الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Télécommunications.

Spécialité : Réseaux et Télécommunications.

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Classification des pathologies pulmonaires COVID-19 à l'aide d'un apprentissage profond à partir d'images échographiques

Présenté par : ***BOUCHOUAREB wafi et DJABALLAH nouh***

Encadrant : *Neili Zakaria* *MAB* *Université BAdji Mokhtar Annaba*

Jury de Soutenance :

BOULMAIZ Amira	MCB	Université d'Annaba	Présidente
Neili zakaria	MAB	Université d'Annaba	Encadrant
NASRI Seif Allah El Mesloul	MCB	Université d'Annaba	Co-encadrant
Fezari Mohamed	Prof	Université d'Annaba	Examinateur

Année Universitaire : 2021/2022

Remerciments

Je remercie avant tout Allah tout puissant de m'avoir donné la force et la volonté pour accomplir ce modeste travail.

Au début, on souhaite adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

On tient à remercier tout particulièrement notre encadrant Dr. Zakaria Nieli pour nous avoir suivis et conseillés tout au long de la réalisation de ce mémoire, ainsi qu'à Dr. Seif Allah El Mesloul Nasri, le Co-encadreur, pour leurs précieux conseils et leurs orientations.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif des membres de notre famille, surtout nos parents qu'ils nous ont toujours encouragé moralement et matériellement et à qui on tient à les remercier

Enfin on tient à exprimer vivement nos remerciements avec une profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à sa réalisation car un projet ne peut pas être le fruit d'un seul Person

Merci àtous et àtoutes.

Dédicaces

Du profond de mon cœur, je dédie ce travail

A ma très chère mère Yamina qui m'a soutenu et encouragée durant ces années d'études. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon très cher père Djamal Tu as toujours été à mes côtés pour me soutenir et m'encourager, je ne pouvais espérer avoir un père plus formidable que toi j'ai eu beaucoup de chance que le bon Dieu t'a choisi pour que tu sois mon papa, je ne saurai point vous remercier comme il se doit.

A mes chers frères Youcef, Yasser et Hmenna, mes sœurs et tous mes chères cousines et voisins.

Aux responsables et le personnel de la société d'Algérie Télécom de Constantine qui m'ont aidé durant toute la période de mon stage.

A tous ceux qui ont une place dans mon cœur.

Ainsi que tous mes amis et collègues, en souvenir des moments agréables que nous avons passés ensemble.

Bouchouareb wafi.

Dédicaces

Je dédie ce mémoire :

À ma très chère mère

Quoi que je fasse ou que je dise, je ne pourrai jamais te remercier comme il se doit.

Ta présence à côté de moi a toujours été ma source de force

À mon très cher père

Tu as toujours été à mon côté pour me soutenir et m'encourager.

Que ce travail traduit ma gratitude et mon affection.

À mon cher frère et mes très chers familles et amis pour leur soutien et leur

Encouragement.

À tous ceux que j'aime. À tous ceux qui ont toujours cru en moi et poussé à réussir,

je vous dédie ce travail.

DjaballahNouh

Abstract

In recent years, many classification algorithms have been implemented to distinguishing between different lung diseases. Machine learning techniques have been widely used for lung disease classification and have particularly focused on deep neural networks, which appear advantageous with large training datasets. The objective of this work is to provide a fully automatic images classification system.We propose an alternative representation of the input data called ultrasound images. Our approach is implemented on two different architectures of deep neural networks – Visual Geometry Group 16 (VGG16) and AlexNet for the classification of pulmonary pathologies. The ultrasound database was chosen as an input to classify three types of lung conditions – COVID19, Pneumonia and Non-COVID19. The proposed system has 53% for precision and accuracy using VGG-16 and 54% accuracy and precision using AlexNet, these results are taken with imbalanced data and without preprocessing for our data

Keywords:

Lung diseases, machine learning, deep neural networks, fully automatic classification system, ultrasound images, VGG16, Alex Net, classification of linear regression pulmonary pathologies, COVID19 and Non-COVID19 imbalanced data, preprocessing data

Résumer

Ces dernières années, plusieurs algorithmes de classification ont été mis en œuvre pour distinguer les différentes maladies pulmonaires. Les techniques d'apprentissage automatique ont été largement utilisées pour la classification des maladies pulmonaires ; ces techniques se basent particulièrement sur les réseaux de neurones profonds, qui semblent avantageux avec larges ensembles de données d'entraînement. L'objectif de ce travail est de fournir un système de classification entièrement automatique. Nous proposons une représentation alternative des données d'entrée appeléeimages échographiques.Notre approche a été implémentée sur deux architectures différentes de réseaux de neurones profonds - Visual Geometry Group 16 (VGG16) et Alex Net pour la classification des pathologies pulmonaires. La base de données échographique a été choisie comme entrée pour classer trois types d'affections pulmonaires - COVID19, Pneumonie et Non-COVID19. Le système proposé à 53% de précision et de précision avec VGG-16 et 54% de précision avec AlexNet, ces résultats sont pris avec des données déséquilibrées et sans prétraitement pour nos données.

Mots clés :

Maladies pulmonaires., apprentissage automatique, réseaux neuronaux profonds, système de classification entièrement automatique, images échographiques, VGG16, Alex Net, classification des pathologies pulmonaires, COVID19 et Non-COVID19, données déséquilibrées, prétraitement de donées .

ملخص

في السنوات الأخيرة، تم تطبيق العديد من خوارزميات التصنيف للتمييز بين أمراض الرئة المختلفة.حيث تم استخدام تقنيات التعلم الآلي على نطاق واسع لتصنيف أمراض الرئة مع تركيز خاص على الشبكات العصبية العميقة،التي أثبتت كفاءتها مع مجموعات بيانات التدريب الكبيرة. الهدف من هذا العمل هو توفير نظام تصنيف آلي بالكامل، كما نقترح تمثيلاً بديلاً لبيانات الإدخال يسمى صور الموجات فوق الصوتية. تم تنفيذ مشروعنا على بنيتين مختلفتين للشبكات العصبية العميقة VGG16وAlex Netلتصنيف أمراض الرئة.تم اختيار قاعدة بيانات الموجات فوق الصوتية كمدخل لتصنيف ثلاثة أنواع من أمراض الرئة - COVID19 والالتهاب الرئوي وغير COVID19. يحتوي النظام المقترح على 53٪ للدقة والدقة باستخدام VGG-16 و 54٪ الدقة والدقة باستخدام AlexNet، ويتم أخذ هذه النتائج ببيانات غير متوازنة ودون معالجة مسبقة لبياناتنا

الكلمات المفتاحية:

أمراض الرئة، التعلم الآلي، الشبكات العصبية العميقة، نظام التصنيف آلي بالكامل، صور الموجات فوق الصوتية، VGG16، AlexNet، تصنيف أمراض الرئة COVID19وغير COVID19.ببيانات غير متوازنة , معالجة مسبقة للبيانات

Liste des figures

Liste des tableaux

Acronymes

IA	Intelligence Artificielle.
GPU	Graphiques procession unit.
OMS	Organisation mondiale de la Santé
LUS	Lung ultrasound.
ML	Machine Learning.
SVM	Support vector machine.
NLP	Natural language processing.
CNN	Convolutional neural network.
ReLU	Rectified linear activation unit.
DNN	Deep neural network.
NN	Neural network.
GAN	Generative adversarial network.
RNN	Recurrent neural network.
ILSVR	ImageNet Large Scale Visual Recognition Challenge.
FC	Fully connected.
VGG	Visual Geometry Group.
TPU	Tensor processing unit.
OpenCV	Open Source Computer Vision library.
NumPy	Numeric Python.
SciPy	Scientific Python.
API	Application Programming Interface.
VPP	Valeur prédictive positive.

Table de matière

Introduction générale

Le SRAS-CoV-2, un nouveau coronavirus, est à l'origine de l'épidémie de coronavirus de 2019 (COVID-19). Pour dépister les résultats du COVID-19, évaluer leur gravité ou suggérer d'autres étiologies de la maladie, les évaluateurs de patients peuvent choisir d'obtenir des radiographies. Grâce aux tests radiographiques, cela modifiera l'intérêt pour les traitements COVID-19 et peut influencer le traitement médical ou les décisions thérapeutiques de soutien, telles que l'hospitalisation, la nécessité d'une surveillance accrue ou l'hypothèse de risques de maladie. [1]

L'American Collège de Radiologie (ACR) recommande que la tomodensitométrie (**CT scan** ne soit pas utilisée pour le test COVID-19 ou le traitement de première ligne, pour les patients symptomatiques admis présentant des signes cliniques particuliers de CT. Le CT sera utilisé de manière vigilante et réservée, des protocoles de gestion des infections adéquats peuvent être utilisés avant le dépistage des futurs patients, et un scanner thoracique régulier n'indique pas qu'une personne n'a pas d'infection au COVID-19. Comparativement aux radiographies pulmonaires dans l'évaluation des patients COVID 19 présentant des symptômes de pneumonie, l'échographie pulmonaire (LUS) peut donner une meilleure précision de diagnostic. Le LUS est très sensible et de nombreuses radiographies thoraciques de maladies pulmonaires peuvent passer par là. L'échographie pulmonaire a déjà créé une méthode de détection de la pneumonie et des troubles pulmonaires associés [2]. L'approche du diagnostic des maladies pulmonaires a été préconisée comme favorable, en particulier dans les contextes de ressources limitées telles que les crises ou les pays à faible revenu. Cette approche a commencé le remplacement des rayons X comme diagnostic de première intention [3] [4]. Plusieurs chercheurs ont travaillé sur le développement de technologies pour identifier le COVID-19 à l'aide d'images radiographiques (tomodensitogrammes ou images radiographiques) depuis l'apparition du virus. Par contre, un nombre relativement limité de chercheurs travaillent sur des systèmes qui utilisent des images LUS et des sons de toux [5].

L'apprentissage en profondeur est utilisé dans la majorité des systèmes de dépistage du COVID-19, en particulier les réseaux de neurones convolutifs (CNN) pour la classification des images. Certains systèmes utilisaient des modèles CNN préformés, tandis que d'autres construisaient leurs

propres modèles CNN.Il existe deux types de modèles CNN préformés: 1)modèles très profonds et étroits, et 2) modèles modérément profonds et larges. Les modèles de la première catégorie considèrent la profondeur comme l'élément le plus important pour augmenter la précision dont Alex Net et VGG sont deux exemples. La largeur des filtres convolutifs est la caractéristique la plus critique dans la deuxième catégorie de modèles. Les réseaux résiduels étendus et ResNet sont deux exemples de la deuxième catégorie. En termes de précision et de quantité de paramètres, les deux types présentent des avantages et des inconvénients [6]. En fin, on ne veut pas jouer le rôle du docteur mais jute lui donner de l'aide

Notre projet de fin d'étude se compose des chapitres suivants :

Dans le premier chapitre nous allons définir

L'apprentissage automatique et l'intelligence artificielle, ensuit on parle de les réseau de neurones et l'apprentissage profonde dans le deuxiem chapitre, en termine par le troisieme chapiter qui traiter l'implementation et la conception de notre système

Chapitre 1 : Intelligence artificielle et apprentissage automatique

1.1 Introduction

L'apprentissage automatique est une forme d'IA qui permet à un système d'apprendre à partir de données plutôt que par une programmation explicite. Cette méthode de technologique diffère considérablement de la façon dont les organisations utilisaient les données auparavant. . Cependant, L'apprentissage automatique n'est pas un processus simple, c'est un domaine d'étude qui s'appuie sur des approches mathématiques et statistiques. Plus largement, il fait référence à la conception, l'analyse, l'optimisation, le développement et la mise en œuvre de telles méthodes.[7]

L'apprentissage automatique utilise une gamme d'algorithmes pour améliorer, caractériser et prédire les résultats en apprenant de manière itérative à partir des données. Il est possible de générer des modèles de plus en plus précis basés sur des données d'apprentissage au fur et à mesure que les algorithmes les absorbent. Lorsque vous entraînez votre algorithme d'apprentissage automatique à l'aide de données, le résultat est un modèle d'apprentissage automatique. Lorsque vous donnez une entrée à un modèle après l'entraînement, il renvoie une sortie. Un algorithme prédictif, par exemple, générera un modèle prédictif. Ensuite, lorsque vous alimentez les données du modèle prédictif, vous obtenez une prévision basée sur les informations que vous lui avez fournies. L'apprentissage automatique est actuellement requis pour le développement de modèles d'analyse. [8] [9]

1.2 L'intelligence artificielle

L'intelligence artificielle (IA) est définie comme "l'ensemble des théories et des méthodologies mises en pratique afin de créer des robots capables de simuler l'intellect humain". En conséquence, il englobe un ensemble de concepts et de technologies plutôt qu'un sujet distinct.En termes simples, l'intelligence artificielle (IA) fait référence à des systèmes ou à des robots qui imitent l'intellect humain pour effectuer des tâches et peuvent s'améliorer en fonction des données collectées par itération [9].Derrière l'intelligence artificielle se cache une multitude d'appareils ayant un point commun : l'apprentissage automatique. Cette technologie permet de stocker une grande quantité de données dans un cerveau virtuel ou un réseau de neurones [10]. Il existe alors deux thèses : l'une définissant l'intelligence artificielle comme une science cognitive et l'autre comme une branche de l'informatique.

