissés seuls pour trouver des distributions intéressantes dans les données. Les problèmes de ce type peuvent être regroupés en deux catégories :

- **Le regroupement** : lorsque nous devons trouver les combinaisons essentielles dans les données, comme le regroupement des clients par comportement d'achat.

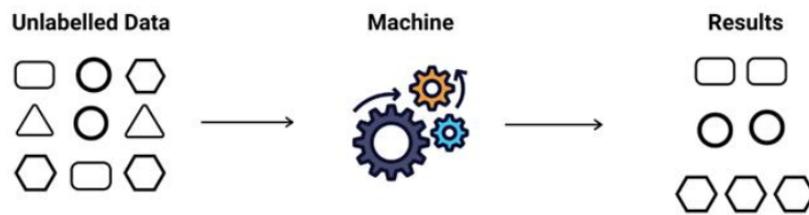


FIGURE 1.3 – schéma d'apprentissage non supervisé

- **L'association** : lorsque nous devons induire des règles décrivant de grandes parties de nos données, comme les personnes qui achètent X ont également tendance à acheter Y.

1.5.3 Apprentissage semi-supervisé

Dans le but de combiner la puissance de l'apprentissage supervisé et non supervisé, l'apprentissage semi-supervisé et l'apprentissage actif (AL) ont été conçus[8]

SSL fait référence aux algorithmes dans lesquels une combinaison de données étiquetées et non étiquetées sont utilisées pour l'apprentissage du modèle, pour faire face à la rareté des données étiquetées, où l'annotation est coûteuse et prend du temps. Dans la littérature, on distingue deux types légèrement différents de SSL, l'apprentissage inductif et l'apprentissage transductif [8]

1.5.3.1 L'apprentissage inductif

C'est l'approche d'apprentissage supervisé la plus courante. Elle consiste à construire un modèle d'apprentissage automatique et à l'entraîner sur un ensemble d'apprentissages étiqueté, puis à utiliser le modèle entraîné pour prédire les étiquettes d'un ensemble de tests jamais rencontrés auparavant.

1.5.3.2 L'apprentissage transductif

Dans ce type d'apprentissage, le modèle d'apprentissage automatique a observé au préalable les données de formation et de test. Le modèle apprend à partir de l'ensemble de formation précédemment observé, puis prédit les étiquettes de l'ensemble de tests. Même si les étiquettes des ensembles de tests ne sont pas disponibles, on peut bénéficier des modèles et de toute information supplémentaire présents dans le processus d'apprentissage.

1.5.4 L'apprentissage auto-Supervisé (Self Supervised Learning SSL)

C'est un apprentissage qui se fait sans la présence de grande quantité de données étiquetées avec annotations humaines, donc c'est une méthode d'apprentissage dans laquelle les modèles sont explicitement formés avec des pseudo-étiquettes générées automatiquement sans aucune annotation humaine [9]. Ce processus peut être considérée comme une solution efficace pour la rareté des données annotées et surtout les données médicales, donc elle est devenue un choix populaire dans l'analyse d'image médicale.

Plusieurs articles se réfèrent à ce type d'apprentissage comme un sous-ensemble d'apprentissage non-supervisé [10], [11], [12], [13], [14], [15], [16], [17], etc. tandis que d'autres, le considèrent comme un algorithme incrémental dont l'idée principale est de construire d'abord une fonction f sur des données étiquetées, la fonction f est ensuite utilisée pour prédire les étiquettes des données non étiquetées[18]

De nombreux ouvrages classent les méthodes d'apprentissage auto-supervisées en trois différentes approches Générative, Contrastive et Générative-Contrastive [9][19],

L'apprentissage auto-supervisé a gagné une popularité grâce à sa capacité à éviter le coût de l'annotation d'ensembles de données à grande échelle[20].

1.6 Les neurones

1.6.1 Les neurones biologiques

Le neurone où ce qu'on appelle la cellule nerveuse est nommé unité fonctionnelle de base biologique pour la distinguer des neurones artificiels en sciences informatiques. Les neurones donnent naissance à la pensée humaine, l'esprit, la conscience, l'imagination, l'opinion et en général l'intelligence [21]

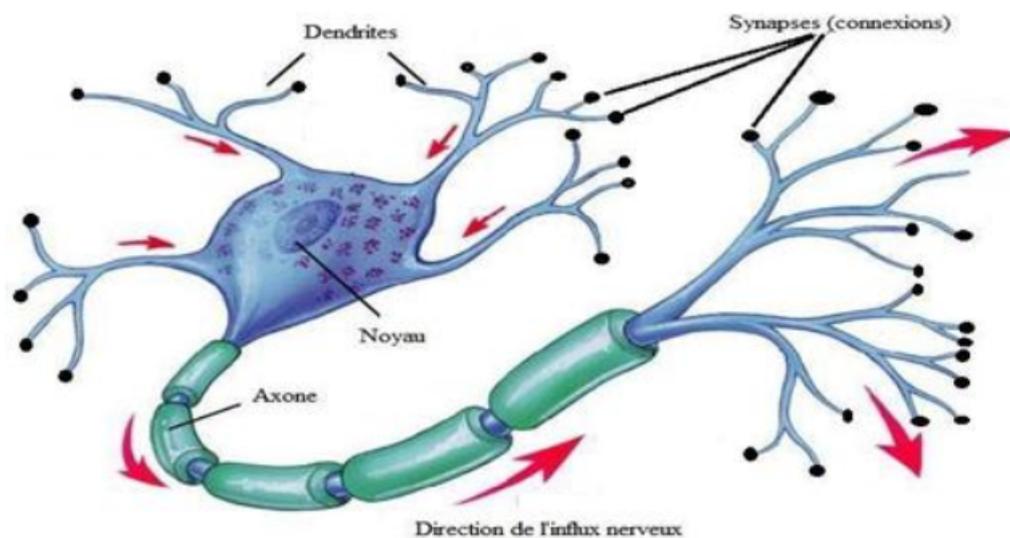


FIGURE 1.4 – Neurones biologiques[21]

1.6.2 Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels (ANN) ou systèmes connexionnistes sont des systèmes informatiques inspirés des réseaux de neurones biologiques

Un ANN est basée sur une collection d'unités connectées appelées neurones artificiels (analogues aux neurones biologiques dans un cerveau biologique) ou chaque connexion(synapse) entre neurones peut transmettre un signal à un autre neurone.

Le neurone récepteur (postsynaptique) peut traiter les signaux, puis signaler les neurones en aval qui lui sont connectés. Les neurones peuvent avoir un état, généralement représenté par des nombres réels compris entre 0 et 1. Les neurones et les synapses peuvent également avoir un poids qui varie au fur et à mesure de l'apprentissage, ce qui peut augmenter ou diminuer la force du signal qu'ils envoient en aval [22]

1.6.3 Les types de réseaux de neurones artificiels

Les types de réseau de neurones diffèrent par plusieurs paramètres (Figure 1.5) :

- La topologie des connexions entre les neurones,
- La fonction de seuillage utilisée (sigmoïde, échelon, fonction linéaire, fonction de Gauss),
- L'algorithme d'apprentissage (rétro propagation du gradient, cascade corrélative),
- La fonction d'agrégation utilisée (somme pondérée, distance pseudoeuclidienne, etc.)

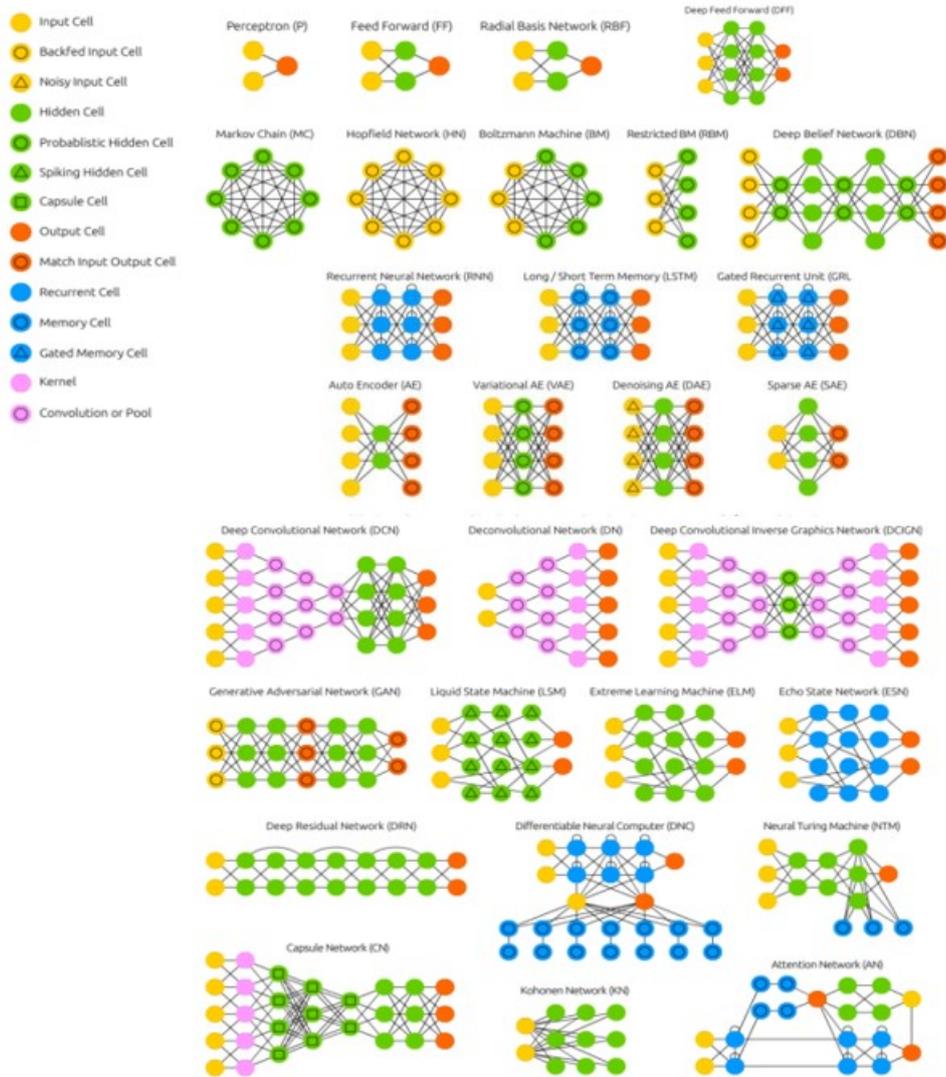


FIGURE 1.5 – Schéma des réseaux de neurones artificiels[23]

1.7 La classification

Le terme classification consiste à organiser des données en classes ou en catégories selon des critères convenus. Le regroupement des données en doit être le plus homogène possible.

Généralement, les méthodes de classification ont pour but de regrouper les éléments d'un ensemble " $X = X_1, X_2, X_3, \dots, X_n$ ", en un nombre N optimal de classes selon leurs ressemblances[24].

1.8 La segmentation d'images

La segmentation de l'image est un processus très essentiel dans le domaine de traitement de l'image et à la reconnaissance de formes. Elle consiste à partitionner ou diviser une image I suivant un caractère d'homogénéité

On appelle segmentation d'une image l'opération consistant à identifier les structures d'intérêt dans cette image, d'une autre façon nous pouvons dire que le but de toute méthode de segmentation est l'extraction des régions d'intérêt du fond ou des autres régions dans une image[25][26].

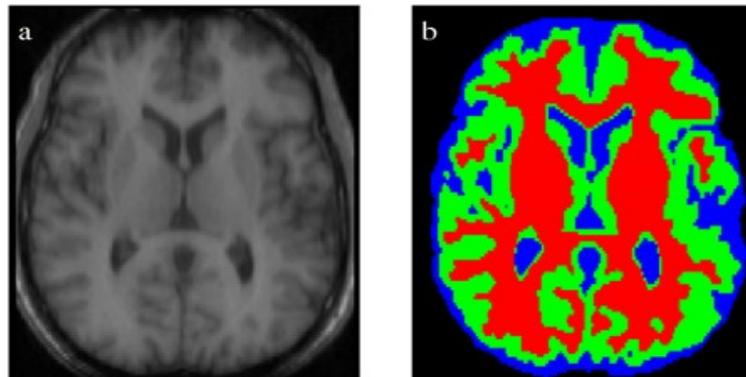


FIGURE 1.6 – Exemple de segmentation d'une image médicale[27]

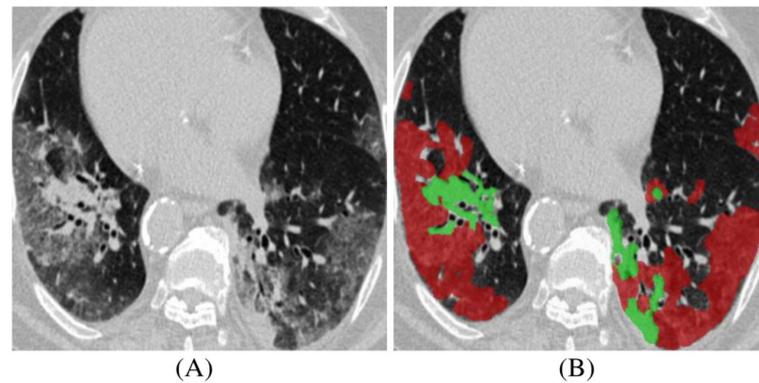


FIGURE 1.7 – Exemple de régions infectées par COVID-19 (B) en coupe axiale CT (A)[28]

1.9 Conclusion

Nous avons présenté dans ce chapitre des notions de base de l'intelligence artificielle, ensuite nous avons présenté quelques types d'apprentissage tels que l'apprentissage supervisé et non supervisé et surtout l'apprentissage auto-supervisé que nous utiliserons dans le développement de notre système.

CHAPITRE 2

DÉTECTION DE COVID-19 PAR IMAGE CT ET SEGMENTATION DES RÉGIONS INFECTÉES

2.1 Introduction

À cause du problème du coût, de temps et de disponibilité requis pour faire les tests conventionnels de RT-PCR (Reverse Transcription Polymerase Chain Reaction) pour détecter COVID-19, les chercheurs ont essayé d'exploiter d'autres méthodes qui soient moins coûteuses et offrant une qualité de résultats similaire à un test PCR. Les solutions basées sur les images médicales comme les radiographies et surtout la tomodensitométrie (TDM) ont donné des résultats prometteurs pour détecter cette maladie infectieuse qui a été signalée pour la première fois en décembre 2019.

2.2 Covid-19



Covid-19 fait référence à « Coronavirus Disease 2019 », la maladie provoquée par un virus de la famille des Coronaviridae le SARS-CoV-2. Cette maladie infectieuse est une zoonose, dont l'origine est encore débattue et qui a émergé en décembre

2019 dans la ville de Wuhan dans la province du Hubei en Chine. Elle s'est rapidement propagée d'abord dans toute la Chine, puis à l'étranger provoquant une épidémie mondiale[29][1].

Selon l'OMS, des millions de personnes sont infectées par ce virus dans le monde. En Algérie du 3 janvier 2020 jusqu'au 17 mars 2022, on a comptabilisé **265 496** cas confirmés de COVID-19 avec **6869** décès et **13 704 895** doses de vaccins administrés[30].

La plupart des personnes infectées par le virus présentent une maladie respiratoire d'intensité pouvant être mortelle chez les patients fragilisés par l'âge ou par autre maladie chronique. Elle se transmet par contact rapproché avec des personnes infectées. La maladie pourrait aussi être transmise par des patients asymptomatiques, mais les données scientifiques manquent pour en attester avec certitude [29][1].

Le test RT-PCR est parmi les méthodes les plus utilisées pour identifier distinctement le coronavirus. C'est une procédure d'assemblage d'échantillons à partir d'une région du corps d'une personne, où le coronavirus est le plus susceptible de se rassembler, comme le nez ou le throa d'une personne. Ensuite, cet échantillon passe par un processus appelé extraction, qui sépare les données génétiques de tout virus qui peut exister[31]. Cependant, la disponibilité de RT-PCR ou des autres outils de diagnostic biologique est limitée dans plusieurs pays du monde, en plus il a un taux élevé de faux négatif, où le modèle prédit le résultat comme négatif, mais en fait il est positif (faux-négatif).

Par conséquent, les images médicales telles que la tomodensitométrie (TDM) et les images radiographiques peuvent être les meilleures solutions alternatives pour détecter ce virus[31].

2.2.1 Les symptômes de la maladie Covid-19

Plusieurs symptômes ont été détectés dans les malades, les principaux sont[29] :

- La pneumonie (la complication la plus fréquente du Covid-19),

- Perte du goût ou de l'odorat,
- La fièvre, la fatigue et une toux sèche,
- La douleur,
- La dyspnée,
- Une congestion et un écoulement nasal,
- Des maux de gorge,
- Diarrhée, nausées

Ces symptômes sont généralement bénins, mais environ une personne sur six présente des symptômes plus sévères. Il existe aussi des cas asymptomatiques, c'est-à-dire que les patients n'ont aucun symptôme apparent malgré la détection du virus[32].

2.3 L'imagerie par la tomodensitométrie TDM (CT scan)

La tomodensitométrie (en anglais **Computed Tomography**) est une méthode diagnostique d'imagerie médicale qui fournit des images en coupes transversales du corps à l'aide de rayons X [33], elle a influencé de nombreux domaines depuis sa création au cours des années 1970 [34].

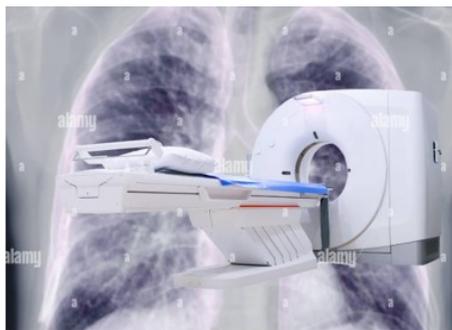


FIGURE 2.1 – Appareil scanner CT multi-détecteur sur fond de radiographie thoracique pour le diagnostic Covid-19[35]

Le CT ressemble à un gros cube (le portique) qui comporte une ouverture en son centre, le patient sera positionné sur une table mobile qui se déplace à travers cette

ouverture. À l'intérieur de la machine, un tube à rayons X et des détecteurs tournent, ce qui occasionne un bruit peu important durant l'examen[33].

L'appareil CT peut être utilisé pour le diagnostic de plusieurs maladies infectieuses et ostéo-articulaires comme[33] :

- La détection d'une hémorragie,
- La localisation des fractures du crâne ainsi que les lésions traumatiques du cerveau,
- L'étude des sinus,
- L'étude des tissus et des os de la face chez les patients présentant des traumatismes et chez qui une chirurgie de reconstruction est prévue

2.4 Diagnostique de Covid-19 par tomодensitométrie

La tomодensitométrie joue un rôle important dans le diagnostic des maladies pulmonaires. Au cours des deux dernières années, la détection automatisée des infections pulmonaires à partir d'images CT offre un grand potentiel pour renforcer la stratégie de soins de santé traditionnelle pour diagnostiquer Covid-19 [28].

Une étude clinique menée auprès de 1014 patients à Wuhan, en Chine, a montré que l'analyse TDM peut atteindre 0,97% de sensibilité, 0,25% de spécificité et 0,68% de précision pour la détection de COVID-19, avec les résultats de la RT-PCR à titre de référence[28].

Des observations similaires ont également été rapportées dans d'autres études[36],[37], suggérant que l'imagerie par tomодensitométrie pourrait être utile pour diagnostiquer Covid-19[28].

Travaux	Type d'images	La base de données utilisée
[38] [39]	CT	The Lung Image Database Consortium (LIDC). Lien : https://doi.org/10.1118/1.3528204
[40]	CT	China University Hospitals of Geneva (HUG). Lien : www.ChainZ.cn , El-Camino Hospital (CA), Zhejiang Province
[41]	CT	Societa Italiana di Radiologia Medica e Interventistica. Lien : https://www.sirm.org/
[42]	CT	Lung Segmentation and Candidate Points Generation. Lien : https://www.kaggle.com/arturscussel/lungsegmentation-and-candidate-points-generation
[43] [28]	CT	COVID-19 CT segmentation dataset. Lien : http://medicalsegmentation.com/COVID19/
[44]	CT	COVID-CT. Lien : https://github.com/UCSD-AI4H/COVID-CT
[45]	Xray et CT	BIMCV COVID-19+. Lien : https://bimcv.cipf.es/bimcv-projects/bimcvCOVID19/

TABLE 2.1 – Quelques travaux effectués avec des images CT

2.5 Les travaux connexes

Le besoin d'identification automatique des maladies thoraciques et pulmonaires par les techniques de la radiographie telles que X-ray ou tomographie (CT-scan) est devenu urgent et peut être réalisé par l'intelligence artificielle, surtout dans ces dernières deux années avec l'émergence de la pandémie de Coronavirus. De nombreux scientifiques ont utilisé des images radiologiques pour le développement des systèmes intelligents de classification et de segmentation visant à diagnostiquer cette maladie.

Pour la détection de COVID-19 et la segmentation de l'infection au niveau pulmonaire, plusieurs travaux d'apprentissage en profondeur sur les images radiographiques du thorax et les tomographies ont émergé et signalés dans les prochains articles :

Amyar Amine et al. [46] ont présenté un outil de classification automatique et de segmentation pour aider à dépister la pneumonie liée au COVID-19 avec les images CT. Ils proposent un nouveau modèle d'apprentissage approfondi multitâche pour identifier conjointement le patient atteint de Covid-19 et segmenter la lésion Covid-19 à partir des images CT.

Dans [47], Ali Narin et al. ont créé un réseau de neurones à convolution profonde pour détecter automatiquement le Covid-19 sur les images radiographiques X-ray. Ils ont utilisé une approche basée sur l'apprentissage par transfert avec des architectures très profondes telles que ResNet50, InceptionV3 et Inception-ResNetV2. Les algorithmes ont été formés sur la base de 100 images (50 COVID contre 50 non COVID) en cinq validations croisées. Les auteurs ont affirmé 97% d'exactitude en utilisant InceptionV3 et 87% en utilisant Inception-ResNetV2.

Le modèle présenté par Jinyu Zhao et al. [48] utilise un conteneur pour les CT-scans, initialement avec 275 CT Covid-19 images, sur lequel ils ont également appliqué un algorithme d'apprentissage par transfert à l'aide de chest-x-ray14[49] avec 169-layer DenseNet [50], la performance du modèle est de 84,7% avec une surface sous la courbe ROC de 82,4%.