Si l'intelligence artificielle est considérée comme une science cognitive, le défi devient une meilleure compréhension des phénomènes humains et des processus de raisonnement. Ceci est accompli grâce au développement de modèles mathématiques qui peuvent être utilisés pour décrire les raisonnements. L'intelligence artificielle a pour but le développement de modèles, de théories, et leur validation, non pas sur des sujets humains comme les autres sciences cognitives, mais par la programmation d'algorithmes. C'est alors le travail de l'intelligence artificielle de simuler le raisonnement humain. Il s'agit d'abord de modéliser les savoirs et les modes de raisonnement d'un expert humain, puis de les mettre à la disposition d'un non-informaticien. [11]

La deuxième thèse considère l'intelligence artificielle comme un sous-domaine de l'informatique. L'ordinateur n'est plus seulement un outil de recherche ; il est devenu l'axe fondamental de l'exploration. Il s'agit alors de concevoir des méthodes et des dispositifs qui tirent parti des capacités des ordinateurs pour effectuer des tâches qui auparavant auraient été considérées comme étant uniquement des capacités humaines, tâches qui nécessitent des raisonnements symboliques plutôt que des calculs arithmétiques [12]. Il est donc nécessaire de concevoir des programmes et des ordinateurs capables de traiter des problèmes pour lesquels aucune méthode de résolution directe et fiable n'est disponible [13]. Elle comprend maintenant les domaines dans lesquels nous traitons l'information sans savoir comment nous le faisons [14].

La disponibilité de plates-formes et d'outils informatiques plus sophistiqués, ainsi que de sources

de données de plus en plus utiles sur les patients, tant à l'intérieur qu'à l'extérieur des milieux cliniques, a le potentiel de révolutionner les soins de santé. La reconnaissance d'images à l'aide de méthodes d'intelligence artificielle (IA) est l'une des branches les plus développées du domaine, et ces techniques sont désormais couramment utilisées dans notre vie quotidienne. Dans le domaine de l'imagerie médicale, les approches basées sur l'IA sont particulièrement prometteuses, avec de nombreuses applications et un grand engouement pour la recherche de nouveaux biomarqueurs [15].

1.3 Types d’apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient.

1.3.1 Apprentissage supervisé

L'apprentissage supervisé ou supervised learning est une méthode de machine learning s'appuyant sur des données bien défini et une certaine compréhension de la façon dont ces données sont classifiées.Le but de l'apprentissage supervisé est de découvrir des modèles dans les données et de les appliquer au processus analytique. Ces données comprennent des caractéristiques liées aux balises qui définissent leur signification. Par exemple, vous pouvez créer une application d'apprentissage automatique en fonction des points communs détectés avec les symptômes connus d'autres patients (exemples), le système pourrait classer les nouveaux patients comme développant un risque estimé (probabilité) en fonction de leur analyse médicale (maladies spécifiques)[16][9].

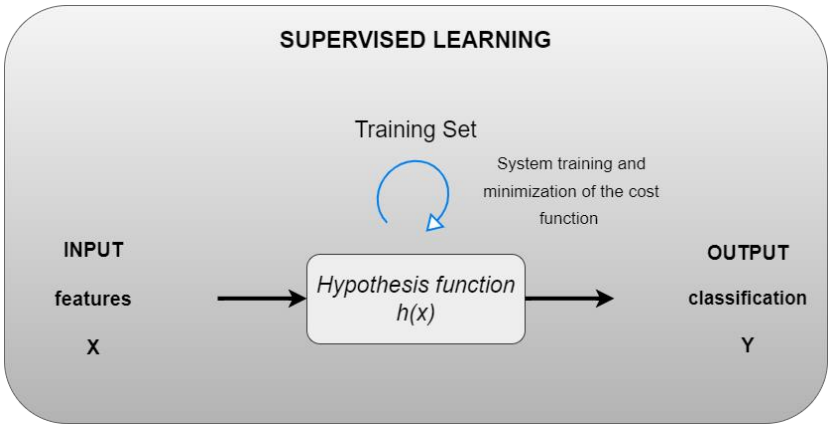

Figure 1: Principe de l'apprentissage supervisé

Il existe de nombreux algorithmes supervisés on peut citer certains d'entre eux qui sont très populaires en machine learning.

1.3.1.1 Régression linéaire

La popularité de la régression linéaire peut être attribuée à sa facilité d'utilisation. L'algorithme est simple à comprendre, simple et ne nécessite que quelques paramètres à configurer. Les algorithmes de ce type sont bien connus dans la pratique statistique et ils sont fréquemment utilisés pour prévoir les ventes ou les risques.La régression linéaire est la meilleure solution lorsque vous cherchez à

prédire votre valeur ou une classe[8].

1.3.1.2 Machine à vecteurs de support (SVM)

Une machine à vecteurs de support ou SVM (Support Vector Machine) est un algorithme qui divise les données en classes. Pendant l'entrainement, le SVM trouve une ligne qui divise un ensemble de données en classes spécifiques et maximise la marge (la distance entre la limite de séparation et l'échantillon le plus proche) pour chaque classe. Après avoir appris les lignes catégorielles, le modèle peut les appliquer à de nouvelles données.

Les SVMs entrent dans la catégorie des "classificateurs linéaires", l'algorithme est idéal pour identifier des classes simples, qui sont séparées par des vecteurs appelés hyperplans. Des algorithmes peuvent également être programmés pour des données non linéaires qui ne peuvent pas être clairement séparées par des vecteurs. Cependant, pour les données d'entraînement super complexes, les systèmes de classes deviennent plus petits et plus difficiles à identifier et nécessitent un peu plus d'assistance humaine. Les machines à vecteurs de support sont très utilisées dans la finance. Elles offrent une grande précision sur les données actuelles et futures. Les SVMs dits non linéaires sont souvent mis à contribution pour classifier des images ou des mots, des phrases et des entités (NLP) [8].

1.3.1.3 Arbre de décision

Les algorithmes d'arbre de décision dessinent des données dans des branches pour montrer les résultats possibles de diverses opérations. Ils classent et préditent les variables de réponse en fonction des décisions passées,. Les résultats de l'arbre de décision sont faciles à interpréter. Les data scientistes citoyens n'auront aucun mal à les expliquer. Même si l'ensemble de données d'entrée est incomplet, les décisions et leur impact possible sur le résultat final sont faciles à voir. Cependant, lors de la combinaison de grandes quantités de données et de variables complexes, les arbres de décision deviennent difficiles à lire. C'est pourquoi ils sont utilisés pour les décisions à faible risque[8].

1.3.2 Apprentissage non supervisé

L'apprentissage non supervisé est un terme utilisé pour décrire une situation d'apprentissage automatique dans laquelle les données ne sont pas étiquetées, et que le nombre de classes et leur nature n'ont pas été prédéterminées. En conséquence, il s'agit de déterminer quelles structures se cachent derrière ces données non étiquetées.L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine, l'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données[10].

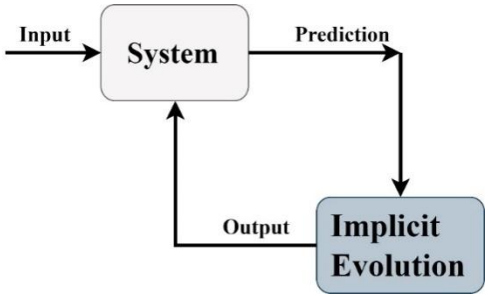

Figure 2: Principe de l'apprentissage non supervisé

Il existe aussi de nombreux algorithmes non-supervisés, on peut citer certains d'entre eux qui sont très populaires :

1.3.2.1 La répartition en K-moyennes (K-means)

L'algorithme K-means repose sur une approche itérative pour regrouper les points de données en fonction de caractéristiques similaires, et les affectations K-means sont connues pour être précises, tout en étant capables de traiter des groupes de données dans un temps relativement court. Cet algorithme est également utilisé par les éditeurs de moteurs de recherche pour fournir des résultats pertinents, ou par des entreprises souhaitant catégoriser le comportement des utilisateurs. Cette technique est également efficace dans l'analyse des performances des ordinateurs.

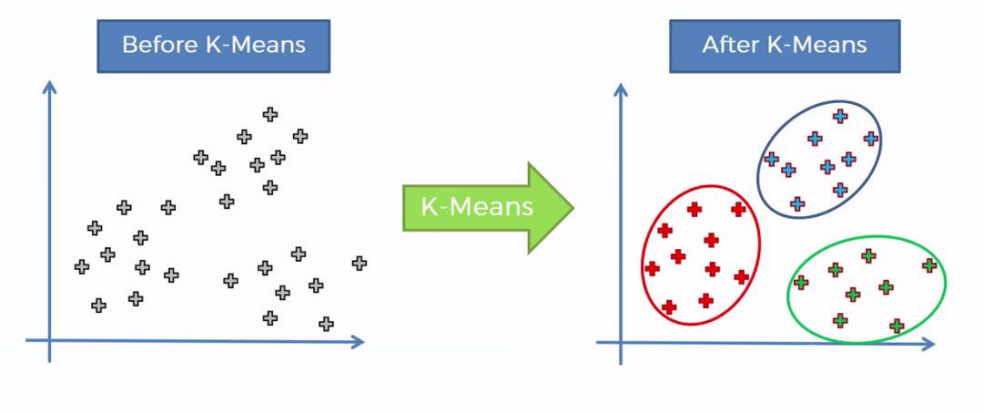

Figure 3: La répartition en k-means

1.3.2.2 Les algorithmes Apriori

Dans le domaine de l'apprentissage des règles d'association, l'algorithme Apriori est un algorithme d'exploration de données, il cherche les affinités entre deux éléments d'un jeu de données afin d'identifier s'il y a une corrélation négative ou positive entre eux.Cet algorithme est utilisé pour déterminer une catégorie en reconnaissant les qualités qui apparaissent souvent dans une donnée.

1.4 Conclusion

L'apprentissage automatique, tel que nous le comprenons maintenant, est une méthode permettant d'apprendre aux robots à exécuter les mêmes tâches que le cerveau humain, bien qu'un peu plus rapidement et mieux. L'apprentissage automatique est composé de modélisation (ensemble d'hypothèses + fonction objectif) et d'optimisation, et un ensemble de données approprié pour l'apprentissage des connaissances est nécessaire pour réaliser l'apprentissage automatique. Nous introduisons la notion de base, la catégorisation, la structure et les critères de l'apprentissage automatique d'un point de vue théorique. [17]

Chapitre 2 : Réseau de neurones et apprentissage profond

2.1. Introduction

Les réseaux de neurones sont l'un des plus beaux modèles de programmation d'inspiration biologique qui permet à un ordinateur d'apprendre à partir de données d'observation. L'apprentissage profond en anglais « DeepLearning », est un puissant et important ensemble de techniques d'apprentissage dans les réseaux de neurones.Lesréseaux neuronaux et l'apprentissage profond offrent actuellement les meilleures solutions à des problèmes très compliqués, Ils sont largement utilisés par de nombreux chercheurs dans de nombreuses applications différentes telles que la robotique [17-18], la reconnaissance vocale [19, 20], la reconnaissance des visages humains [21, 22], les applications médicales [23-24], la fabrication [25, 19] et l'économie [20, 26] ainsi que la traduction automatique.

Dans le cadre de ce chapitre nous allons présenter une étude sur les réseaux de neurones, Ensuite, nous allons présenter l'apprentissage profond tout en mettant l'accent sur le fonctionnement des réseaux de neurones convolutifs (CNN), qui sont utilisés dans notre travail.

2.2. Neurones

Le cerveau est représenté par un grand réseau de neurones, la quantité de neurones et leur structure en réseau déterminent un niveau de l'intellect de cette créature. L'idée de base derrière le neurone est de recevoir des signaux d'autres neurones et de combiner un signal transformé basé sur des entrées et le transférer à d'autres neurones, neurone naturel se compose de trois parties principales : dendrite, noyau et axone.

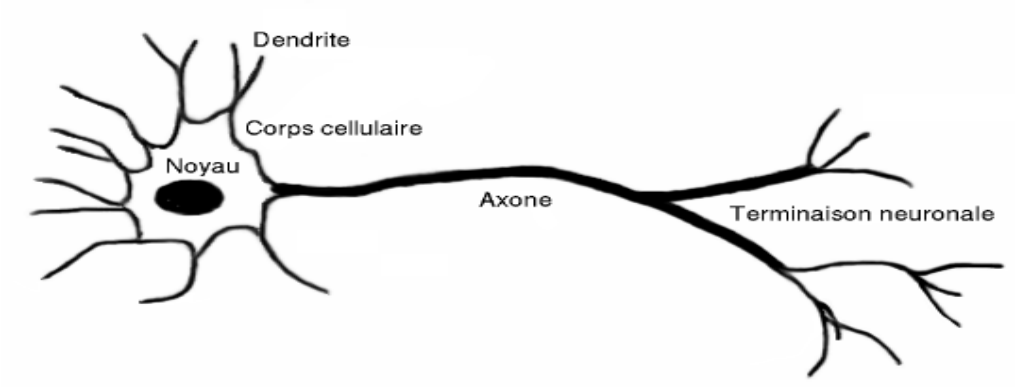

Figure 4 : Composition de neurone [27]

Le neurone artificiel est représentation mathématique du vrai neurone qui fonctionne par des règles de base. Le nucleus est remplacé par simple fonction mathématique appelée « fonction d'activation », lesdendritessontles fonctionsde multiplication, ils multiplient le signal entrant par un poids qui est toujours changé après l'apprentissage, axone n'a pas de fonctions de transformation.

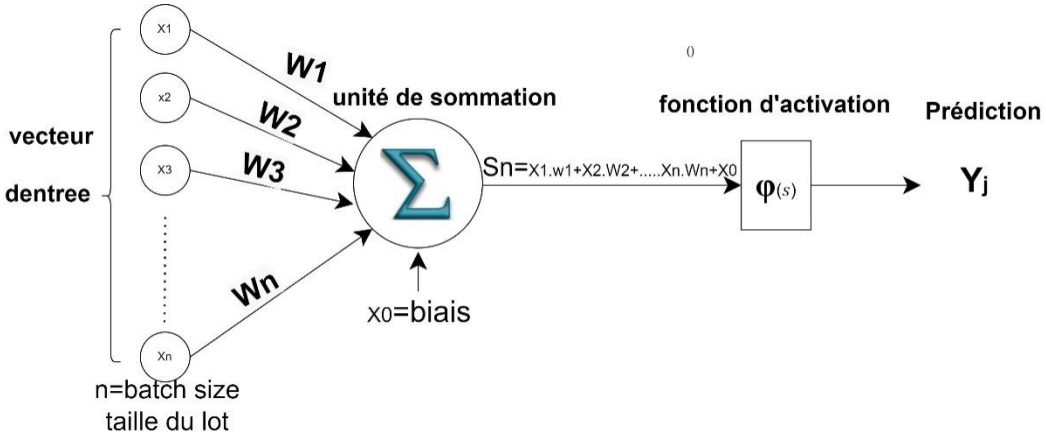

Figure 5 : Modèle non linéaire d'un neurone artificiel

Il se trouver deux types de neurones artificiels. Les neurones transitionnels sont ceux qui effectuent un calcul qui ne dépend d'aucun poids, Les neurones intelligents sont des neurones qui contiennent un poids qui ajuste les processus qui se produisent pendant l'activation

2.3. Réseaux de neurones

A cause de la complexité du réseau neuronal naturel, nous ne pouvons pas en faire un modèle complet, lacréérationde modèle qui répète tous les processus influençant les neurones serait un problème très difficile à résoudre. Même si, maintenant, les modèles construits pour reconnaître les images ou la voix sont relativement très simples par rapport à un cerveau d'un plus petit être.

Les réseaux de neurones artificiels sont auto-apprenants et peuvent produire de meilleurs résultats lorsque davantage de données sont disponibles. Un réseau de neurones artificiels est un ensemble de plusieurs perceptrons/neurones sur chaque couche. Le réseau neuronal se compose de trois couches. La première couche est la couche d'entrée. Il possède des neurones d'entrée qui envoient des informations à la couche cachée. Ce dernier effectue des calculs sur les données d'entrée et transmet la sortie à la couche de sortie. Il comprend les poids et la fonction d'activation.

L'architecture du réseau de neurones artificiels est illustrée dans la figure suivante :

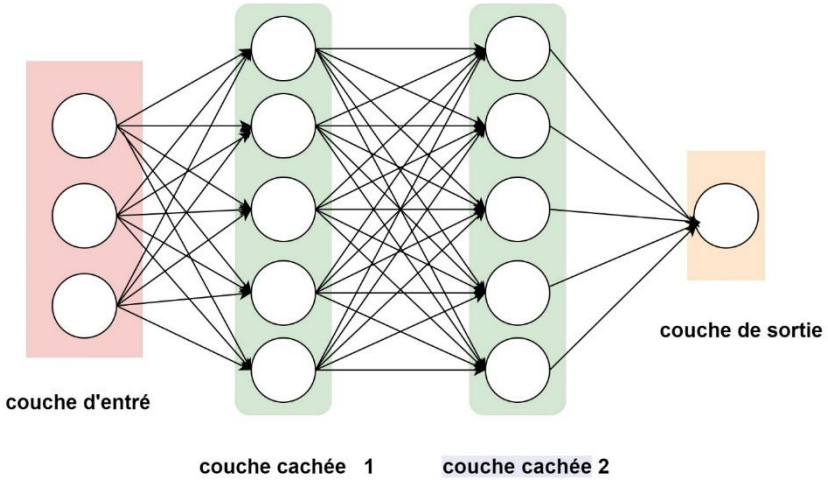

Figure 6: Architecture des réseaux de neurones

2.4 Fonction d'activation

La fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation".La fonction d'activation définit la sortie du neurone en fonction du champ local induit. On peut citer deux types de fonctions d'activation sont : la sigmoïde et la fonction Relu.

2.4.1 La Fonction ReLU

Elle est utilisée comme fonction d'activation dans le contexte du réseau de neurones artificiels pour sa simplicité de calcul en particulier de sa dérivée. Elle remplace les valeurs négatives reçues en entrées par des zéros.

Mathématiquement la fonction Unité Linéaire Rectifiée (ou ReLU pour *RectifiedLinear Unit*) est définie par :

$$f(x) = max(0, x) \qquad (1)$$

ReLu est appliquée à chaque pixel d'une image après convolution, Comme mentionne auparavant, cette fonction remplace chaque valeur négative par un 0. Si cette fonction n'est pas appliquée, la fonction créée sera linéaire.

Quel que soit la valeur de x.

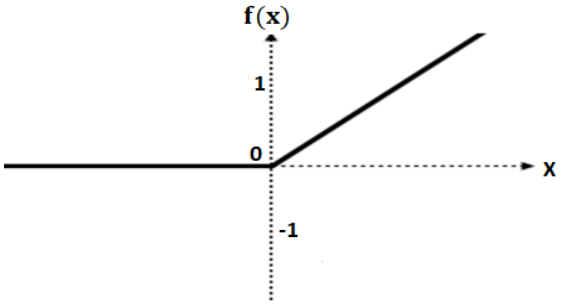

Figure 7: l'allure de la fonction ReLU

2.4.2 La Fonction d'activation sigmoide

L'exemple de la fonction sigmoïde est la "fonction logistique" et c'est une fonction continue, dans l'intervalle [0,1] en peut définir sa sortie et parmi c'est avantage sa dérivée existe en tout point, et elle est définie par :

$$\varphi(s) = \frac{1}{1 + e^{-\alpha s}} \qquad (2)$$

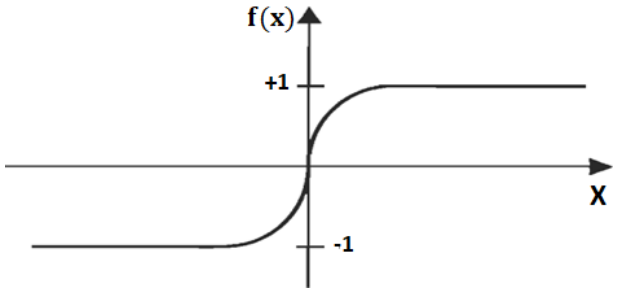

Figure 8: l'allure de la fonction sigmoide

Il y a aussi **la fonction de seuil**, définite par la relation suivante :

$$\varphi(s) = \begin{cases} 0, & s < 0 \\ 1, & s \geq 0 \end{cases} \qquad (3)$$

2.5. Apprentissage profond

L'apprentissage profond également appelé un réseau neuronal profond en anglais « Deep Neural Network (DNN) » est une classe d'algorithmes d'apprentissage automatique qui est un sous domaine de l'intelligence artificiel. DNN appartient au domaine global qui est la vision par ordinateur. L'apprentissage profond utilise l'apprentissage supervisé pour fournir à un programme des milliers de données étiquetées, qu'il devra apprendre à reconnaître. L'apprentissage profond s'appuie sur le concept des réseaux de neurones expliquer précédemment.Il existe différents types de réseaux neuronaux mais ils sont toujours constitués des mêmes composants : neurones, synapses, poids, biais et fonctions. Ces composants fonctionnent de manière similaire au cerveau humain et peuvent être entraînés comme n'importe quel autre algorithme ML.L'apprentissage profond a été appliqué dans plusieurs domaines nécessaire et il a prouvé son efficacité notamment dans: le domaine médical où certains programmes qui utilisent l'apprentissage profond sont parfois plus fiable que l'analyse humaine, domaine de l'automobile [28], le domaine militaire [29].Dans le domaine d'apprentissage profond on trouve. Réseau neuronal artificiel ANN(Artificial neural

network),Réseau neuronal convolutif CNN ,Réseau neuronal Récurrent RNN, Réseaux Génératifs Adversariaux GAN(Générative Adversarial Networks).

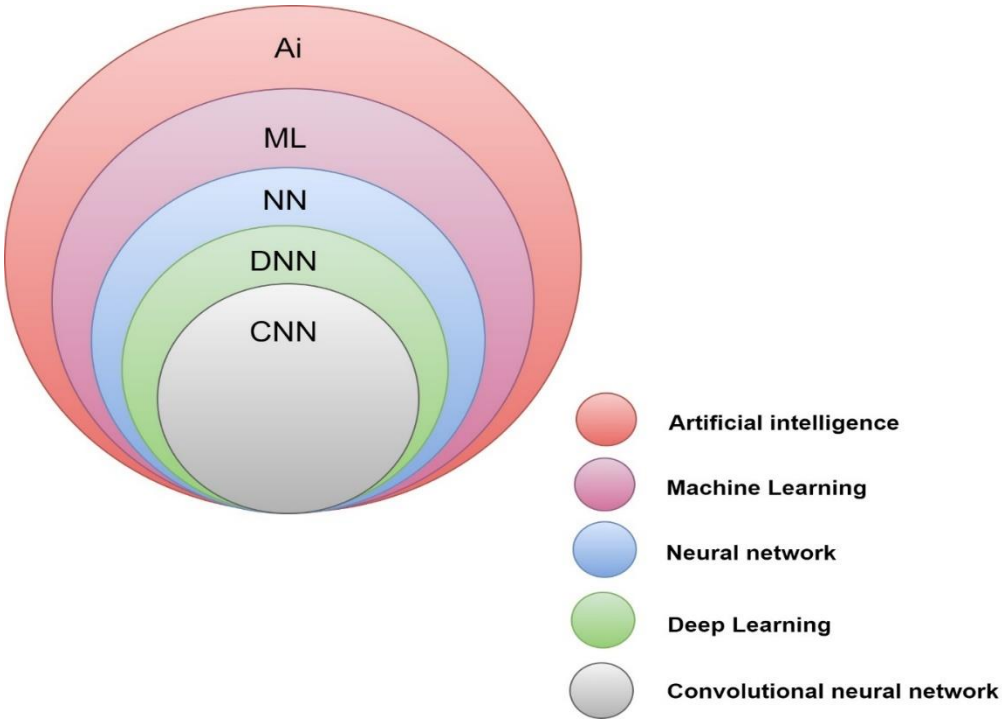

Figure 9 : Schéma représente la relation entre Ai et ML-NN- DNN -CNN

2.6. Réseau neuronal convolutif

Les réseaux de neurones convolutifs ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé, ils sont un type particulier de réseau neuronal artificiel qui utilise une opération mathématique appelée convolution dans au moins une de ses couches au lieu de la multiplication générale de la matrice. Ils sont conçus pour traiter les données de pixels pour la reconnaissance et le traitement des images et l'identification des objets.

Il existe quatre types de couches pour un réseau de neurones convolutif : la couche de **convolution**, la couche de ***pooling,*** la couche de **correction ReLU**et la couche ***fully-connected*** comme montre la figure 10.

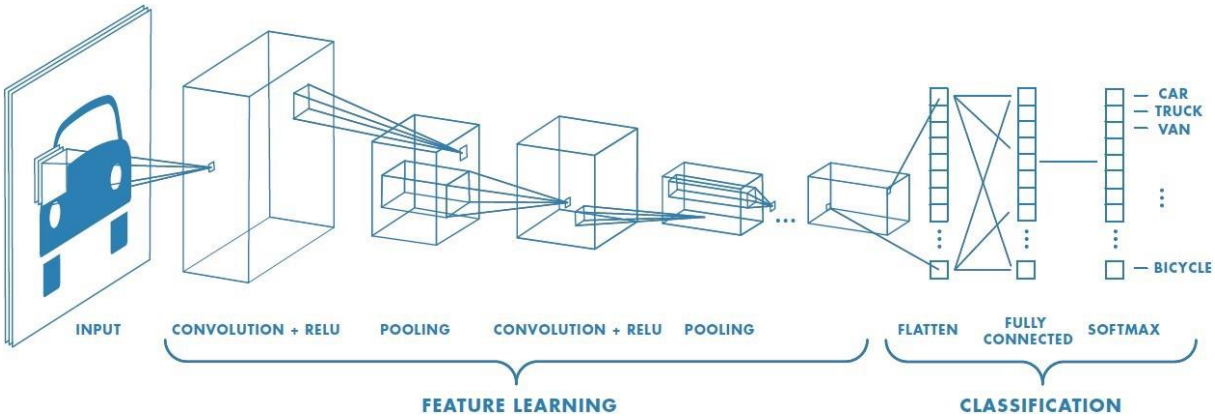

Figure 10 : Schéma représentant l'architecture d'un CNN.[30]

- Traitementdes donnéesentrées par Lacouchedeconvolution(CONV),
- La compression de l'informationenréduisantlatailledel'image-intermédiaire(souvent par sous-échantillonnage) Par La couche de Pooling (POOL),
- Lacoucheentièrementconnectée(FC),est une couchedetypeperceptron.
- Fonctiond'activationnon-linéaireReLU.
- Lacouche deperte(LOSS).

2.6.1. La couche convolution

La convolution, d'un point de vue simple, est l'application un filtre mathématique à une image. D'un point de vue plus technique, il s'agit de faire glisser de gauche à droite et de haut en bas une matrice par-dessus une image pour chaque pixel. Cette technique nous permet de trouver des parties de l'image qui pourraient nous être intéressantes,Les paramètres pris en compte lors de cette partie sont **la profondeur** appelé nombre de noyaux de convolution., **le pas** et **la marge** qui permet de contrôler la dimension spatiale du volume de sortie.