Dans [51], Jin et al. ont créé et déployé un outil d'intelligence artificielle pour analyser les images scanographiques de Covid-19 en quatre semaines. Pour ce faire, une équipe multidisciplinaire de 30 personnes a collaboré à l'aide d'une base de données de 1136 images, dont 723 images positives de COVID-19 provenant de cinq hôpitaux pour obtenir une sensibilité de 0,974 % et une spécificité de 0,922 %. Le système a été déployé dans 16 hôpitaux et effectué plus de 1300 dépistages par jour. Ils ont proposé un modèle combiné pour la classification et la segmentation montrant les régions de lésion en plus des résultats du dépistage.

Les auteurs dans [52] ont utilisé plusieurs modèles dont 3D U-NET++, V-NET, FCN-8S pour la segmentation et InceptionV3, ResNet50 et autres pour la classification. Ils ont pu atteindre un coefficient de Dice de 0.754 en utilisant 3D U-NET ++

formé sur 732 cas, La combinaison de 3D U-NET ++ et ResNet50 a donné une zone sous la courbe OCR de 0,991 avec une sensibilité de 0,974 et une spécificité de 0,922. Il convient de noter ici que les deux modèles sont indépendants et qu'ils ne peuvent donc pas s'aider mutuellement pour améliorer à la fois les performances de classification et de segmentation.

Y. Pathak et al.[53] ont utilisé l'apprentissage par transfert pour classer les patients infectés par Covid-19. ils ont introduit une fonction de perte en douceur supérieure à 2 avec des attributs sensibles aux coûts pour gérer le bruit et un ensemble de données déséquilibré. Le modèle a été formé sur une base de données publique d'images CT et obtenu une précision de 0,93%, une sensibilité de 0,91% et une spécificité de 0,94%.

Dans les travaux présentés par S. Dilbag et al.[54] on trouve un réseau de neurones convolutionnels à évolution différentielle multi-objectif pour classer les patients atteints de Covid-19 à partir d'images CT.

2.6 Architecture du système proposé

Nous allons proposer un système qui permet de détecter COVID-19 dans les images CT d'un patient suspect et de localiser les régions infectées par le virus si ce dernier est malade. Nous allons présenter l'architecture du système et détailler les différents modules qui le composent. Les choix des algorithmes utilisés sont basés sur l'étude présentée précédemment afin de sélectionner les outils les plus appropriés et les plus compatibles avec le domaine de l'imagerie médicale, la classification et la segmentation des maladies pulmonaires.

Nous avons divisé notre travail en deux grandes étapes : classification et segmentation

Étape 1 :Elle est également divisée en deux phases, une classification (Covid ou Normal) des images CT dans l'environnement Colab avec le modèle "Resnet50". La première est une classification avec l'apprentissage en profondeur en utilisant les

images CT. Le chemin de la Figure 2.2 explique les étapes de développement de ce système.

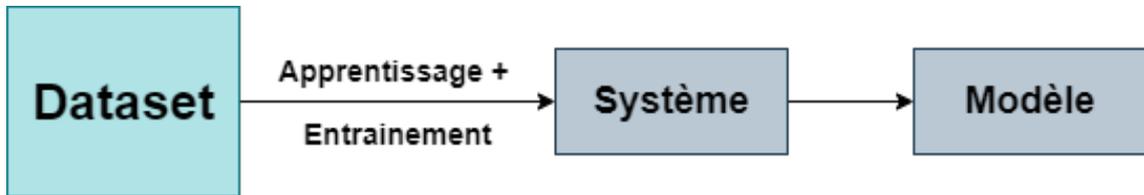


FIGURE 2.2 – Les étapes de développement du système de classification avec DL

Et la deuxième est une classification des mêmes images mais avec l'apprentissage auto-supervisé, Le chemin de la Figure 2.3 explique les étapes de développement de ce système. L'auto apprentissage est utilisé dans le but d'augmenter la précision et la validité des résultats.

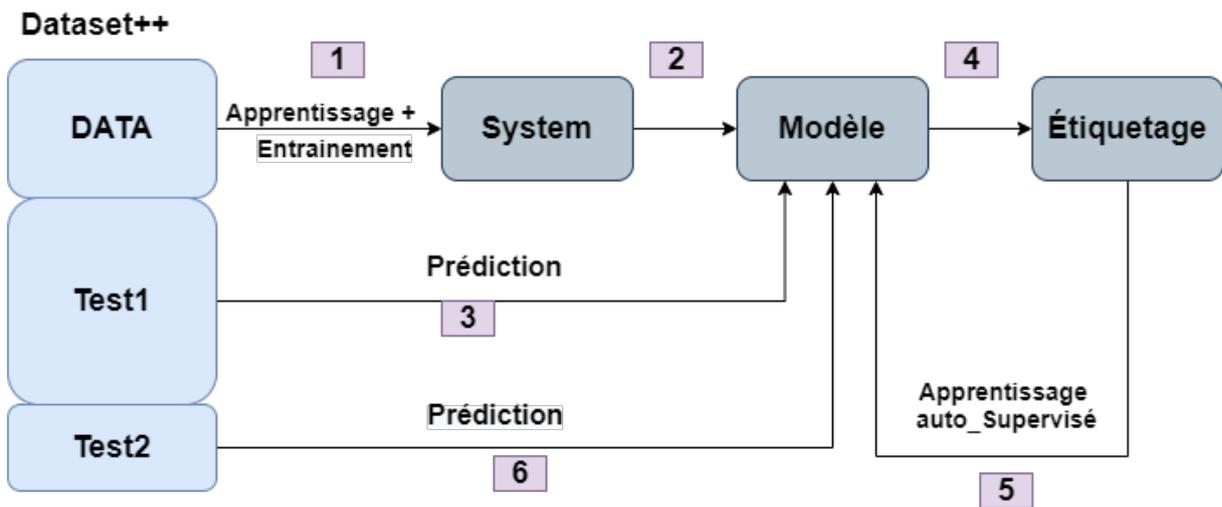


FIGURE 2.3 – Les étapes de développement du système de classification avec SSL

Étape 2 : cette étape aborde la segmentation des images CT détectées par le classifieur en tant que Covid-19 en suivant le modèle de la Figure 2.4

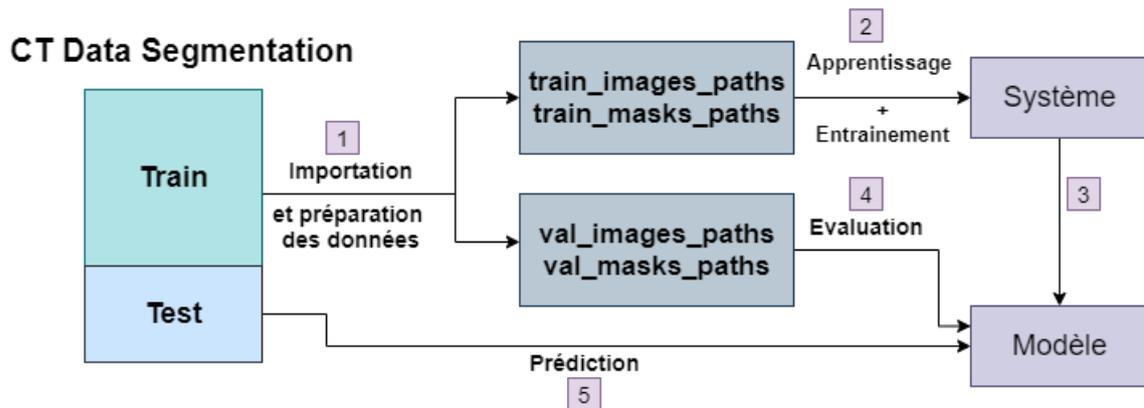


FIGURE 2.4 – Les étapes de développement du système de segmentation

2.6.1 Description du modèle utilisé pour la classification

L'idée générale de ce système est d'appliquer l'auto apprentissage pour construire un système de classification Covid-19 de haute performance et qui peut aider les radiologues pour prendre des décisions.

Pour les deux systèmes de classification que nous avons développée, nous utilisons "ResNet50", à cause des bons résultats que ce dernier a donné dans les systèmes de classification des images depuis sa création.

2.6.1.1 ResNet

ResNet, abréviation de Residual Network, est un type spécifique de réseau neuronal convolutif qui a été introduit en 2015 par Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun dans leur article "Deep Residual Learning for Image Recognition"[55].

Ce type de neurones a remporté la première place au concours de classification ILSVRC 2015 avec un taux d'erreur de 3,57% dans le top-5 et aussi dans les concours ILSVRC et COCO 2015 en détection et localisation de la base de données ImageNet et la détection plus la segmentation de la base de données Coco.

Le résiduel peut simplement être compris comme la soustraction d'une caractéristique apprise de l'entrée de cette couche. ResNet le fait en utilisant une connexion par raccourci (connectant directement l'entrée de la nième couche à certaines (n+x)

couche), il a prouvé que la formation de cette forme de réseaux est plus facile que la formation de réseaux neuronaux convolutionnels simples et aussi le problème de la dégradation de la précision est résolu[56]

ResNet a principalement été inspirées de la philosophie des réseaux VGGnets mais avec moins de filtres et moins de complexité que VGG. Il a également plusieurs variantes notamment : 'Resnet-34, Resnet-50, Resnet-101, Resnet-152, Resnet-1000, ...etc.', les numéros qui figurent dans le nom du modèle expriment le nombre de blocs résiduels[55] L'apprentissage résiduel est adopté à toutes les couches empilées, la Figure 2.5 présente l'architecture d'un bloc résiduel [55]

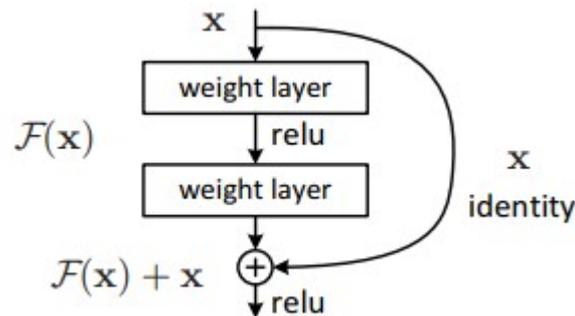


FIGURE 2.5 – l'architecture d'un bloc résiduel[55]

Le Resnet original avait 34 couches et utilisait des blocs à 2 couches, d'autres variantes avancées ont utilisé des blocs à 3 couches pour assurer une meilleure précision et moins de temps d'entraînement[55].

2.6.1.2 ResNet50

L'un des ResNets les plus populaires, l'architecture 'ResNet50' est basée sur le 'ResNet34', où chaque bloc de 2 couches dans le ResNet de 34 couches est remplacé par un bloc goulot d'étranglement de 3 couches, le nombre d'opérations en virgule flottante pouvant être effectuées par une entité de calcul en une seconde dans ce modèle est de 3,8 milliards (FLOPs en Anglais). Cette architecture donne une précision bien plus élevée que le mode ResNet à 34 couches, il est capable de reconnaître des objets avec une meilleure précision[55].

2.6.1.3 Les paramètres utilisé pour le modèle

Les paramètres que nous avons utilisé pour le modèle sont :

- **La taille des images** : la taille originale des images est de (512x512), nous les avons redimensionnées à (224x224)
- **La méthode de compilation** : pour configurer le modèle d'entraînement, nous avons utiliser la méthode de compilation '**compile**'
- **L'optimiseur 'Adam'** : il s'agit de l'un des optimiseurs les plus utilisés à cause de son rapidité. Il converge également de façon très rapide pour corriger la latence du taux d'apprentissage et le contraste élevé. Nous avons utilisé Adam avec une valeur égale à ($lr=0.0001$).
- **La fonction de perte (loss function)** : elle quantifie la différence entre le résultat attendu et le résultat produit par le modèle, pour notre modèle nous avons utilisé '**Categorical Crossentropy**', cette fonction de perte est utilisée lorsqu'il existe deux ou plusieurs classes d'étiquettes, dans notre cas deux classes (Covid et Normal).
- **Activation** : pour garantir une meilleure performance de calcul nous utilisons '**ReLU**' dans les couches cachées et dans la dernière couche de sortie nous avons essayé les deux fonctions d'activation '**Softmax**' et '**Segmoid**', et finalement nous avons décidé d'utiliser '**Softmax**', parce qu'elle donne des bons résultats par rapport à '**Segmoid**'.
- **Metrics** : est une fonction utilisée pour évaluer les performances du modèle, nous utilisons '**Accuracy**'.

D'après la figure 2.6, nous pouvons dire qu'il a plus de 23 millions de paramètres de formation.

Afin d'entraîner le modèle, nous avons utilisé la fonction **model.fit()** du package Keras, Cette fonction nous permet d'adapter littéralement nos données aux maquette.

- **Epochs** : les epochs sont le nombre de répétitions typiques de l'entraînement, dans notre programme le nombre des epochs est non déterminé, il est lié à la fonction d'arrêt

activation_48 (Activation)	(None, 7, 7, 2048)	0	['add_15[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['activation_48[0][0]']
flatten (Flatten)	(None, 2048)	0	['avg_pool[0][0]']
dense (Dense)	(None, 2)	4098	['flatten[0][0]']
=====			
Total params: 23,591,810			
Trainable params: 23,538,690			
Non-trainable params: 53,120			

FIGURE 2.6 – Le modèle ResNet50

- **Early Stopping** Pour éviter l’overfitting et à la base de cette fonction le modèle détermine et calcul le nombre d’epochs.
- **Batch size** : est un paramètre de l’algorithme d’apprentissage, représente le nombre des exemples de formation utilisés dans l’estimation du gradient d’erreur, nous utilisons 32.
- **Model check point (callback)** : qui est utilisé pour enregistrer le modèle.

Nous validons les modèles obtenus à partir de la classification avec DL et la classification avec SSL selon le taux de précision et le taux d’erreur.

2.6.2 Description du modèle utilisé pour la segmentation

Pour construire un système de segmentation du Covid-19 avec des images CT, nous avons utilisé le modèle "U-Net"[57] en raison de son succès remarquable et de ses bons résultats dans le domaine médical et la détection des anomalies et des tumeurs.

2.6.2.1 U-Net

U-net est l’un des cadres de segmentation les plus importants pour un réseau de neurones convolutionnels (CNN), il a été créé par un groupe de chercheurs en 2015 dans leurs documents "U-Net : Convolutional Networks for Biomedical Image Segmentation"[57].

Le modèle s'agit d'un assemblage de couches de convolution et de couches de " max pooling " permettant de réduire la taille de l'image pour diminuer le nombre de paramètres du réseau et de créer sa carte de caractéristiques[57].

Visuellement, l'architecture de U-Net a une forme de la lettre « U », elle est basée sur le réseau entièrement convolutif illustré dans [58], elle se compose d'un chemin de contraction (côté gauche) et un chemin expansif (côté droit). Le chemin de contraction suit l'architecture typique d'un réseau convolutionnel, à chaque étape d'échantillonnage descendant, le nombre de canaux fonctionnels a doublé. Dans le chemin expansif, chaque étape consiste en un échantillonnage ascendant de la carte des caractéristiques, suivi d'une convolution 2x2 qui réduit le nombre de canaux des caractéristiques à la moitié et dans la couche finale les créateurs du modèle utilisent une convolution 1x1. À la fin de cette architecture, la taille de sortie est égale à la taille d'entrée [57][59].

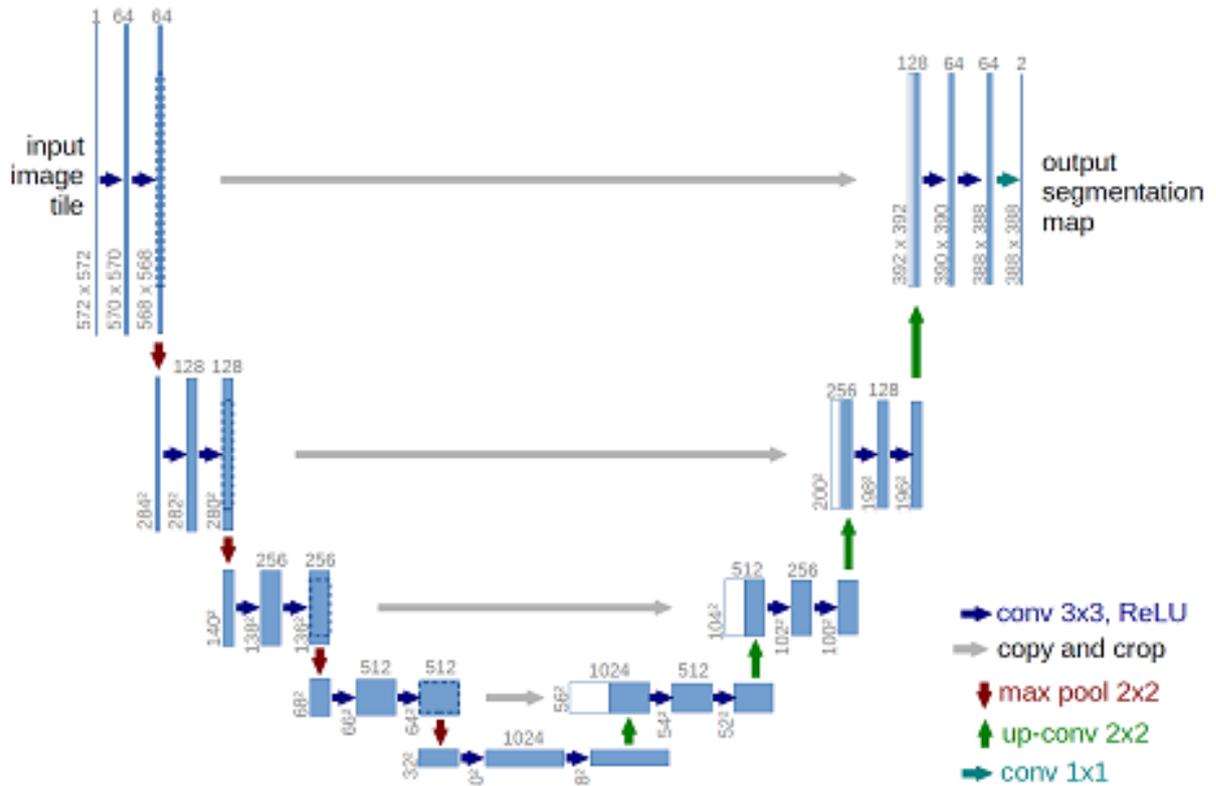


FIGURE 2.7 – Architecture U-Net (exemple pour 32x32 pixels dans la résolution la plus basse)[57]

Le principal avantage du modèle U-Net est le grand nombre de canaux dans la partie du chemin expansif (partie ascendante), qui permet au réseau de propager l'information contextuelle aux couches de résolution supérieure[60].

U-Net est capable de faire la segmentation d'une image en prédisant l'image pixel par pixel. De plus, il est efficace même avec un ensemble de données limité, donc il peut faire des bonnes prédictions basées sur peu de données d'entraînement et c'est grâce à l'augmentation de données[57][59].

Il existe plusieurs variantes très précises, et rapides de ce réseau adaptées à l'analyse des images biomédicales : U-SegNet[61], Residual U-Net[53], IU-Net[62], MR-UNet[63],etc.

2.6.2.2 Les paramètres utilisés pour le modèle

Le modèle utilisé contient plusieurs paramètres qu'il faut choisir soigneusement afin de maximiser la qualité des résultats. Le paragraphe suivant détaillées les paramètres que nous avons utilisés :

- **La taille des images imputées** : la taille originale des images= (512x512), nous avons les redimensionnées à (256x256).
- **La méthode de compilation** : Compile.
- **L'optimiseur** : Adam(learning rate=0.0001).
- **La fonction de perte (loss function)** : Binary Crossentropy.
- **Activation** : nous utilisons 'ReLU' dans tous les couches sauf la dernière ou nous avons utilisé 'Sigmoid'
- **Metrics** : pour évaluer le modèle de segmentation nous avons choisi de travailler avec 'IoU', alors qu'est-ce que c'est IoU?
- **IoU** : abreviation de "Intersection over Union", est une mesure d'évaluation commune pour la segmentation des images, selon [64] pour une classe individuelle, l'IoU est définie comme suit :

$$IoU = TruePositives / (TruePositives + FalsePositives + FalseNegatives)$$

```
def iou_coef(y_true, y_pred, smooth=1):  
    intersection = K.sum(K.abs(y_true * y_pred), axis=[1,2,3])  
    union = K.sum(y_true, [1,2,3])+K.sum(y_pred, [1,2,3])-intersection  
    iou = K.mean((intersection + smooth) / (union + smooth), axis=0)  
    return iou  
  
model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=[iou_coef])
```

FIGURE 2.8 – La fonction IoU avec la compilation du modèle

Pour commenc  l'entra nement du mod le sur les CT images avec la fonction **model.fit()** nous utilisons les param tres ci-dessous :

- **Epochs** : 135.

- **Btch size** : 32
- **Steps per epoch** : le nombre des images de train / batch size

```
batch_size = 32
steps = int(len(train_images_paths)/batch_size)
epochs = 200
for _ in range(epochs):
    i = _+1
    print(' Epoch', i , '/200 ')
    model.fit(datagen(train_images_paths, train_masks_paths, batch_size=batch_size), epochs=1, steps_per_epoch=steps)

Epoch 45 /200
65/65 [=====] - 71s 1s/step - loss: 0.0103 - iou_coef: 0.5397
```

FIGURE 2.9 – Les paramètres d’entraînement du modèle

2.7 Conclusion

La santé est un domaine privilégié de recherche dans le monde. Sur le plan informatique, cette recherche peut concerner entre autres l’analyse des images médicales pour l’assistance au pronostic et au diagnostic médicale.

Nous avons expliqué dans ce chapitre l’architecture du système proposé pour détecter COVID-19 dans les images CT des pommons humain. Nous avons également présenté les paramètres utilisés ainsi que leurs valeurs afin de faciliter la reproduction des résultats.

À la fin de ce chapitre, nous avons pris connaissance du virus Corona et de sa structure, ainsi que l’imagerie par la tomographie TDM (CT scan) et son rôle dans le diagnostic de Covid-19.