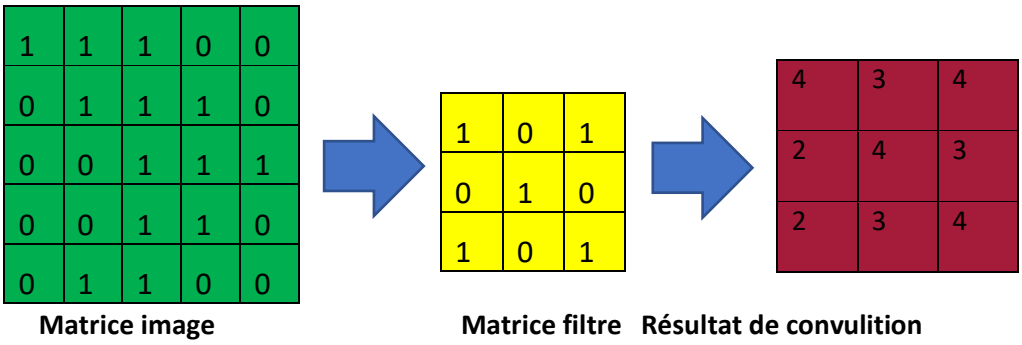

Figure 11: Principe de la couche de convulition

2.6.2La couche pooling

Ce type de couche est généralement placé entre deux couches convolutives : il reçoit plusieurs cartes d'entités en entrée et applique une opération de regroupement à chacune d'elles. Les opérations de mutualisation consistent à réduire la taille des images tout en préservant leurs caractéristiques importantes. Pour ce faire, divisez l'image en cellules régulières et conservez la valeur maximale dans chaque cellule.

L'approche utilisée consiste à imaginer une fenêtre de 2 ou 3 pixels glissant sur l'image, comme une convolution. Cependant, cette fois, nous avons 2 pas pour une fenêtre de taille 2 et 3 pas pour 3 pixels. La taille de la fenêtre est appelée "taille du noyau" et les étapes sont appelées « foulées » Pour chaque pas, on prend la valeur la plus élevée présente dans la fenêtre, qui constitue un nouveau pixel dans la nouvelle image. C'est ce qu'on appelle la mise en commun maximale.

La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2x2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de « Max-Pool 2x2 ».

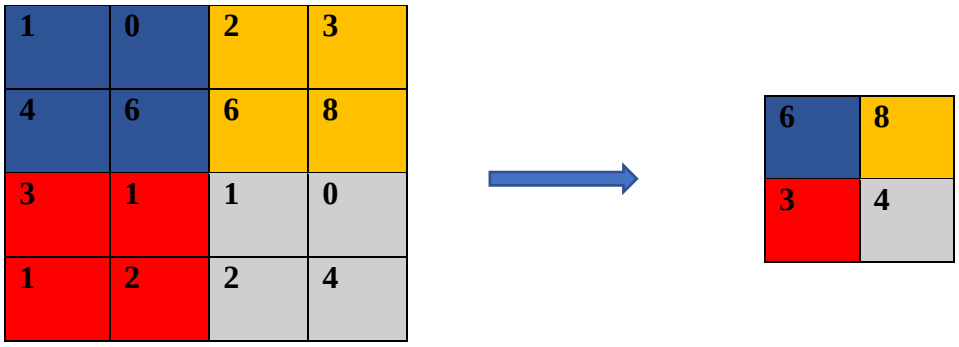

Figure 12: : Pooling avec un filtre 2x2 et un pas de 2

Il existe plusieurs types de Pooling :

- Le « max Pooling » qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image.

- Le « meanPooling » (ou averagePooling), soit la moyenne des pixels de la sélection : on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels.

- Le « sumPooling » c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme).

2.6.4 La couche entièrement connectée

La couche entièrement connectée en anglais « fully-connected » placé toujours à la fin d'architecture d'un réseau de neurones CNN. Ce type decouche reçoit un vecteur en entrée et produitunnouveauvecteurensortie.Pourcela,elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

La dernière couche fully-connected permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille N, où N est le nombre de classes dans notre problème declassificationd'images.Chaqueélément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe.

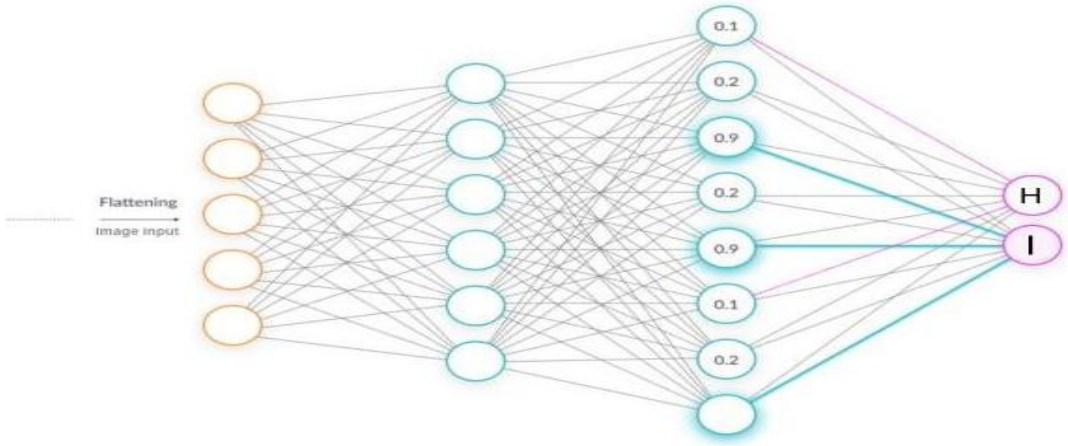

Figure 13 : Architecture d'un réseau neuronal a la phase 'fully-connected'.[30]

2.6.5 Fonction de perte (LOSS)

La couche de perte spécifie comment l'entrainement du réseau pénalise l'écart entre le signal prévu et réel. La fonction de perte E (W) est définie sur l'espace de poids. L'objectif est de rechercher le vecteur de poids W qui peut minimiser E (W). Limitation : Aucune méthode efficace ne peut résoudre l'extremum en mathématiques sur la surface complexe de haute dimension.

$$E(w) = \frac{1}{2}\sum_{(d\in D)}(t_d - o_d)^2 \qquad (4)$$

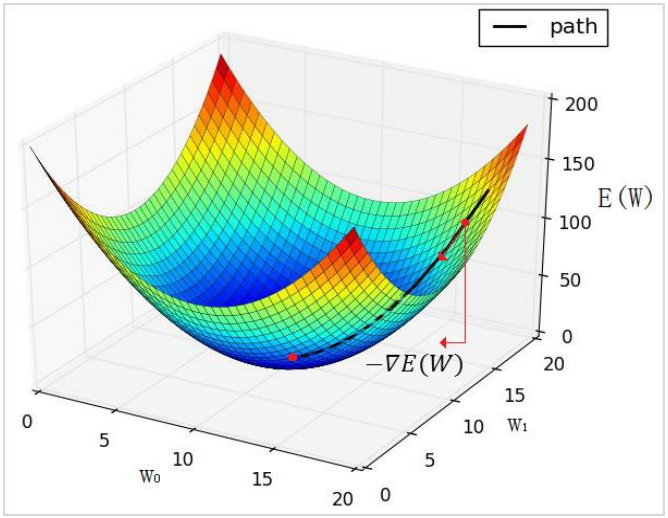

Figure 14: la fonction de perte [31]

2.7. Les hyperparamètres

Les hyperparamètres sont des paramètres dont les valeurs contrôlent le processus d'apprentissage qui sont défini avant le début du processus d'apprentissage. Ces paramètres sont réglables et peuvent affecter directement la qualité de l'apprentissage d'un modèle. Quelques exemples d'hyperparamètres dans l'apprentissage automatique et profond : comme dans la couche de convolution possède quatre hyperparamètres: Le nombre de filtres, La taille des filtres, le pas avec lequel on fait glisser la fenêtre correspondant au filtre sur l'image. Il y a aussi : Pooling size, Batch size, le personnage de division de la base de données à entrainement, validation, et le test « Train-test split ratio ». La valeur de drop-out dans le réseau neurones, Learning rate et epoch.

2.7.1. La taille du lot (Batch size)

La taille du lot définit le nombre d'échantillons qui seront propagés dans le réseau. Ces échantillons sont utilisés dans une époque « epoch » pour l'entrainement d'un réseau neuronal.Par exemple, disons que vous disposez de 1000 échantillons d'entraînement et que vous souhaitez définir une taille de lot égale à 100. L'algorithme prend les 100 premiers échantillons (du 1er au 100e) de l'ensemble de données d'apprentissage et entraîne le réseau. Ensuite, il prend les 100 autres échantillons (du 101e au 200e) et entraîne à nouveau le réseau. Nous pouvons continuer cette procédure jusqu'à ce que nous ayons propagé tous les échantillons à travers le réseau. La taille du lot affecte certains indicateurs tels que le temps d'apprentissage global, le temps d'apprentissage par époque, la qualité du modèle, etc.

Habituellement, nous choisissons la taille du lot comprise entre 16 et 512. Mais généralement, la taille de 32 est une règle empirique et un bon choix initial.

2.7.2. Le taux d'apprentissage (Learning rate)

Le taux d'apprentissage est un terme que nous utilisons dans l'apprentissage automatique et les statistiques. En termes simples, cela fait référence à la vitesse à laquelle un algorithme converge vers une solution. Le taux d'apprentissage est l'un des hyperparamètres les plus importants pour l'entraînement des réseaux de neurones. Par conséquent, il est important de définir une valeur aussi proche que possible de la valeur optimale.En général, la formation de modèles de réseaux de

neurones nécessite l'utilisation de techniques d'optimisation basées sur des algorithmes de descente de gradient. Après calcul du gradient de la fonction de perte par rapport aux poids, ce gradient a une direction vers l'optimum local.

Nous utilisons l'hyperparamètre du taux d'apprentissage pour ajuster les poids dans cette direction et optimiser le modèle. Le taux d'apprentissage représente la taille du pas de descente du gradient vers l'optimum localComme la figure 15 montre :

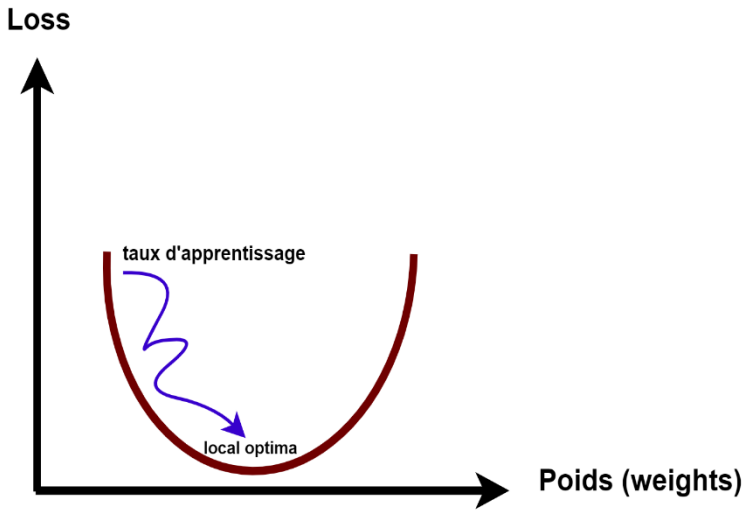

Figure 15 : la relation entre le taux d'apprentissage et le poids et la perte

2.7.3. Epoche

On parle d'une époque lorsqu'un ensemble de données ENTIER est passé en avant et en arrière c'est à dire fait un aller et routeur par le réseau neuronal une seule fois.Nous utilisons plus d'une époque en passant l'ensemble de données via un réseau de neurones ne suffit pas et nous devons transmettre l'ensemble de données complet plusieurs fois au même réseau de neurones, donc c'est pour ça on utilise plus d'une époque.

2.8. Les modèles de réseau neuronal convolutif

Dans le réseau neural convolutif on trouve nombreux des modelés Comme :

LeNet-5 (1998), AlexNet(2012), VGG-16(2014), Inception-v1(2014), Inception-v3(2015), ResNet-50(2015),Xception(2016), Inception-v4(2016), Inception-ResNet-v2(2016) et le dernier modele ResNeXT-50(2017).

Dans notre travail, nous avons utilisé les deux réseaux de neurones profonds les plus couramment utilisés, à savoir VGG16 et AlexNet.