CHAPITRE 3

IMPLÉMENTATION, TESTS ET RÉSULTATS

3.1 Introduction

Dans ce dernier chapitre, et après avoir présenté la conception détaillée dans le Chapitre 2, nous expliquerons tout ce qui concerne l'implémentation des systèmes que nous avons proposés depuis l'environnement de développement et les bibliothèques jusqu'à l'explication des résultats.

3.2 Environnement de développement

Nous avons fait la classification et la segmentation dans le même environnement et avec le même langage de programmation.

3.2.1 Google Colab

Est un produit de Google Research, qui est un environnement Jupyter portable gratuit qui fonctionne entièrement dans le cloud et qui ne nécessite aucune configuration. Colab est adapté aux techniques de Machine Learning, il offre un service cloud

gratuit avec GPU gratuit ce qui il permet d'améliorer les compétences de codage en langage de programmation Python[65][66].

Plus important encore, les Notebooks créées par l'utilisateur peuvent être modifiés simultanément par d'autres membres de son choix afin d'assurer un meilleur travail collaboratif. Il intègre toutes les bibliothèques nécessaires pour la ML tels que : PyTorch, TensorFlow, Keras, OpenCV, etc., et permet d'importer ou enregistrer des Notebooks depuis ou vers Google Drive ainsi que la possibilité d'importer ou de publier des programmes ou des bases de données à partir de GitHub ou Kaggle[65][66][67].

3.2.2 Pourquoi les GPU ?

GPU signifie unité de traitement graphique. Pour la plupart des approches de ML ou DL, les GPU sont essentiels en raison de la quantité de données sur laquelle le programme va opérer[67]. Les exécutions des programmes Deep Learning sur un CPU peuvent prendre des mois ! Mais ces exécutions peuvent être assignées à des GPU pour un calcul plus rapide. Cependant, les GPU sont trop chers, c'est pour cette raison que Colab vient à la rescousse en fournissant **GPU Nvidia Tesla K80** gratuit. Donc on peut dire que Colab facilite le travail des développeurs surtout dans les domaines de l'Intelligence Artificielle[65][67].

3.2.3 Python

Python est un langage de programmation puissant et facile à apprendre utilisé pour la programmation générale, mais efficace pour la programmation orientée objet. Il a été créé par Guido van Rossum et sorti en 1991[68][69][70].

L'utilisation des espaces dans Python est très importante, il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Il dispose d'une bibliothèque standard vaste et complète et la caractéristique la plus importante est qu'il prend en charge de multiples paradigmes de programmation, y compris

orientés objet, impératifs, fonctionnels et procéduraux. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation[70].



FIGURE 3.1 – Le logo de Python[70]

3.3 Description des bibliothèques utilisées

1. Import the necessary packages

```
import os
import glob as gb
import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from skimage.io import imread, imshow
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import SGD, Adam
from keras.callbacks import EarlyStopping, ModelCheckpoint
```

(A)

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras.losses import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.metrics import *
import tensorflow.keras.backend as K
from tensorflow.keras.preprocessing.image import load_img, img_to_array

import numpy as np
from sklearn.utils import shuffle

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import cv2

import os
```

(B)

FIGURE 3.2 – (A) : Les bibliothèques utilisées pour la classification, (B) : Les bibliothèques utilisées pour la segmentation

3.3.1 Pandas

Est une bibliothèque open source sous licence BSD qui fournit des structures de données et des outils d'analyse de données haute performance et faciles à utiliser pour le langage de programmation Python[71].

3.3.2 NumPy

Est une bibliothèque destinée pour le calcul scientifique en Python, elle fournit des objets de tableau multidimensionnel, des objets dérivés (tels que les tableaux masqués et les matrices). C'est un assortiment de routines pour des opérations rapides sur des tableaux, y compris les mathématiques, la logique, la manipulation de formes, le tri, la sélection, les transformations discrètes de Fourier, l'algèbre linéaire de base, les opérations statistiques de base, la simulation aléatoire et bien plus encore[72].

3.3.3 Matplotlib

Matplotlib est une bibliothèque de traçage complète pour créer des visualisations statiques, animées et interactives en Python, elle a beaucoup d'extension comme l'extension de mathématiques numériques NumPy[73].

3.3.4 OpenCV

OpenCV a été lancé chez Intel en 1999 par Gary Bradsky. La première version est sortie en 2000, il a été plus tard soutenu par Willow Garage puis Itseez[74].

OpenCV-Python est une bibliothèque de liaisons Python destinées à résoudre les problèmes de vision par ordinateur en temps réel[74].

3.3.5 TensorFlow

TensorFlow est une plate-forme Open Source de bout en bout dédiée aux machines learning, créée et est maintenu par l'équipe Google Brain au sein de la machine de Google, une organisation de recherche en intelligence. TensorFlow publié sous la licence open source Apache 2.0 [75].

3.3.6 Keras

Keras est une API d'apprentissage profond écrite en Python, fonctionnant sur la plateforme d'apprentissage automatique TensorFlow. Le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux [76].

3.4 Les bases de données utilisées

Dans notre travail, nous avons utilisé trois bases de données :

3.4.1 Dataset

C'est une base de données Covid-19 qui contient 14684 images CT, composée de deux classes principales une classe nommée "Covid" pour les images CT des patients Covid-19 et la seconde nommée "Normal" pour les images CT des patients normaux.

Cette base de données a été créée dans le cadre d'une collaboration du laboratoire LabSTIC avec l'université "Yildiz Technical University" en Turquie sous la direction du professeur Muhammet Kurulay, l'architecture est définie dans le schéma de la Figure 3.3

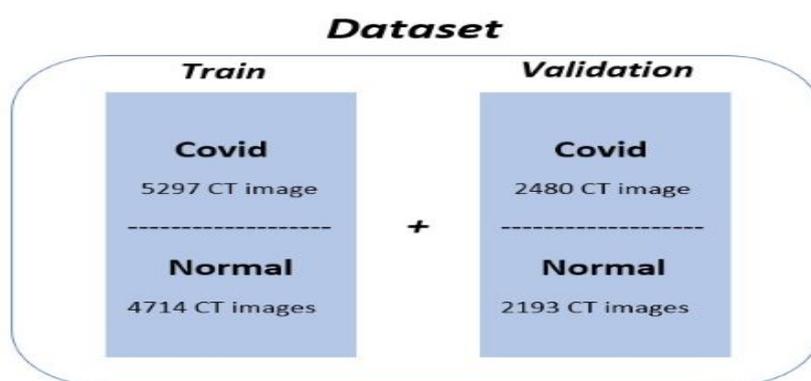


FIGURE 3.3 – Architecture de la base de données "Dataset"

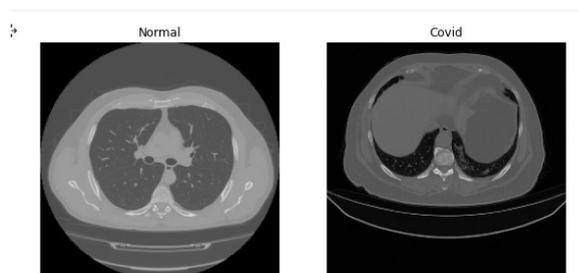


FIGURE 3.4 – La visualisation des images de "Dataset" dans Colab

3.4.2 Dataset++

Cette base est la même que la première base "**Dataset**" mais nous avons changé sa forme pour être cohérente avec le SSL, tout en préservons le nombre d'images et leur forme, la nouvelle architecture de la base et exprimé dans la Figure 3.5. Le tableau 3.1 montre la nouvelle distribution de la base.

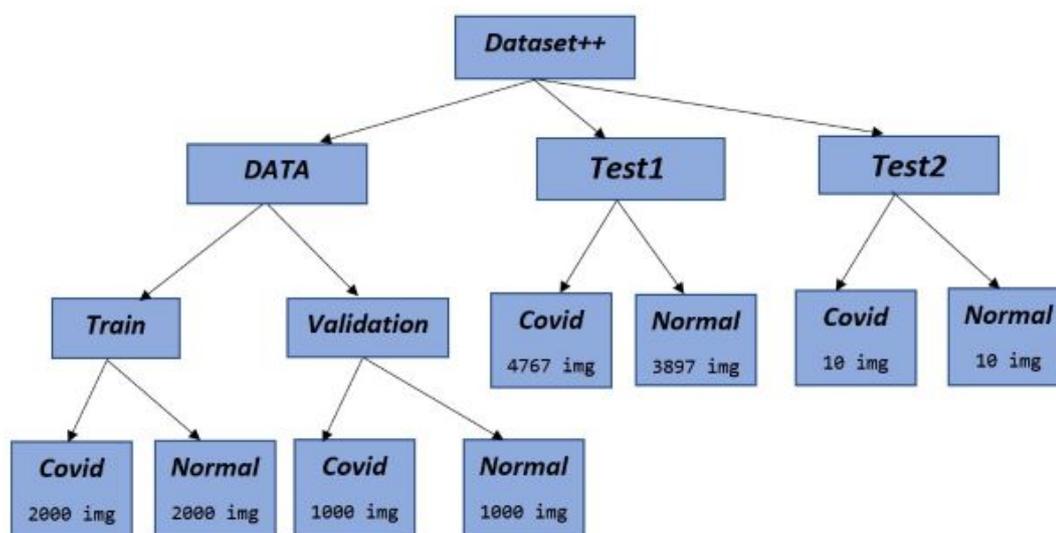


FIGURE 3.5 – Architecture de la base de données "Dataset++"

	DATA	Test1	Test2
Nombre des images	6000	8664	20
Pourcentage	40.87%	59%	0.13%

TABLE 3.1 – La distribution des images sur les trois parties de la base

3.4.3 CT Data Segmentation

Cette base, téléchargée à partir du site web "<https://www.kaggle.com/datasets/maedemaftouni/covid19-ct-scan-lesion-segmentation-dataset>", intervient dans l'étape de segmentation pour l'apprentissage du classifieur. C'est une grande base de données de tomodensitométrie pulmonaire pour COVID-19, contient 2729 images avec leurs masques.

Les masques de lésions de la COVID-19 et leurs cadres provenant de trois bases de données publiques. Tous les différents types de lésions sont mappés en couleur blanche pour assurer la cohérence entre les ensembles de données. Nous avons ajouté une partie qui contient 20 Covid CT images de la base de données "Dataset" pour faire un test du modèle.

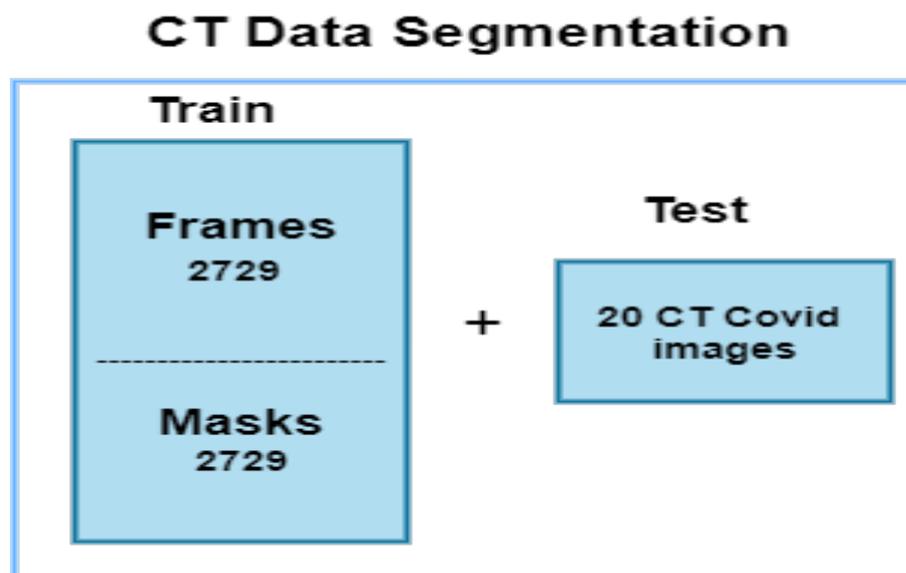


FIGURE 3.6 – Architectutre de la base de données "CT Data Segmentation"

3.5 Résultats du modèle de classification

3.5.1 Entraînement

Nous validons chaque modèle obtenu selon le taux de précision et le taux d'erreur.

Dans le système de classification avec l'apprentissage en profondeur que nous avons développé sur les images CT de la base de données "Dataset", le modèle a terminé l'entraînement avec un taux de précision $\approx 74\%$. La Figure 3.7 présente l'évaluation du modèle.