2.8.1 VGG-16

VGG16 est une architecture de réseau neuronal à convolution (CNN) qui a été utilisée pour remporter le concours ILSVR(Imagenet) en 2014. Il est considéré comme l'une des excellentes architectures de modèles de vision à ce jour. La chose unique à propos de VGG16 est qu'au lieu d'avoir un grand nombre d'hyper-paramètres, ils se sont concentrés sur le fait d'avoir des couches de convolution de filtre 3x3 avec un stride 1 et ont toujours utilisé le même padding et la couche maxpool de filtre 2x2 de stride 2. Cette disposition des couches de convolution et de maxpool est suivie de manière cohérente dans toute l'architecture. A la fin, il a 2 FC (fullyconnectedlayers) suivis d'un softmax pour la sortie. Le 16 dans VGG16 fait référence au fait qu'il a 16 couches qui ont des poids. Ce réseau est assez vaste et compte environ 138 millions de paramètres (approximativement). [32]

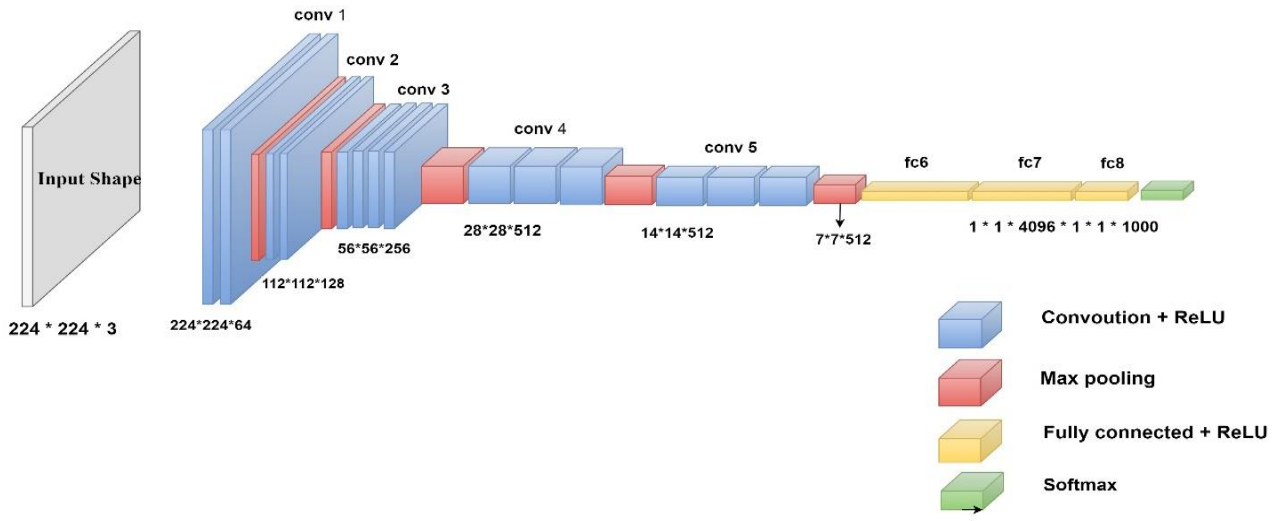

Figure 16: : architecture de VGG-16

2.8.2 AlexNet

AlexNet est le nom de l'architecture Convolutional Neural Network (CNN), conçue par Alex Krizhevsky en collaboration avec Ilya Sutskever et Geoffrey Hinton, AlexNet est considéré comme l'un des articles les plus influents sur la vision par ordinateur et a incité de nombreux autres à utiliser les CNN et les GPU Deep Learning 2021, selon Google Scholar, l'article d'AlexNet sera cité plus de 80 000 fois.

AlexNet se compose de 8 huit couches ; les cinq premières sont des couches convolutives, certaines sont des couches de regroupement maximum et les trois dernières sont des couches entièrement connectées.Il utilise une fonction d'activation non saturante ReLU, qui montre de meilleures performances d'apprentissage que tanh et sigmoïde.

En 2015, AlexNet a été dépassé par le CNN ultra-profond de Microsoft Research Asia avec plus de 100 couches et a remporté le concours ImageNet 2015. [33]

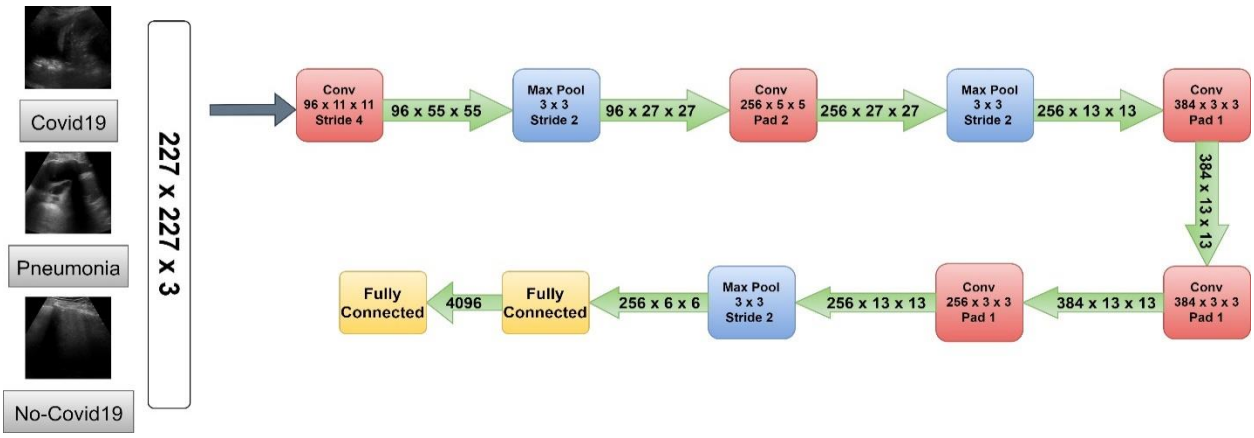

Figure 17: Architecture de AlexNet

Tableau 1: Les détails de configuration de l'architecture AlexNet

Layer	# Filtres / neurons	Filter size	Stride	Padding	Size of featuremap	Activation function
Input	-	-	-	-	227 x 277 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU

2.9. Conclusion

Dans ce chapitre nous avons commencé par définir le neurone artificiel et citer trois fonctions qui l'active, ensuite on a expliqué le réseau de neurone ainsi que son fonctionnement, en passant par son architecture, Aussi qu'une vision générale sur l'apprentissage profond, toute en présentant la méthode choisie dans notre travail de recherche qui est le CNN. Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour notre système.

Chapitre 3 : Résultats et interprétations

3.1. Introduction

Pour évaluer l'efficacité de notre méthode proposée, les expériences décrites dans ce travail sont divisées en deux parties principales, et chacune de ces parties se compose d'architectures VGG16 et AlexNet. Dans ce travail, un algorithme d'optimisation nommé Adaptive Moment Estimation (Adam) a été examiné pour mettre à jour les poids du réseau pendant l'entrainement des modèles VGG16 et AlexNet. Les paramètres expérimentaux pour la modélisation du réseau sont répertoriés ci-dessous dans le tableau2.

Table **2** : Hyperparamètres utilisés pour le modèle vgg16 et AlexNet formé

Settings. No	Hyperparameters	Valeurs
1	Learning rate	[0.1,0.01,0.001]
2	Numbers of Epochs	[12, 22]
3	Batch Size	[32, 64, 128]
4	Optimizer	Adam

Notre Environnement de développement inclut les dispositifs suivants

Tableau 2: Dispositifs matérielles

Matériel Marque	packardbell .
Taille de Stockage	1TB HDD
Mémoire (RAM)	6 Go.
Processeur	AMD E2-3800 APU with Radeon(TM) HD Graphics 1.30 GHz.
Système d'exploitation	Windows 10, X64.

3.2. Base de données

L’ensemble de données d'échographie pulmonaire (POCUS) a été utilisé dans notre étude.(https://github.com/jannisborn/covid19_pocus_ultrasound). La base de données est mise à jour régulièrement. Les données de cette base de données sont divisées en deux catégories : convexe et liner. Seules les données convexes ont été utilisées dans notre étdue. . Cette base de données se compose d’un ensemble d’images et vidéos. Il y a 122 vidéos au total, 30 pour COVID-19, 36 pour la pneumonie et 56 pour le Non-Covid19. Il y a également 29 photos, dont 5 pour COVID-19, 14 pour une pneumonie bactérienne 10 pour le Non-Covid19. L'ensemble de données POCUS a été utilisé dans plusieurs travaux scientifiques. Dans le cas des vidéos, un maximum de 30 images par seconde ont été sélectionné pour chaque vidéo.La figure 18 représente les trois classes utilisées dans cette notre étude, à savoir COVID-19, Pneumonie et Non-COVID-19.

(a) COVID-19 (b) Pneumonia (c) Non-COVID-19

Figure 18 : les trois classes utilisées dans cette étude, à savoir COVID-19 (a), Pneumonie (b) et Non-COVID-19 (c)

Nous avons utilisé le code source ci-dessous(Figure 19) pour obtenir des images de chaque vidéo.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 14 01:55:46 2022
@author: Bou-R
"""
import cv2
import os
count = 0
for files in os.listdir(r"C:\Users\Bou-R\Desktop\fps test"):
```

cap = cv2.videocapture(os.path.join(r'C:\Users\Bou-R\Desktop\fps test', files))

success, image = cap.read()

while success:

success, image = cap.read()

if not success:

break

cv2.imwrite(r"C:\Users\Bou-R\Desktop\outpot" + str(count) + '.jpg' , image")

if cv2.waitkey(10)==27:

break

count +=1

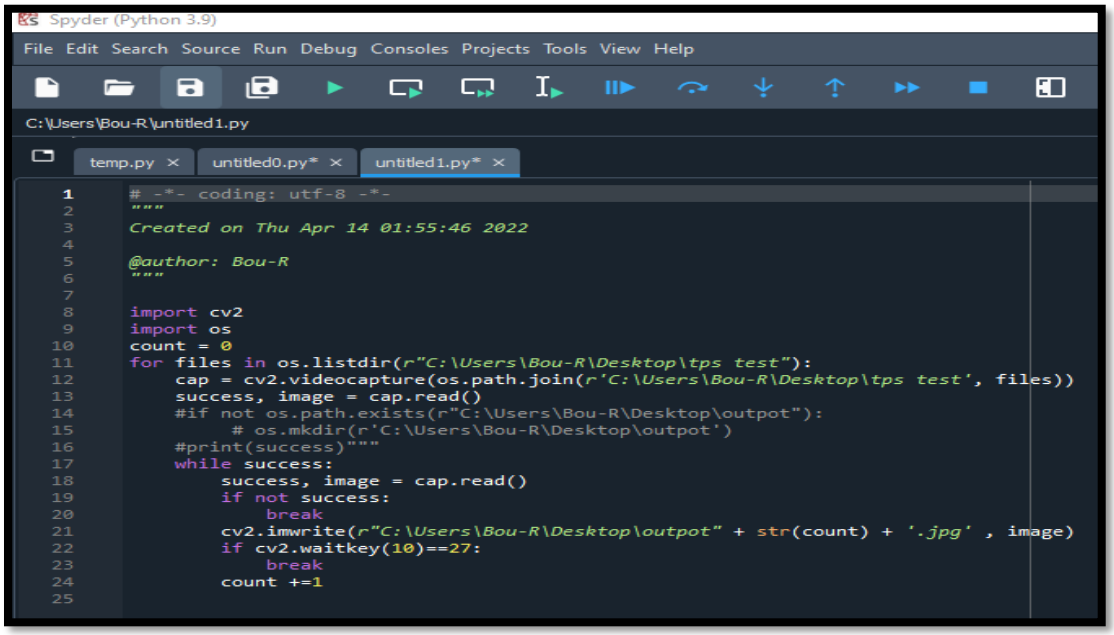

Figure 19 : Code permettent l'extraction des images à partir d'un video

3.2.1. Méthode utilisée pour diviser la base de données

Il existe trois méthodes différentes couramment utilisées lorsque les réseaux de neurones profonds sont utilisés pour la classification, nous pouvons citer ces méthodes comme suit :

3.2.1.1. Train/Test

Dans cette méthode, les données seront divisées en deux parties principales, la première concerne les données d'entraînement qui prennent presque la grande partie des échantillons et l'autre partie concerne les données de test utilisées pour la prédiction.

3.2.1.2. Train/Val/Test

Dans cette méthode, les données seront divisées en trois parties principales, la première concerne les données de formation qui prennent également presque la grande partie des échantillons et la seconde partie concerne les données de validation utilisées pour valider les performances du modèle d'apprentissage en profondeur, et la dernière utilisé pour la prédiction

3.2.1.3. K-Fold CrossValidation

La validation croisée est une procédure de rééchantillonnage utilisée pour évaluer les modèles d'apprentissage automatique sur un échantillon de données limité.La procédure a un seul paramètre appelé k qui fait référence au nombre de groupes dans lesquels un échantillon de données donné doit être divisé. En tant que telle, la procédure est souvent appelée validation croisée k-fold. Lorsqu'une valeur spécifique pour k est choisie, elle peut être utilisée à la place de k dans la référence au modèle, telle que k = 10 devenant une validation croisée de 10 fois.La validation croisée est principalement utilisée dans l'apprentissage automatique appliqué pour estimer la compétence d'un modèle d'apprentissage automatique sur des données invisibles. Autrement dit, utiliser un échantillon limité afin d'estimer comment le modèle devrait fonctionner en général lorsqu'il est utilisé pour faire des prédictions sur des données non utilisées lors de la formation du modèle.

Dans notre cas, nous avons utilisé la deuxième méthode (**Train/Val/Test**)et le code source utilisé pour cela est basé sur python. La figure 20 montre le code source de la méthode sélectionnée :