```
[18] model.load_weights('/content/drive/My Drive/MyModel1.h5')
      model.evaluate(X_valid, y_valid)

147/147 [=====] - 13s 90ms/step - loss: 0.4960 - accuracy: 0.7404
[0.49601051211357117, 0.7404237389564514]
```

FIGURE 3.7 – Évaluation du modèle de classification DL

Dans la classification par l'auto apprentissage que nous avons développé sur les images CT de la base de données "Dataset++", on peut diviser le processus de développement de ce système en trois grandes étapes, chaque étape étant liée à une partie spécifique de la base de données.

Étape 1 : Tout d'abord, et dans un nouveau Notebook Colab, nous avons commencé à entraîner le modèle avec les images de la partie "DATA" de la base, les résultats obtenus sont exprimés dans les deux figures suivantes :

accuracy=0.8305%, loss=0.4680%

```
[20] model.load_weights("/content/drive/My Drive/MyModel3.h5")
      model.evaluate(X_valid, y_valid)

63/63 [=====] - 6s 88ms/step - loss: 0.4680 - accuracy: 0.8305
[0.46795380115509033, 0.8305000066757202]
```

FIGURE 3.8 – Évaluation du modèle

Étape 2 : Nous pouvons appeler cette étape par l'étape d'auto apprentissage du modèle, où "MyModel2" fait une prédiction sur les images de la partie "Test1" en fonction de ce qu'il a appris de l'étape précédente, et puis il refait un entraînement sur les résultats qu'il a donnés. Nous avons enregistré le modèle obtenu sur le nom "MyModel3".

La deuxième partie de la base "Test1" est divisée en deux parties (Covid et Normal), donc nous utilisons la fonction "train-test-split" pour diviser l'ensemble de données en deux groupes d'apprentissage et de test. Les résultats obtenus sont exprimées

```
[29] x_train, x_valid, y_train, y_valid = train_test_split(X_test, y_test_pred, train_size= 0.8)

print('Training Samples =:      ',len(x_train) ,' and the number of labels is      ', len(y_train))
print('Validation Samples =:      ',len(x_valid) ,' and the number of labels is      ', len(y_valid))

Training Samples =:      6931 and the number of labels is      6931
Validation Samples =:      1733 and the number of labels is      1733

hist = model.fit(x = x_train, y = y_train , batch_size=32 , epochs=1, validation_data=(x_valid, y_valid), verbose=2)
217/217 - 72s - loss: 0.2938 - accuracy: 0.9074 - val_loss: 0.2801 - val_accuracy: 0.9008 - 72s/epoch - 331ms/step
```

FIGURE 3.9 – L'apprentissage auto-Supervisé du modèle avec une seule époque

dans la figure 3.10.

```
[35] #load the model
model= load_model('/content/drive/My Drive/MyModel3.h5')

model.load_weights("/content/drive/My Drive/MyModel3.h5")
model.evaluate(x_valid,y_valid)

55/55 [=====] - 6s 87ms/step - loss: 0.2801 - accuracy: 0.9008
[0.28014567494392395, 0.9007501602172852]
```

FIGURE 3.10 – Résultat de l'apprentissage auto-Supervisé

La figure 3.10 montre clairement que l'accuracy a augmentée (de 83% à 90%), suivi d'une diminution dans le loss (de 46% à 28%). Donc le SSL a apporté un plus et une force au système en termes de qualité et de précision des résultats.

Étape 3 : Dans cette étape nous avons importé la troisième partie de la base "Test3" qui contient 10 CT images infectées par Covid-19 et 10 CT images Normal dans le but

de faire un test du modèle construit par l'auto apprentissage pour pouvoir comparer les résultats de la prédiction du modèle avec les valeurs originaux.

```
print('y_prediction.shape = ' ,y_prediction.shape)
print('y_true.shape = ' ,y_true.shape)

y_prediction.shape = (20,)
y_true.shape = (20,)
```

```
[43] y_prediction

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1])
```

```
y_true

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

FIGURE 3.11 – Affichage des résultats de prédiction avec le modèle auto-suprvisé et les valeurs originaux

3.5.2 Classification

Nous avons développé un système de classification de Covid-19 avec les images CT basé sur le Deep Learning, et un autre système de classification de Covid-19 avec les images CT basé sur le l'apprentissage auto supervisé. Nous avons utilisé le même modèle et les mêmes paramètres pour les deux systèmes pour pouvoir faire une comparaison objective entre ces deux systèmes.

D'après les résultats de la Figure 3.12 , le système de classification basé sur DL a fait une erreur dans la classification de 4 images sur 20.

D'après les résultats de la Figure 3.13 , le système de classification basé sur l'auto apprentissage a fait une erreur dans la classification de 2 images sur 20.

Donc, nous pouvons dire que l'auto apprentissage à donné une performance supplémentaire en termes de qualité et de précision des résultats.

```

▶ from tensorflow.keras.models import load_model
   model=load_model('/content/drive/My Drive/MyModel2.h5')

[8] y_test2_pred = model.predict(X_test2)

[9] # predicted probabilities generated by sklearn classifier
   y_prediction = np.argmax(y_test2_pred, axis=1)
   y_true = np.argmax(y_test2, axis=1)

   print('y_prediction.shape = ', y_prediction.shape)
   print('y_true.shape = ', y_true.shape)

   y_prediction.shape = (20,)
   y_true.shape = (20,)

▶ y_prediction
   array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1])

[11] y_true
   array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

```

FIGURE 3.12 – Résultats de test de la classification du système basé sur DL

```

print('y_prediction.shape = ', y_prediction.shape)
print('y_true.shape = ', y_true.shape)

y_prediction.shape = (20,)
y_true.shape = (20,)

✓ [43] y_prediction
   array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1])

✓ ▶ y_true
   array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

```

FIGURE 3.13 – Résultats de test de la classification du système basé l'apprentissage auto supervisé

3.6 Résultat du modèle de segmentation

3.6.1 Préparation des données

La base de données utilisé contient 2729 images avec 2729 masks. Nous utilisons 2100(77%) images pour l'entraînement et le reste(23%) pour l'évaluation du modèle (Validation).

```
[ ] print('Total Number of Samples:', len(all_images_paths))
```

```
Total Number of Samples: 2729
```

```
[ ] train_images_paths = all_images_paths[:2100]
    train_masks_paths = all_masks_paths[:2100]
```

```
    val_images_paths = all_images_paths[2100:]
    val_masks_paths = all_masks_paths[2100:]
```

FIGURE 3.14 – Division d'images pour l'entraînement et l'évaluation du modèle

```
▶ print('Total Number of Samples in train_images_paths:', len(train_images_paths))
  print('-----')
  print('Total Number of Samples in train_masks_paths:', len(train_masks_paths))
  print('-----')
  print('Total Number of Samples in val_images_paths:', len(val_images_paths))
  print('-----')
  print('Total Number of Samples in val_masks_paths:', len(val_masks_paths))
```

```
Total Number of Samples in train_images_paths: 2100
```

```
-----
```

```
Total Number of Samples in train_masks_paths: 2100
```

```
-----
```

```
Total Number of Samples in val_images_paths: 629
```

```
-----
```

```
Total Number of Samples in val_masks_paths: 629
```

FIGURE 3.15 – Nombres des images dans chaque variable

3.6.2 Entraînement

Après avoir entraîné le modèle sur les images CT Covid avec 89 époques, il a atteint 0.6662 % d'IoU, alors nous avons décidé d'augmenter le nombre d'époques pour garantir une meilleure performance. Alors, nous avons essayé avec 120 époques

```
Epoch 88 .
65/65 [=====] - 74s 1s/step - loss: 0.0061 - iou_coef: 0.6641
Epoch 89 .
65/65 [=====] - 75s 1s/step - loss: 0.0061 - iou_coef: 0.6635
```

FIGURE 3.16 – Résultat avec 89 époques

puis avec 135 époques et la performance du modèle en termes de la fonction IoU a augmenté, mais à cause des limites de temps de session dans Colab nous n'avons pas pu ajouter d'autres époques.

3.6.3 Évaluation du modèle

Pour l'évaluation du modèle sur les images de la partie validation nous utilisons la fonction de Keras "keras.model.evaluate()", le résultat de l'évaluation est présenté dans la Figure 3.17

▼ Evaluate Model

```
[ ] batch_size = 32
steps = int(len(val_images_paths)/batch_size)
model.evaluate(datagen(val_images_paths, val_masks_paths, batch_size=batch_size), steps=steps)

19/19 [=====] - 12s 632ms/step - loss: 0.0157 - iou_coef: 0.6026
[0.015665218234062195, 0.6026473045349121]
```

FIGURE 3.17 – Évaluation du modèle

3.6.4 Prédiction

La prédiction est effectuée sur les 20 images de la partie Test de la base.