```
# -*- coding: utf-8 -*-
"""
Created on Sun Jun  5 14:13:52 2022
@author: Bou-R
"""
import splitfolders
Dataset= (r'C:\Users\Bou-R\Desktop\Data\')
splitfolders.ratio( Dataset , 'C:/Users/Bou-R/Desktop/after split/' , seed=1337 , ratio=(0.7,0.2,0.1))
```

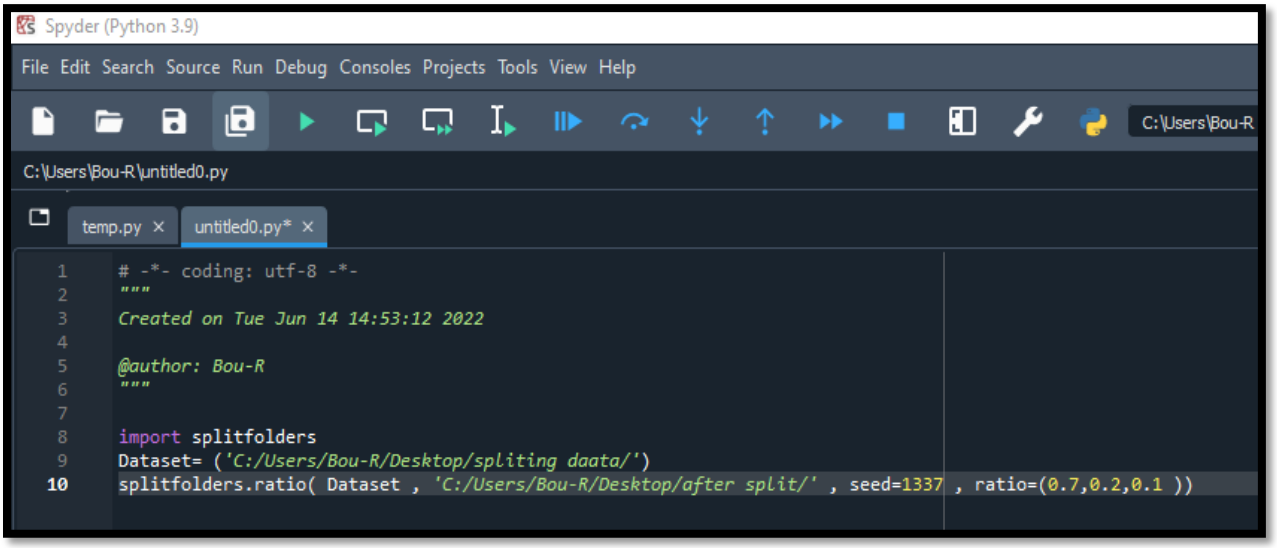

Figure 20: Le code source de la méthode (Train/Val/Test)

3.3. Langage de programmation utilisé

3.3.1. Python

Python est un langage de programmation de haut niveau populaire pour les applications à usage général Il a été initialement conçu par Guido van Rossum en 1991, et sa mise en œuvre a été lancée et développé par Python Software Foundation en décembre 1989. Python est le langage de programmation open source le plus utilisé par les informaticiens. Ce langage de programmation excelle dans les domaines de la gestion d'infrastructure, l'analyse de données et du développement de logiciels. Python, en fait, permet aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la façon dont ils le font, grâce à ses fonctionnalités. Il libère les programmeurs des contraintes des anciens langages de codage qui étaient alors obsolètes. Par conséquent, écrire du code en Python est plus rapide que d'écrire du code dans d'autres langages.

3.3.2. Plate-forme informatique utilisée (Google Colab)

Pour gérer tous les essais de l'entrainment et de test dans ce travail, On a utilisé Google Collaboratory.Cette plate-forme est un environnement de bloc-notes Jupyter, qui est open-source et permet aux utilisateurs d'exécuter et de partager du code source en utilisant Python comme langage de programmation. C'est un environnement particulièrement adapté au machine learning,

à l'analyse de données et à l'éducation. L'utilisation de collab nous a permis d'effectuer des calculs intensifs, en particulier dans les applications de réseaux de neurones profonds, à l'aide d'un puissant matériel, avec les GPU appropriés et les unités de traitement de tenseur (TPU), ce qui a entraîné environ 12 h de la formation, ce qui a entraîné un temps de formation d'environ 12 heures. Plus important encore, la plate-forme de collaboration pour les applications d'apprentissage en profondeur fournit les ressources de traitement nécessaires pour effectuer une énorme catégorisation de données gratuitement, sur n'importe quel ordinateur, à tout moment, et elle distribue l'enseignement et la recherche en apprentissage automatique, ainsi qu'une carte graphique NVIDIA Tesla T4 de 12 Go. Nous pouvons enregistrer une copie de notre session d'apprentissage ainsi que télécharger les données depuis Google Drive. Keras et TensorFlow ont été utilisés dans cette expérience pour construire les architectures VGG16 et AlexNet basées sur cette plateforme.

3.4. Logiciel utilisé

3.4.1. Anaconda

Anaconda est une distribution de langage de programmation Python et R gratuite et open source pour le développementd'applications de science des données et d'apprentissage automatique, l'intelligence artificielle, dans le but de faciliter la gestion et le déploiement des packages, car il contient plusieurs packages (Il contient plus de 1500 packages de science des données) nécessaires dans ce domaine notamment Python, Condapromt , Jupyter, Spyder ... etc, Il a intégré de nombreuses bibliothèques tierces très utiles. L'installation d'Anaconda équivaut à l'installation automatique de Python et de certaines bibliothèques couramment utilisées telles que Numpy, Pandas, Tensorflow, et Matplotlib, ce qui rend l'installation beaucoup plus facile que l'installation standard de Python. Et comme le langage Python, il est multiplateforme.Ce logiciel est indispensable pour tous les développeurs dans le domaine de la science des données. Grâce aux outils qu'il fournit, il permet de collecter et de transformer des données à grande échelle.

3.4.2. Spyder

Spyder est un puissant environnement scientifique écrit en Python pour Python,il combine de manière unique les capacités avancées d'édition, d'analyse, de débogage et de profilage d'outils de développement complets avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les capacités de visualisation des progiciels scientifiques, aussi Il intègre un certain

nombre de bibliothèques scientifiques, notamment Matplotlib, NumPy,Keras, SciPy et Tensoreflow, il est gratuit et multiplateforme.Spyder est livré avec une bibliothèque plus grande que vous téléchargez lorsque vous installez le programme avec Anaconda.

3.5. Bibliothèques Python utilisées

3.5.1. TensorFlow

C'est une bibliothèque open source développée par l'équipe Google Brain en 2012.La première de la liste des bibliothèques Python pour la science des données est TensorFlow. TensorFlow est une bibliothèque de calculs numériques haute performance avec plus de 35 000 commentaires et une communauté dynamique de plus de 1 500 contributeurs. Il est utilisé dans une variété de domaines scientifiques.TensorFlow est essentiellement un cadre pour définir et exécuter des calculs qui impliquent des tenseurs, qui sont des objets de calcul partiellement définis qui produisent finalement une valeur.L'une des meilleures fonctionnalités de TensorFlow est la simplicité d'écriture du code. Les API permettent aux utilisateurs de gagner du temps en évitant d'avoir à réécrire une partie du code qui autrement prendrait du temps. TensorFlow accélère la formation d'un modèle. De plus, la probabilité d'erreurs logicielles est minimisée, souvent de 55 à 85 %.En général, une partie considérable du calcul est consacrée à la formation du modèle. De plus, la procédure de formation est répétée plusieurs fois pour résoudre les difficultés qui peuvent se développer. En raison de l'augmentation de la consommation d'énergie, vous devrez utiliser l'informatique distribuée. TensorFlow simplifie le traitement de quantités massives de données en exécutant le code de manière distribuée.

3.5.2. NumPy

NumPy, appelé aussi Python numérique. Il s'agit d'une bibliothèque Python open source pour les calculs mathématiques et numériques, ainsi que pour la programmation scientifique, d'ingénierie et de science des données.NumPy est le package nécessaire pour effectuer des calculs mathématiques et statistiques en Python. Il s'agit également d'un package de traitement de tableau à usage général, comprenant des objets multidimensionnels hautes performances appelés tableaux et des outils pour interagir avec eux. Il est particulièrement adapté aux tableaux multidimensionnels et aux multiplications matricielles. NumPy est généralement importé sous l'alias np.

3.5.3. Keras

Keras est une interface de programmation d'application (API) de réseau neuronal efficace de haut niveau écrite en Python. Keras a été adopté et intégré à TensorFlow. Cette bibliothèque de réseaux de neurones open source est conçue pour fournir une expérimentation rapide avec des réseaux de neurones profonds et peut s'exécuter sur TensorFlow et Theano, c'est donc une bonne option si vous ne voulez pas plonger dans les détails de TensorFlow. Keras se concentre sur la modularité, la convivialité et l'évolutivité. Il ne gère pas les calculs de bas niveau au lieu de cela, il les transfère à une autre bibliothèque appelée Backend.L'une des caractéristiques les plus importantes de cette bibliothèque est qu'elle fournit un grand nombre d'ensembles de données préformatés qui peuvent être utilisés pour importer et charger directement des données. Il contient également un certain nombre de couches et de paramètres intégrés qui peuvent être utilisés pour la construction, la configuration, la formation et l'évaluation des réseaux de neurones.

3.5.4. Splitfolders

Split-folders est une bibliothèque Python couramment utilisée dans les applications d'intelligence artificielle, d'apprentissage automatique et d'apprentissage en profondeur. Split-folders n'a aucun bogue, aucun bogue, il a une licence permissive et son support est faible. Cependant, les fichiers de construction de dossiers divisés ne sont pas disponibles. Vous pouvez l'installer avec la commande "pipinstall split-folders" in annaconda prompt utilisant anaconda prompt ou le télécharger depuis GitHub, PyPI(the Python Package Index). Divisez le dossier contenant les fichiers (tels que les images) en dossiers d'apprentissage, de validation et de test (ensemble de données), cette bibliothèque contient deux class splitfolders.ratio qui divise par personnage y a aussi splitfolders.fixed qui fonction pour un chiffre fixe.

3.5.5. scikit-learn

Scikit-learndeja connu sous le nom de sklearn, est la principale bibliothèque d'outils dédiés au machine learning et à la data-science dans l'univers Python. Il propose divers algorithmes de classification, de régression et de regroupement, notamment les machines vectorielles, les forêts aléatoires « randomforest », le boosting de gradient et les k-means.

Scikit-learn s'installe comme la majorité des paquets Python :pip3 installscikit-learn.

Sklearnest conçu pour interagir avec les bibliothèques scientifiqueset numériques SciPy Python et NumPy.

3.6. Critères d'évaluation des performances

Dans cette étude, les résultats expérimentaux ont été évalués à l'aide de quatre paramètres - précision globale (accuracy), précision, sensibilité (recall) et score F1. La précision globale mesure le nombre d'anomalies normales et anormales correctement classées.Échantillons correspondant à tous les échantillons d'essai. La précision est définie comme la valeur prédite positive (VPP) qui fournitles résultats pertinents pour une classification précise. La sensibilité sont définis comme des segments classifiés vrais positifs,divisé par le nombre total de segments positifs. Le score F1 est basé sur la moyenne harmonique de la spécificitéet sensibilité, et les définitions sont données en (5), (6), (7) et (8) respectivement comme suit :

$$\text{précision globale} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{précision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{sensibilité (recall)} = \frac{TP}{TP + FN} \quad (7)$$

$$F - \text{score} = 2 * \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (8)$$

Où :TP : signifie vrai positif ; TN : signifie vrai négatif ; FP : signifie faux positif ;FN : signifie faux négatif.

3.7. Comparer et interpréter les résultats expérimentaux

Nous avons divisé nos expériences en deux parties principales et nous pouvons les décrire comme suit :

3.7.1. Partie 1 : Performance d'un system du Classification des images échographiques COVID-19 à l'aide de AlexNet

Tout d'abord, nous commençons à exécuter le modèle d'apprentissage en profondeur AlexNet avec des valeurs initiales d'hyparparamètres qui incluent la taille du lot et le taux d'apprentissage, pour classer les trois classes à savoir COVID-19, Non-COVID-19 et Pneumonie.

Étape 1

Nous avons fixé l'époque en 22 (dans chaque époque, il y a 112 itérations pour la formation et le reste pour la validation) comme valeur et nous exécutons le système d'apprentissage dans google colab. La figure 21 montre l'historique des étapes de training et de validation. Comme nous pouvons le voir, à l'époque finale égale à 22, la perte de training est de 1,7893 et la précision de training est de 0,3399, ainsi que, pour l'étape de validation, nous avons obtenu 1,1410 comme perte de validation et 0,2007 comme précision de validation, ces résultats obtenus lorsque le taux d'apprentissage est fixé sur 0,1 et la taille du lot sur 32. Nous exécutons à nouveau le même processus mais au lieu d'utiliser 32 comme valeur de la taille du lot, nous l'avons changé en 64 pour voir l'effet de la taille du lot dans notre application. À partir de la figure 22, nous pouvons observer que la première matrice de confusion (a) donnait comme point de vue que les échantillons du covid et du non covid étaient presque meilleurs que la confusion (b) et (c). Par conséquent, nous pouvons sélectionner la taille de lot de 32 comme valeur prédominante dans le reste de nos expériences.

```
Epoch 21/22
224/224 [==============================] - ETA: 0s - loss: 1.7873 - accuracy: 0.3346
Epoch 21: val_accuracy did not improve from 0.52295
224/224 [==============================] - 1020s 5s/step - loss: 1.7873 - accuracy: 0.3346 - val_loss: 1.0639 - val_accuracy: 0.5229
Epoch 22/22
224/224 [==============================] - ETA: 0s - loss: 1.7893 - accuracy: 0.3399
Epoch 22: val_accuracy did not improve from 0.52295
224/224 [==============================] - 1018s 5s/step - loss: 1.7893 - accuracy: 0.3399 - val_loss: 1.1410 - val_accuracy: 0.2007
```

(a) Epoch = 22; Batch = 32; Learning rate = 0.1

```
Epoch 21/22
112/112 [==============================] - ETA: 0s - loss: 10.9444 - accuracy: 0.3414
Epoch 21: val_accuracy did not improve from 0.54590
112/112 [==============================] - 30s 270ms/step - loss: 10.9444 - accuracy: 0.3414 - val_loss: 3.0552 - val_accuracy: 0.5283
Epoch 22/22
112/112 [==============================] - ETA: 0s - loss: 11.5918 - accuracy: 0.3298
Epoch 22: val_accuracy did not improve from 0.54590
112/112 [==============================] - 30s 267ms/step - loss: 11.5918 - accuracy: 0.3298 - val_loss: 1.8547 - val_accuracy: 0.5205
```

(b) Epoch = 22; Batch = 64; Learning rate = 0.1

```
Epoch 21/22
56/56 [==============================] - ETA: 0s - loss: 8.5148 - accuracy: 0.3298
Epoch 21: val_accuracy did not improve from 0.56055
56/56 [==============================] - 15s 277ms/step - loss: 8.5148 - accuracy: 0.3298 - val_loss: 8.8632 - val_accuracy: 0.2852
Epoch 22/22
56/56 [==============================] - ETA: 0s - loss: 6.0948 - accuracy: 0.3426
Epoch 22: val_accuracy did not improve from 0.56055
56/56 [==============================] - 15s 271ms/step - loss: 6.0948 - accuracy: 0.3426 - val_loss: 2.3310 - val_accuracy: 0.2637
```

(c) Epoch = 22; Batch = 128; Learning rate = 0.1

Figure 21: l'historique des étapes de training et de validation

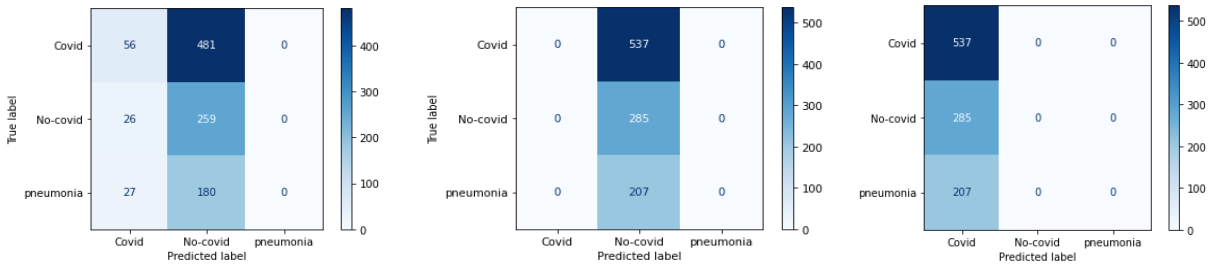

Figure 22: Trois matrices de confusion du modèle AlexNet utilisant trois tailles de lots différentes avec des taux d'apprentissage et des époques fixes.

Tableau 3: Rapport de classification d'AlexNet avec une taille de lot de 32 et un taux d'apprentissage de 0,1

	Précision	Sensibilité	F1-score
Covid	0.51	0.10	0.17
No-covid	0.28	0.91	0.43
Pneumonia	0.00	0.00	0.00

Tableau 4: Rapport de classification d'AlexNet avec une taille de lot de 64 et un taux d'apprentissage de 0,1

	Précision	Sensibilité	F1-score
Covid	0.52	1.00	0.69
No-covid	0.00	0.00	0.00
Pneumonia	0.00	0.00	0.00

Tableau 5: Rapport de classification d'AlexNet avec une taille de lot de 128 et un taux d'apprentissage de 0,1

	Précision	Sensibilité	F1-score
Covid	0.00	0.00	0.00
No-covid	0.28	1.00	0.43
Pneumonia	0.00	0.00	0.00

Étape 2

Nous avons fixé l'époque en 22 et nous exécutons le système d'apprentissage. La figure23 montre l'historique des étapes de formation et de validation.

```
Epoch 21/22
224/224 [==============================] - ETA: 0s - loss: 1.7223 - accuracy: 0.4264
Epoch 21: val_accuracy did not improve from 0.52246
224/224 [==============================] - 58s 259ms/step - loss: 1.7223 - accuracy: 0.4264 - val_loss: 1.1060 - val_accuracy: 0.2773
Epoch 22/22
224/224 [==============================] - ETA: 0s - loss: 1.7227 - accuracy: 0.3160
Epoch 22: val_accuracy did not improve from 0.52246
224/224 [==============================] - 58s 260ms/step - loss: 1.7227 - accuracy: 0.3160 - val_loss: 1.0970 - val_accuracy: 0.5225
```

Figure 23: historique des étapes de formation et de validation d'AlexNet (taille du lot = 32 ; taux d'apprentissage = 0,01)

Nous exécutons à nouveau le même processus mais au lieu d'utiliser 0.1 comme valeur de taux d'apprentissage, nous l'avons changé en 0.01 pour voir l'effet de taux d'apprentissage dans notre application.

Comme nous pouvons le voir, à l'époque finale égale à 22, la perte de training est de 1.7227 et la précision de training est 0,3160, ainsi que, pour l'étape de validation, nous avons obtenu 1.0970 comme perte de validation et 0.5225 comme précision de validation, ces résultats obtenus lorsque le taux d'apprentissage est changé en 0,01 et la taille du lot fixé sur 32.

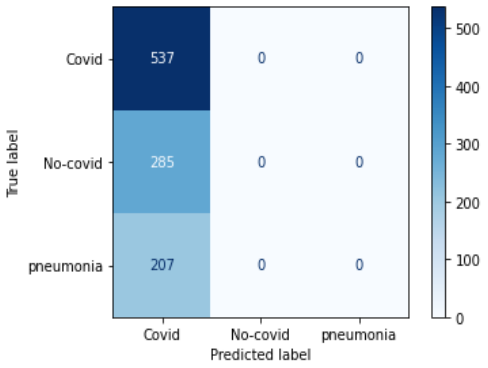

Figure 24: Matrice de confusion d'AlexNet dans le lot 32 et taux d'apprentissage 0,01

Tableau 6: Rapport de classification d'AlexNet avec une taille de lot de 32 et un taux d'apprentissage de 0,01

	Précision	Sensibilité	F1-score
Covid	0.52	1.00	0.69
No-covid	0.00	0.00	0.00
Pneumonia	0.00	0.00	0.00

À partir de la figure 23, comme nous pouvons le voir lorsque nous utilisons un taux d'apprentissage égal à 0,01, nous obtenons des résultats similaires à ceux que nous avons obtenus dans la première étape de cette section. On peut donc passer à la valeur suivante du taux d'apprentissage qui est 0,001.

Étape 3

Nous avons aussi fixé l'époque en 22 comme valeur et nous exécutons le système d'apprentissage toujours dans google colab.

La figure 25 montre l'historique des étapes de training et de validation. Comme nous pouvons le voir, à l'époque finale égale à 22, la perte de training est de 0.0740 et la précision de training est de 0.9854, ainsi que, pour l'étape de validation, nous avons obtenu 0.0100 comme perte de validation et 0.9951 comme précision de validation, ces résultats obtenus lorsque le taux d'apprentissage est fixé sur 0,001 et la taille du lot sur 32.

```
Epoch 21/22
224/224 [==============================] - ETA: 0s - loss: 0.2448 - accuracy: 0.9790
Epoch 21: val_accuracy did not improve from 0.99854
224/224 [==============================] - 58s 260ms/step - loss: 0.2448 - accuracy: 0.9790 - val_loss: 0.1729 - val_accuracy: 0.9424
Epoch 22/22
224/224 [==============================] - ETA: 0s - loss: 0.0740 - accuracy: 0.9854
Epoch 22: val_accuracy did not improve from 0.99854
224/224 [==============================] - 58s 259ms/step - loss: 0.0740 - accuracy: 0.9854 - val_loss: 0.0100 - val_accuracy: 0.9951
```

Figure 25: historique des étapes de formation et de validation d'AlexNet (taille du lot = 32 ; taux d'apprentissage = 0,001)

Nous exécutons à nouveau le même processusqui a conduit à l'étape 3 mais au lieu d'utiliser 0.01 comme valeur de taux d'apprentissage, nous l'avons changé en 0.001 pour voir l'effet de taux d'apprentissagedans le problème de classification d'images.

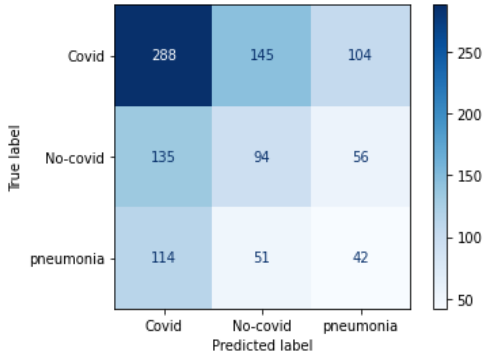

Figure 26: Matrice de confusion d'AlexNet dans le lot 32 et taux d'apprentissage 0,01

Tableau 7: Rapport de classification d'AlexNet avec une taille de lot de 32 et un taux d'apprentissage de 0,001

	Précision	Sensibilité	F1-score
Covid	0.54	0.54	0.54
No-covid	0.32	0.33	0.33
Pneumonia	0.21	0.20	0.21

À partir de la figure 25 et du tableau 8, nous pouvons observer que la précision, le rappel et le score F1 ne subissent aucune dégradation sévère lorsque le taux d'apprentissage de 0,001 est utilisé. Les valeurs globales de précision, de rappel et de score F1 dans les expériences fournissent des preuves préliminaires que le taux d'apprentissage de 0,001 a la capacité de faire en sorte que le réseau AlexNet discrimine un problème multiclasse. Par conséquent, on peut en déduire que la classification des affections pulmonaires est possible en utilisant l'AlexNet proposé avec un taux d'apprentissage de 0,001 en conjonction avec les images échographiques. Ce système donne une précision, un rappel et un score f1 maximum de 54%. L'autre défi principal de notre étude qui s'appelle **UNBALANCED DATASET**, quand Les données déséquilibrées font référence aux types d'ensembles de données où la classe cible a une distribution inégale des observations, c'est-à-dire qu'une étiquette de classe a un nombre très élevé d'observations et l'autre a un nombre très faible d'observations.

3.7.1. Partie 2 : Performance d'un system du Classification des images échographiques COVID-19 à l'aide de VGG16

Etape 1

Pour voir l'effet de notre étude comparative sur une architecture CNN différente, nous avons mis en œuvre les mêmes expériences que précédemment sur un autre réseau d'apprentissage en profondeur plus complexe - VGG16. Les résultats sont montrés est comme suit :

```
Epoch 21/22
224/224 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5226WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
224/224 [==============================] - 105s 468ms/step - loss: nan - accuracy: 0.5226 - val_loss: nan - val_accuracy: 0.5176
Epoch 22/22
224/224 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5236WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
224/224 [==============================] - 105s 466ms/step - loss: nan - accuracy: 0.5236 - val_loss: nan - val_accuracy: 0.5176
```

```
Epoch 21/22
112/112 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5241WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
112/112 [==============================] - 54s 482ms/step - loss: nan - accuracy: 0.5241 - val_loss: nan - val_accuracy: 0.5342
Epoch 22/22
112/112 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5123WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
112/112 [==============================] - 55s 486ms/step - loss: nan - accuracy: 0.5123 - val_loss: nan - val_accuracy: 0.5273
```