Premièrement, nous testons notre modèle sur une image, le mask obtenu par le système et le mask originale de l'image sont exprimé dans la Figure 3.20

▼ Prediction of the model

```
[ ] def predict(images):  
    pred = model.predict(images)  
    pred[pred>=0.5] = 1  
    pred[pred<0.5] = 0  
    return pred
```

FIGURE 3.18 – Fonction de prédiction du modèle de segmentation

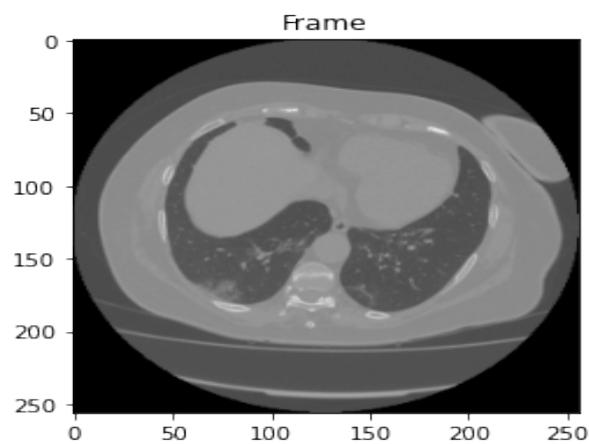


FIGURE 3.19 – L'image CT

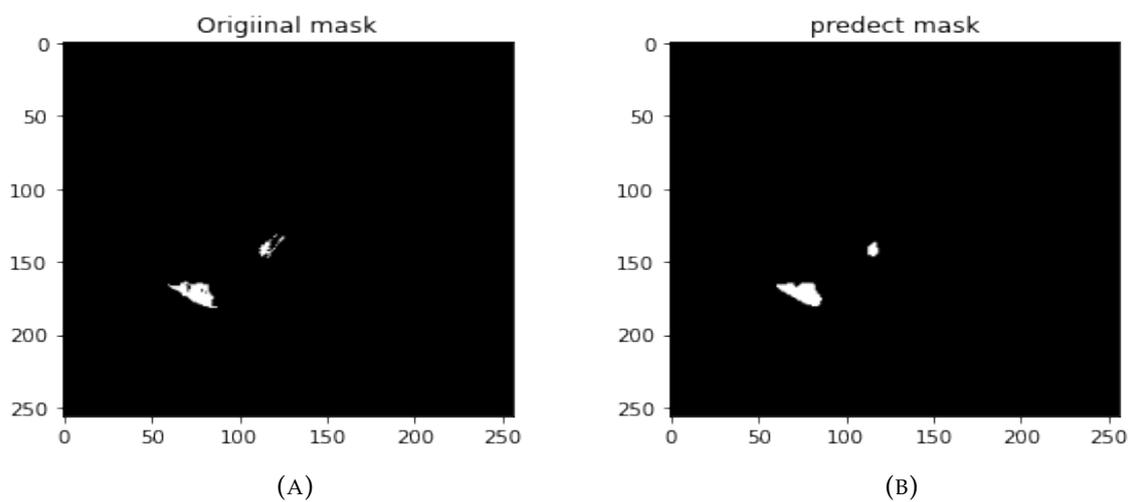


FIGURE 3.20 – (A) : Mask originale de l'image, (B) : Mask donné par le système

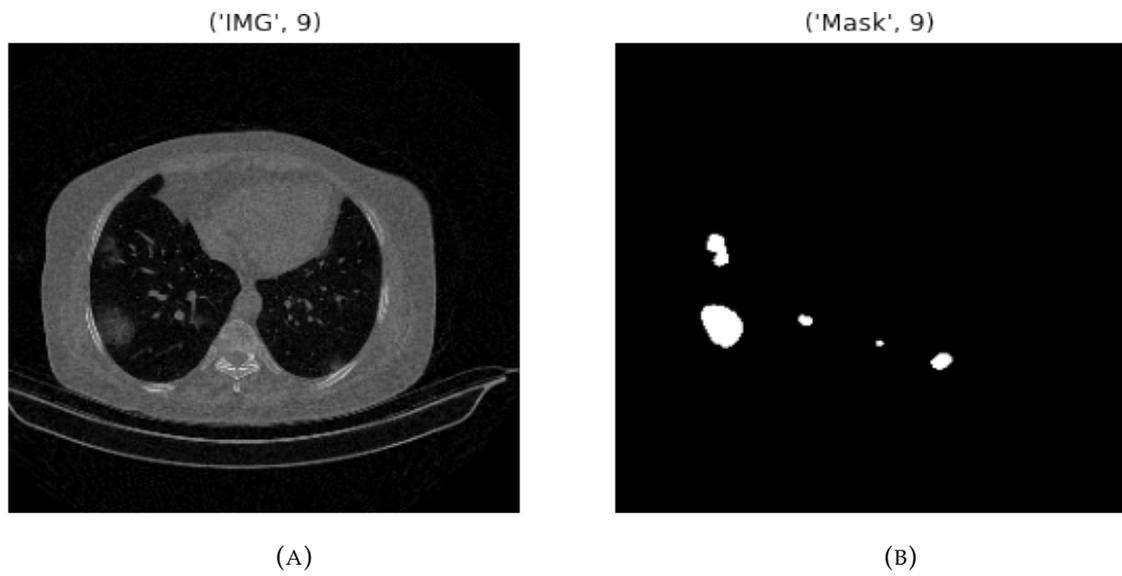


FIGURE 3.25 – (A) :Image 5, (B) : Mask par prédiction 5

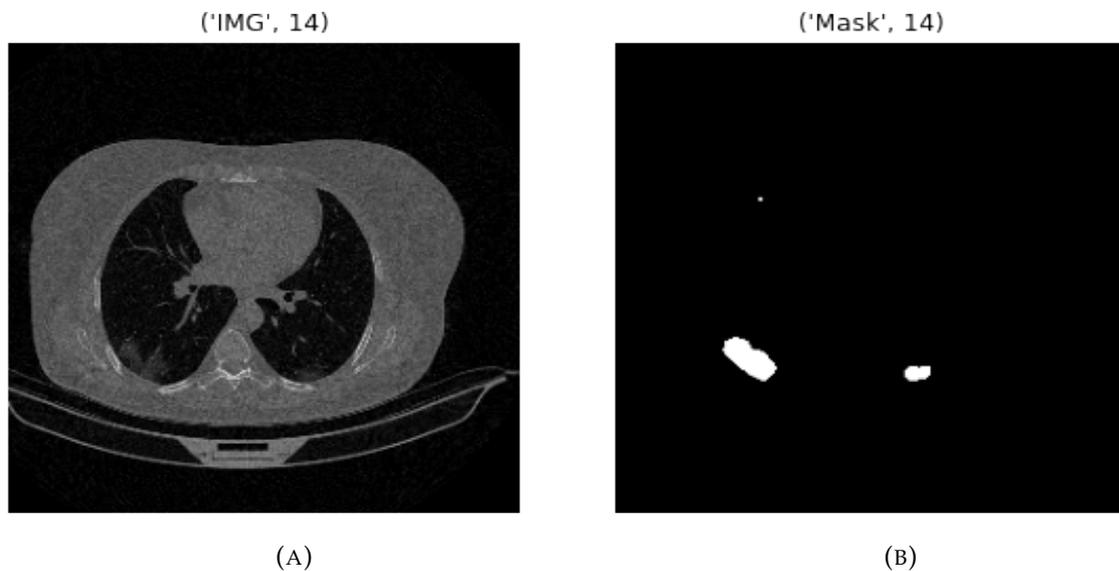


FIGURE 3.26 – (A) :Image 6, (B) : Mask par prédiction 6

En se basant sur les résultats qualitatifs, on peut dire que le modèle proposé a réussi à segmenter les zones atteintes de COVID-19. Les résultats peuvent être qualifiés d'encourageants par rapport à la vérité terrain.

3.7 Conclusion

Nous avons présenté dans ce chapitre les outils utilisés pour l'implémentation des modèles proposés. Nous avons également présenté quelques résultats qualitatifs et quantitatifs des modèles proposés sur un ensemble de 20 images CT de patients susceptible d'avoir le COVID-19. Les résultats montrent que le self Learning permet de combler le manque de données pour aboutir à une performance acceptable par rapport à une base riche de données. Les résultats montrent aussi que le modèle U-net donne de bons résultats de segmentation dans les images CT thoraciques.

CONCLUSION GÉNÉRALE

Des millions de personnes dans le monde souffrent de maladies pulmonaires et de voies respiratoires, et avec la pandémie de coronavirus des deux dernières années les médecins recherchent des techniques de diagnostic rapides et précises pour cette maladie.

La tomodensitométrie présente une technique de radiographie pour le diagnostic de plusieurs types de maladies telle que Covid-19. Cette technique permet d'éviter les problèmes de coût, de la disponibilité et de temps d'autres méthodes de diagnostic, et permet aussi d'ouvrir la voie au traitement des images médicales numériques avec les différents types d'apprentissages tels que l'apprentissage profond et l'auto apprentissage.

Le traitement d'image est né de l'idée de la nécessité de remplacer l'observateur humain par la machine. Dans ce projet, nous avons proposé un système qui permet de détecter COVID-19 dans les images CT d'un patient suspect et de localiser les régions infectées par le virus si ce dernier est malade en se basant sur des techniques de l'apprentissage profond et de l'auto apprentissage. Ce projet permet de faciliter la tâche de détection et de dépistage de cette maladie infectieuse.

Enfin, il sera intéressant dans le futur d'améliorer le système pour qu'il puisse détecter et classer d'autres types de maladie pulmonaire au niveau des hôpitaux.

BIBLIOGRAPHIE

- [1] L. M. de la SANTÉ, <https://www.who.int/fr/emergencies/diseases/novel-coronavirus-2019>, (accédé Le 10 Mars 2022).
- [2] J. MCCARTHY, « What is artificial intelligence? » *Article*, 2007.
- [3] C. BHATT, I. KUMAR, V. VIJAYAKUMAR, K. U. SINGH et A. KUMAR, « The state of the art of deep learning models in medical science and their challenges, » *Multimedia Systems*, t. 27, n° 4, p. 599-613, 2021.
- [4] N. K. CHAUHAN et K. SINGH, « A review on conventional machine learning vs deep learning, » in *2018 International conference on computing, power and communication technologies (GUCON)*, IEEE, 2018, p. 347-352.
- [5] S. SETIOWATI, E. L. FRANITA, I. ARDIYANTO et al., « A review of optimization method in face recognition : Comparison deep learning and non-deep learning methods, » in *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2017, p. 1-6.
- [6] L. GONOG et Y. ZHOU, « A review : generative adversarial networks, » in *14th IEEE conference on industrial electronics and applications(ICIEA)*, IEEE, 2019, p. 505-510.

-
- [7] B. MOHAMMED et B. BRAHIM, « L'apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu, » *UNIVERSITE KASDI MERBAH OUARGLA Faculté des Nouvelles Technologies de l'Information et de la Communication*, 2017.
- [8] O. CHAPPELLE, B. SCHOLKOPF et A. ZIEN, « Semi-supervised learning (chapelpe, o. et al., eds. ; 2006)[book reviews], » *IEEE Transactions on Neural Networks*, t. 20, n° 3, p. 542-542, 2009.
- [9] X. LIU, F. ZHANG, Z. HOU et al., « Self-supervised learning : Generative or contrastive, » *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [10] S. GIDARIS, P. SINGH et N. KOMODAKIS, « Unsupervised representation learning by predicting image rotations, » *arXiv preprint arXiv :1803.07728*, 2018.
- [11] N. SRIVASTAVA, E. MANSIMOV et R. SALAKHUDINOV, « Unsupervised learning of video representations using lstms, » in *International conference on machine learning*, PMLR, 2015, p. 843-852.
- [12] X. WANG et A. GUPTA, « Unsupervised learning of visual representations using videos, » in *Proceedings of the IEEE international conference on computer vision*, 2015, p. 2794-2802.
- [13] H.-Y. LEE, J.-B. HUANG, M. SINGH et M.-H. YANG, « Unsupervised representation learning by sorting sequences, » in *Proceedings of the IEEE international conference on computer vision*, 2017, p. 667-676.
- [14] I. MISRA, C. L. ZITNICK et M. HEBERT, « Shuffle and learn : unsupervised learning using temporal order verification, » in *European Conference on Computer Vision*, Springer, 2016, p. 527-544.
- [15] C. DOERSCH, A. GUPTA et A. A. EFROS, « Unsupervised visual representation learning by context prediction, » in *Proceedings of the IEEE international conference on computer vision*, 2015, p. 1422-1430.