```
Epoch 21/22
56/56 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5201WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
56/56 [==============================] - 27s 485ms/step - loss: nan - accuracy: 0.5201 - val_loss: nan - val_accuracy: 0.5469
Epoch 22/22
56/56 [==============================] - ETA: 0s - loss: nan - accuracy: 0.5084WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
56/56 [==============================] - 27s 484ms/step - loss: nan - accuracy: 0.5084 - val_loss: nan - val_accuracy: 0.5332
```

Figure 27: historique des étapes de formation et de validation de VGG16 (taille du lot = 32 ; taux d'apprentissage = 0,1)

On obtient la même confusion matrice

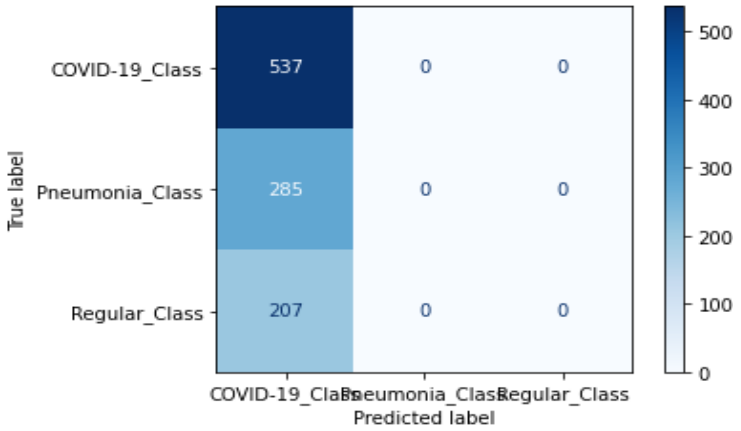

Figure 28: Matrice de confusion d'AlexNet dans le lot 32 et taux d'apprentissage 0,01

Tout d'abord, d'après les mesures de validation et les modèles entraînés avec de petites batch sizes généralisent bien sur la validation set. Nous validons que le taux d'apprentissage de 0,1 a donné les mêmes résultats que ceux votés à l'étape 1 avec AlexNet.

De plus, comme le modèle AlexNet lorsque nous avons fixé la taille du lot sur 32 et l'époque, nous pouvons observer les résultats comme indiqué dans la matrice de confusion

Etape 2

Maintenant, nous formons et validons nos réseaux VGG16 avec différents taux d'apprentissage pour voir l'effet comme nous l'avons fait dans AlexNet. L'historique des résultats du processus d'apprentissage est illustré dans la figure ci-dessous

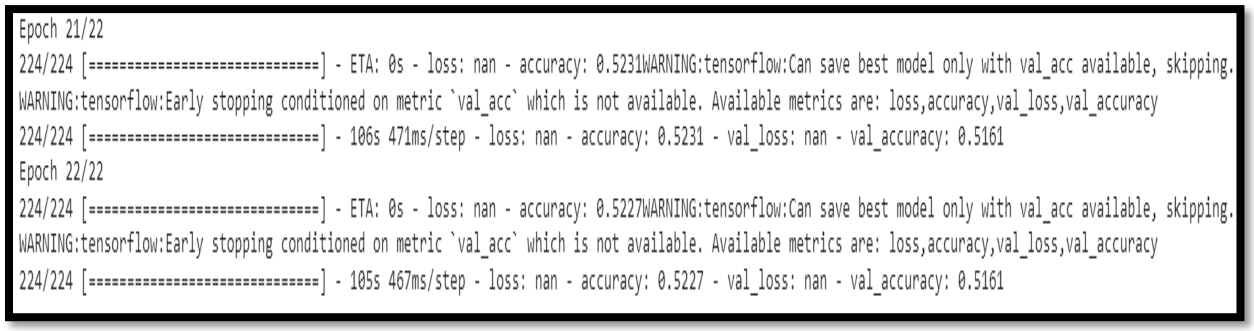

Figure 29: historique des étapes de formation et de validation de VGG16 (taille du lot = 32 ; taux d'apprentissage = 0,01)

Tableau 8: Rapport de classification VGG16 avec une taille de lot de 32 et un taux d'apprentissage de 0,01

	Précision	Recall	F1-score
Covid	0.52	1.00	0.69
No-covid	0.00	0.00	0.00
Pneumonia	0.00	0.00	0.00

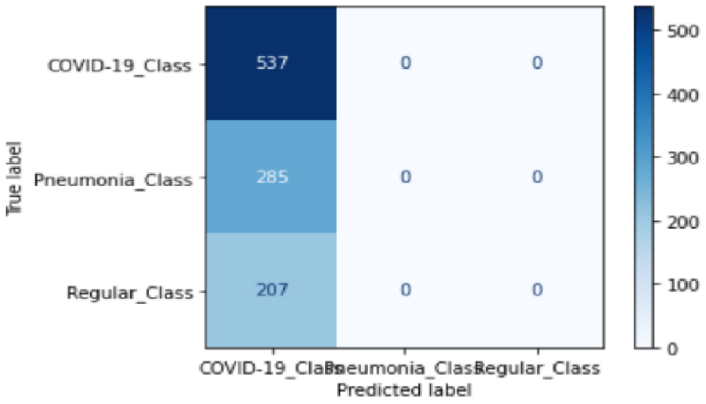

Figure 30: Matrice de confusion VGG16 dans le lot 32 et taux d'apprentissage 0,01

D'après les résultats présentés ci-dessus, nous pouvons dire que lorsque nous utilisons 0,1 ou 0,01 comme valeur pour le taux d'apprentissage et avec différents lots 32, 64,128 et époques fixes, nous avons obtenu de mauvais résultats sur la base du tableau et de la matrice de confusion.Ensuite, nous pouvons mener une autre expérience qui consiste à utiliser un 0,001 comme valeur du taux d'apprentissage comme nous l'avons fait avec AlexNet, et les résultats ont été mentionnés ci-dessous (Etape 3).

Etape 3

```
Epoch 21/22
224/224 [==============================] - ETA: 0s - loss: 3.4823e-08 - accuracy: 1.0000WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
224/224 [==============================] - 110s 490ms/step - loss: 3.4823e-08 - accuracy: 1.0000 - val_loss: 2.3624e-04 - val_accuracy: 1.0000
Epoch 22/22
224/224 [==============================] - ETA: 0s - loss: 2.9306e-08 - accuracy: 1.0000WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
224/224 [==============================] - 110s 490ms/step - loss: 2.9306e-08 - accuracy: 1.0000 - val_loss: 4.6720e-05 - val_accuracy: 1.0000
```

Figure 31: historique des étapes de formation et de validation de VGG16 (taille du lot = 32 ; taux d'apprentissage = 0,001)

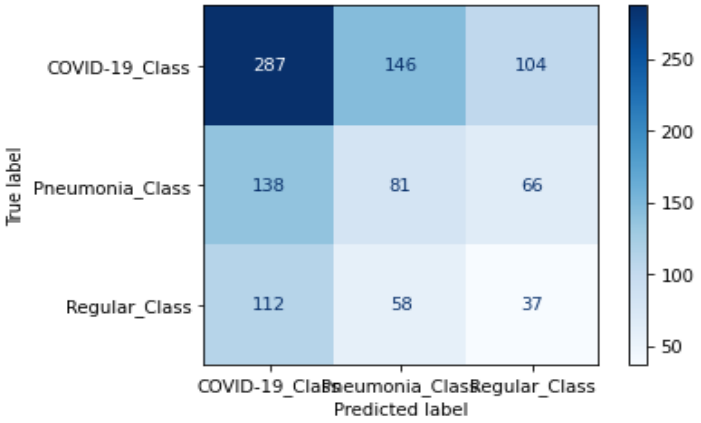

Figure 32: Matrice de confusion VGG16 dans le lot 32 et taux d'apprentissage 0,001

Tableau 9: Rapport de classification VGG16 avec une taille de lot de 32 et un taux d'apprentissage de 0,01

	Précision	Recall	F1-score
Covid-19_class	0.53	0.53	0.53
Pneumonia_class	0.28	0.28	0.28
Regular class	0.18	0.18	0.18

Nous avons conclu qu'après avoir utilisé deux architectures de réseaux naturels profonds différentes pour classer les images échographiques de COVID-19, les hyperparamètres importants qui pourraient être utilisés dans les travaux futurs sont la taille de lot de 32 et le taux d'apprentissage de 0,001, nous ne pouvons pas dire que c'est l'étude limitée mais on peut dire que l'époque aussi c'est des paramètres très importants tant qu'elle offre beaucoup plus d'étapes d'apprentissage pour les réseaux.

Conclusion

Dans ce dernier chapitre, nous avons présenté la conception et les détails de l'implémentation de notre système proposé, on a mis en œuvre un réseau de neurone convolutif pour la classification des images échographiques covid-19 et non covid et pneumonie, Pour la classification des images, on a utilisé une base de données des images échographique des pathologies pulmonaires.

L'entrainement pour la classification d'image a été exploré sur deux algorithme de CNN qui sont VGG-16 et AlexNet dans un environnement de Google Colab, on a entrainé le modèle et montrée l'impact des hyperparamètres comme la taille du lot et Le taux d'apprentissage (Learning rate), et l'epoch, on a justchangéles valeurs de hyperparamètres et remarque la précision et la sensibilité du modèle.

Nous avons conclu que les hyperparamètres ont un impact très important sur les systèmes d'apprentissage basés sur des réseaux de neurones profonds, ainsi que la base de données équilibrée ont également un impact très important sur les performances des modèles d'apprentissage profond

Conclusion générale

Dans ces dernières années dans le domaine de la vision par ordinateur le réseau de neurones convolutionnels est l'un des algorithmes qui ont connu une évolution remarquable a causse de leurs capacités à apprendre.

Dans ce mémoire, nous sommes intéressés à introduire les méthodes d'apprentissage approfondie CNN pour la classification des maladie pulmonaires.

Les avantages après avoir effectué ce travail sont les suivants :

1 - Nous avons compris que le taux d'apprentissage et la taille des lots jouent un rôle crucial dans l'application d'apprentissage en profondeur.

2- Dans le domaine biomédical, le principal problème auquel peuvent être confrontés les chercheurs est le jeu de données déséquilibré.

3- les techniques de prétraitement utilisées dans le traitement d'image sont une étape très importante pour obtenir de bons résultats.

4- le paramétrage des hyperparamètres en faisant varier le taux d'apprentissage et la taille des lots des valeurs les plus hautes aux plus basses c'est une étape obligatoire.

5- les époques ont un effet précieux dans les phases d'apprentissage (train et test) qui donnent aux réseaux profonds le temps d'apprendre en mettant à jour les poids et les biais.

6- L'augmentation des données est l'idée clé pour former le réseau profond, en particulier lorsque les chercheurs disposent d'un ensemble de données limité.

Bibliographie

[1] Muhammad, Ghulam, and M. Shamim Hossain. "COVID-19 and non-COVID-19 classification using multi-layers fusion from lung ultrasound images." *Information Fusion* 72 (2021): 80-88.

[2] Pagano, Antonio, et al. "Lung ultrasound for diagnosis of pneumonia in emergency department." *Internal and emergency medicine* 10.7 (2015): 851-854.

[3] Amatya, Yogendra, et al. "Diagnostic use of lung ultrasound compared to chest radiograph for suspected pneumonia in a resource-limited setting." *International journal of emergency medicine* 11.1 (2018): 1-5.

[4] Shorfuzzaman, Mohammad, and M. Shamim Hossain. "MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients." *Pattern recognition* 113 (2021): 107700.

[5] Hossain, M. Shamim, Ghulam Muhammad, and Nadra Guizani. "Explainable AI and mass surveillance system-based healthcare framework to combat COVID-I9 like pandemics." *IEEE Network* 34.4 (2020): 126-132.

[6] Muhammad, Ghulam, M. Shamim Hossain, and Abdulsalam Yassine. "Tree-based deep networks for edge devices." *IEEE Transactions on Industrial Informatics* 16.3 (2019): 2022-2028.

[7] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." *Science* 349.6245 (2015): 255-260.

[8] https://www.ibm.com/fr-fr/analytics/machine-learning#:~:text=L'apprentissage%20automatique%2C%20%C3%A9galement%20appel%C3%A9,aide%20d'une%20programmation%20explicite

[9] https://www.oracle.com/dz/artificial-intelligence/what-is-ai/

[10]https://www.pwc.fr/fr/decryptages/transformation/intelligence-artificielle-au-service-sante.html#overview

[11] H. Farreny et M. Ghallab, Éléments d'intelligence artificielle, Hermès, 1987.

[12] E.Shortliff e, Computer-based medical Consultations : MYCIN, New York, American Elsevier Publishing, 1976

[13] R. Lindsay, B. Buchman, E. Feigenbaum et J. Lederberg, Th e Dendral project, McGraw Hill, 1980

[14] J. Pitrat, Un démonstrateur automatique de théorèmes, Dunod, 1970.

[15] Sun, Roger, Eric Deutsch, and Laure Fournier. "Intelligence artificielle et imagerie médicale." Bulletin du Cancer (2021).

[16] Mueller, John Paul, and Luca Massaron. *Machine learning for dummies*. John Wiley & Sons, 2021.

[17] Sharkawy A-N, Aspragathos N. Human-Robot CollisionDetection Based on Neural Networks. Int J Mech Eng Robot Res, 2018;. 7(2): 150-157

[18]Sharkawy A-N, Koustoumpardis PN, Aspragathos N. A Neural Network based Approach for Variable Admittance Control in Human- RobotCooperation: Online Adjustment of the Virtual Inertia. Intell Serv Robot 1-37: 2020

[19]Passricha V, Aggarwal RK. Convolutional Neural Networks for Raw Speech Recognition. in From Natural to Artificial Intelligence - Algorithms and Applications, 2018; 21-40.

[20]Palaz D, Magimai-Doss M, Collobert R. Convolutional Neural Networks-Based Continuous Speech Recognition Using Raw Speech Signal. in 2015 IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP), 2015; 4295-4299.

[21]Fadzil MHA, Bakar HA. Human face recognition using neural networks. in Proceedings of 1st International Conference on Image Processing, 1994.

[22]Zhao ZQ, Huang DS, Sun BY. Human face recognition based on multi-features using neural networks committee. Pattern Recognit Lett. 2004; 25(12): 1351-1358,

[23]Fukuoka Y. Artificial Neural Networks in Medical Diagnosis. in Computational Intelligence Processing in Medical Diagnosis Studies in Fuzziness and Soft Computing, S. M., T. HN., J. A., J. A., J. S., and J. L.C., Eds. 2002; 197-228.

[24]Er O, Yumusak N, Temurtas F. Chest diseases diagnosis using artificial neural networks. Expert Syst Appl. 2010; 37(12): 7648-765

[25]Sukthomya W, Tannock J. The training of neural networks to model manufacturing processes. J Intell Manuf. 2005; 16(1): 39-51

[26]Abdelhameed MM, Tolbah FA. Design and implementation of a flexible manufacturing control system using neural network. Int J Flex Manuf Syst. 2002; 14(3): 263-279.

[27] httpphysique.unice.frsem62011-2012PagesWebPTModelisationneruone.html

[28]Benkhelifa. A. (2018). Les systèmes embarqués dans l'automobile. (Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES). Haute École de Gestion de Genèv

[27]Falat L, Pancikova L. Quantitative Modelling in Economics with Advanced Artificial Neural Networks. Procedia Econ Financ. 2015; 34: 194-201

[28] httpphysique.unice.frsem62011-2012PagesWebPTModelisationneruone.html

[29]Découvrez les différentes couches d'un CNN - Classez et segmentez des données visuelles - OpenClassrooms [Accès le 18/06/2021]

[30] missinglink.ai/guides/convolution-neural-networks/fully-connected-layers-convolutional-neuralnetworks-complete-guide/

[31] Qu'est-ce que la fonction de perte? - La Communauté de Support de Huawei Entreprise

[32] https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

[33] https://en.wikipedia.org/wiki/AlexNet