- [16] R. ZHANG, P. ISOLA et A. A. EFROS, « Split-brain autoencoders : Unsupervised learning by cross-channel prediction, » in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, p. 1058-1067.
- [17] M. CARON, P. BOJANOWSKI, A. JOULIN et M. DOUZE, « Deep clustering for unsupervised learning of visual features, » in *Proceedings of the European conference on computer vision (ECCV)*, 2018, p. 132-149.
- [18] X. J. ZHU, « Semi-supervised learning literature survey, » 2005.
- [19] S. SHURRAB et R. DUWAIRI, « Self-supervised learning methods and applications in medical imaging analysis : A survey, » *arXiv preprint arXiv :2109.08685*, 2021.
- [20] A. JAISWAL, A. R. BABU, M. Z. ZADEH, D. BANERJEE et F. MAKEDON, « A survey on contrastive self-supervised learning, » *Technologies*, t. 9, n° 1, p. 2, 2020.
- [21] SCIENCESETAVENIR, https://www.sciencesetavenir.fr/fondamental/materiaux/premiere-des-neurones-de-rat-communiquent-avec-des-neurones-artificiels-par-internet_142252, (accédé Le 3 Mars 2022).
- [22] N. LIMARE, J.-L. LISANI, J.-M. MOREL, A. B. PETRO et C. SBERT, « Simplest Color Balance, » *Image Processing On Line*, t. 1, p. 297-315, 2011, <https://doi.org/10.5201/ipol.2011.11mps-scb>.
- [23] S. LEIJNEN et F. VEEN, « The Neural Network Zoo, » *Proceedings*, t. 47, p. 9, mai 2020. DOI : 10.3390/proceedings47010009.
- [24] E. LOCATELI, « La bibliothèque recherche des sciences de la Terre de l'Université de Lille au fil du temps : historique du patrimoine, un fonds au service de la communauté scientifique, » *Mémoires de la Société géologique du Nord*, XVII, p. 151-173, 2014.
- [25] S. L. HOROWITZ et T. PAVLIDIS, « Picture segmentation by a tree traversal algorithm, » *Journal of the ACM (JACM)*, t. 23, n° 2, p. 368-388, 1976.

- [26] A. K. JAIN, M. N. MURTY et P. J. FLYNN, « Data clustering : a review, » *ACM computing surveys (CSUR)*, t. 31, n° 3, p. 264-323, 1999.
- [27] D. WITHEY et Z. J. KOLES, « A Review of Medical Image Segmentation : Methods and Available Software, » 2008.
- [28] D.-P. FAN, T. ZHOU, G.-P. JI et al., « Inf-net : Automatic covid-19 lung infection segmentation from ct images, » *IEEE Transactions on Medical Imaging*, t. 39, n° 8, p. 2626-2637, 2020.
- [29] futura SANTÉ, <https://www.futura-sciences.com/sante/definitions/coronavirus-covid-19-18585>, (accédé Le 10 Mars 2022).
- [30] L. M. de la SANTÉ, <https://covid19.who.int/region/afro/country/dz>, (accédé Le 17 Mars 2022).
- [31] F. M. SHAH, S. K. S. JOY, F. AHMED et al., « A comprehensive survey of covid-19 detection using medical images, » *SN Computer Science*, t. 2, n° 6, p. 1-22, 2021.
- [32] L. M. de la SANTÉ, https://www.who.int/docs/default-source/coronaviruse/risk-comms-updates/update36_covid19-longterm_effects_fr.pdf?sfvrsn=1ccbb34e_4, (accédé Le 10 Mars 2022).
- [33] S. de radiodiagnostic et radiologie INTERVENTIONNELLE, <https://www.chuv.ch/fr/rad/rad-home/patients-et-familles/nos-examens/ct-scanner#>, (accédé Le 17 Mars 2020).
- [34] T. ULDIN, « Virtual anthropology—a brief review of the literature and history of computed tomography, » *Forensic sciences research*, t. 2, n° 4, p. 165-173, 2017.
- [35] ALAMYIMAGES, <https://www.universalis.fr/encyclopedie/imagerie-medicale/2-scanner-x/>, (accédé Le 17 Mars 2022).
- [36] Y. FANG, « Fang Y, Zhang H, Xie J, et al, » *Sensitivity of chest CT for COVID-19 : comparison to RT-PCR. Radiology*, t. 200432, 2022.

- [37] M.-Y. NG, E. Y. LEE, J. YANG et al., « Imaging profile of the COVID-19 infection : radiologic findings and literature review, » *Radiology : Cardiothoracic Imaging*, t. 2, n° 1, e200034, 2020.
- [38] O. GOZES, M. FRID-ADAR, H. GREENSPAN et al., « Rapid ai development cycle for the coronavirus (covid-19) pandemic : Initial results for automated detection & patient monitoring using deep learning ct image analysis, » *arXiv preprint arXiv :2003.05037*, 2020.
- [39] C. JIN, W. CHEN, Y. CAO et al., « Development and evaluation of an AI system for COVID-19, » 2020.
- [40] O. GOZES, M. FRID-ADAR, N. SAGIE, H. ZHANG, W. JI et H. GREENSPAN, « Coronavirus detection and analysis on chest ct with deep learning, » *arXiv preprint arXiv :2004.02640*, 2020.
- [41] U. ÖZKAYA, Ş. ÖZTÜRK et M. BARSTUGAN, « Coronavirus (COVID-19) classification using deep features fusion and ranking technique, » in *Big Data Analytics and Artificial Intelligence Against COVID-19 : Innovation Vision and Approach*, Springer, 2020, p. 281-295.
- [42] M. Z. ALOM, M. RAHMAN, M. S. NASRIN, T. M. TAHA et V. K. ASARI, « COVID_MNet : COVID-19 detection with multi-task deep learning approaches, » *arXiv preprint arXiv :2004.03747*, 2020.
- [43] X. CHEN, L. YAO et Y. ZHANG, « Residual attention u-net for automated multi-class segmentation of covid-19 chest ct images, » *arXiv preprint arXiv :2004.05645*, 2020.
- [44] A. MOBINY, P. A. CICALESE, S. ZARE et al., « Radiologist-level COVID-19 detection using CT scans with detail-oriented capsule networks, » *arXiv preprint arXiv :2004.07407*, 2020.

- [45] M. d. I. I. VAYÁ, J. M. SABORIT, J. A. MONTELL et al., « Bimcv covid-19+ : a large annotated dataset of rx and ct images from covid-19 patients, » *arXiv preprint arXiv :2006.01174*, 2020.
- [46] R. AMYAR AMINE Modzelewski, H. LI et S. RUAN, « Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia : Classification and segmentation, » *Computers in Biology and Medicine*, t. 126, p. 104 037, 2020.
- [47] A. NARIN, C. KAYA et Z. PAMUK, « Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, » *Pattern Analysis and Applications*, t. 24, n° 3, p. 1207-1220, 2021.
- [48] J. ZHAO, Y. ZHANG, X. HE et P. XIE, « Covid-ct-dataset : a ct scan dataset about covid-19, » *arXiv preprint arXiv :2003.13865*, t. 490, 2020.
- [49] X. WANG, Y. PENG, L. LU, Z. LU, M. BAGHERI et R. M. SUMMERS, « Chestx-ray8 : Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, » in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, p. 2097-2106.
- [50] G. HUANG, Z. LIU, L. VAN DER MAATEN et K. Q. WEINBERGER, « Densely connected convolutional networks, » in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, p. 4700-4708.
- [51] S. JIN, B. WANG, H. XU et al., « AI-assisted CT imaging analysis for COVID-19 screening : Building and deploying a medical AI system in four weeks, » *MedRxiv*, 2020.
- [52] A. SHOEIBI, M. KHODATARS, R. ALIZADEHSANI et al., « Automated detection and forecasting of covid-19 using deep learning techniques : A review, » *arXiv preprint arXiv :2007.10785*, 2020.
- [53] Y. PATHAK, P. K. SHUKLA, A. TIWARI, S. STALIN et S. SINGH, « Deep transfer learning based classification model for COVID-19 disease, » *Irbm*, 2020.

- [54] D. SINGH, V. KUMAR, M. KAUR et al., « Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks, » *European Journal of Clinical Microbiology & Infectious Diseases*, t. 39, n° 7, p. 1379-1389, 2020.
- [55] K. HE, X. ZHANG, S. REN et J. SUN, « Deep residual learning for image recognition, » in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 770-778.
- [56] N. SHARMA, V. JAIN et A. MISHRA, « An analysis of convolutional neural networks for image classification, » *Procedia computer science*, t. 132, p. 377-384, 2018.
- [57] O. RONNEBERGER, P. FISCHER et T. BROX, « U-net : Convolutional networks for biomedical image segmentation, » in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, p. 234-241.
- [58] J. LONG, E. SHELHAMER et T. DARRELL, « Fully convolutional networks for semantic segmentation, » in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, p. 3431-3440.
- [59] N. S. PUNN et S. AGARWAL, « Modality specific U-Net variants for biomedical image segmentation : a survey, » *Artificial Intelligence Review*, p. 1-45, 2022.
- [60] G. DU, X. CAO, J. LIANG, X. CHEN et Y. ZHAN, « Medical image segmentation based on u-net : A review, » *Journal of Imaging Science and Technology*, t. 64, n° 2, p. 20508-1, 2020.
- [61] P. KUMAR, P. NAGAR, C. ARORA et A. GUPTA, « U-segnet : fully convolutional neural network based automated brain tissue segmentation tool, » in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, p. 3503-3507.
- [62] N. S. PUNN et S. AGARWAL, « Inception u-net architecture for semantic segmentation to identify nuclei in microscopy cell images, » *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, t. 16, n° 1, p. 1-15, 2020.

- [63] N. IBTEHAZ et M. S. RAHMAN, « MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation, » *Neural Networks*, t. 121, p. 74-87, 2020.
- [64] KERAS, https://keras.io/api/metrics/segmentation_metrics/, (accédé Le 25 Mai 2022).
- [65] G. RESEARCH, <https://research.google.com/colaboratory/faq.html#:~:text=Colaboratory>, (accédé Le 21 Mai 2022).
- [66] TUTORIALSPPOINT, https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm#:~:text=To, (accédé Le 21 Mai 2022).
- [67] MEDIUM, <https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>, (accédé Le 21 Mai 2022).
- [68] G. VAN ROSSUM et F. L. DRAKE, *An introduction to Python*. Network Theory Ltd. Bristol, 2003.
- [69] G. VAN ROSSUM et F. L. DRAKE JR, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995, t. 620.
- [70] PYTHON, <https://www.python.org/>, (accédé Le 22 Mai 2022).
- [71] PANDA, <https://pandas.pydata.org/docs/>, (accédé Le 22 Mai 2022).
- [72] NUMPY, <https://numpy.org/doc/stable/>, (accédé Le 22 Mai 2022).
- [73] MATPLOTLIB, <https://matplotlib.org/stable/index.html>, (accédé Le 22 Mai 2022).
- [74] OPENCV, https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html, (accédé Le 24 Mai 2022).
- [75] TENSORFLOW, <https://www.tensorflow.org/?hl=fr>, (accédé Le 24 Mai 2022).
- [76] KERAS, <https://keras.io/about/>, (accédé Le 24 Mai 2022).