



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA3/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle (IA)**

Développement d'une architecture CNN pour la classification des images radiologiques d'infections pulmonaires

Par :

MAMEN ABDELKARIM

SAOULI RACHIDA

Professeure

Encadreur

Année universitaire 2020-2021

Remerciements

Tout d'abord, je tiens à exprimer ma gratitude à Allah Tout-Puissant. Sans son soutien, il m'aurait été très difficile de terminer cette recherche toute seule.

Deuxièmement, je voudrais remercier mes parents pour avoir cru en moi et m'avoir soutenu dans toutes les situations difficiles.

ensuit, Je tiens à remercier mon superviseur professeur Saouli Rachida, pour m'avoir guidé dans la bonne voie tout au long du projet.

Enfin, j'aimerais remercier tous les autres membres du département, mes collègues et quelques-uns de mes amis. Ils m'ont toujours gardé motivé et m'ont aidé en me donnant du soutien mental et de la force pour finir ma thèse.

Mamen Abdelkarim

Abstract

The analysis of radiological images is primarily performed by medical specialists, as this is a critical area and patients expect the highest level of care and service, regardless of cost. Despite various advantages of chest radiography, the interpretation of the X-ray image has always been a major issue for the physician and radiologist because of diagnostic errors. Our main objective is to analyze medical X-ray images using deep learning to achieve the classification of viral pneumonia, Covid, bacterial pneumonia and tuberculosis diseases. Our goal is to classify as many lung diseases as possible and thus develop the best performing model that can distinguish them. For this purpose, we used the convolutional neural network (CNN) algorithm because deep learning classification approaches based on CNNs have the ability to automatically extract high-level representations from large data using little preprocessing compared to other image classification algorithms. Our model provides very encouraging results compared to those obtained by works in the literature. This will lead clinicians to better diagnostic decisions for patients to provide them with solutions with good accuracy for medical imaging.

Keywords : Convolutional Neural Networks (CNN), X-ray, Deep Learning, Keras, Radiography, TensorFlow, OpenCV and Artificial Intelligence.

Résumé

L'analyse des images radiologiques est principalement effectuée par des spécialistes médicaux, car il s'agit d'un secteur critique et les malades attendent le plus haut niveau de soins et de services, quel que soit le coût. Malgré divers avantages de la radiographie pulmonaire, l'interprétation de l'image *X-ray* a toujours été un enjeu majeur pour le médecin et le radiologue à cause des erreurs de diagnostic. Notre objectif principal est d'analyser les images radiologiques médicales en utilisant l'apprentissage profond pour parvenir à la classification de maladies : pneumonie virale, covid, pneumonie bactérienne et tuberculose. Notre but est de classifier le maximum de maladies pulmonaires et donc de développer un modèle le plus performant possible qui peut les distinguer. Pour cela, nous avons utilisé l'algorithme des réseaux neuronaux convolutifs (CNN) car les approches de classification par apprentissage profond basées sur les CNN ont la capacité d'extraire automatiquement les représentations de haut niveau à partir de données volumineuses en utilisant peu de prétraitement par rapport aux autres algorithmes de classification d'images. Notre modèle fournit des résultats très encourageants par rapport à ceux obtenus par les travaux de la littérature. Ce qui conduira les cliniciens vers de meilleures décisions de diagnostic pour les patients afin de leur fournir des solutions avec une bonne précision pour l'imagerie médicale.

Mots clés : Réseaux neuronaux convolutifs (CNN), radiographie, Deep Learning, Keras, Radiographie, TensorFlow, OpenCV et Intelligence Artificielle.

Table des figures

- 2.1 les poumons 6
- 2.2 Schéma de la plèvre 7
- 2.3 Structure globale des poumons 8
- 2.4 Structure des lobes 9
- 2.5 modèle double circulation 10
- 2.6 La première radiographie médicale 12
- 2.7 chest x-ray 15
- 2.8 Exemple d'apprentissage supervisé 17
- 2.9 un neurone biologique 20
- 2.10 Un neurone artificiel 20
- 2.11 Un exemple de neurone montrant la fonction d'activation f appliquée à la somme pondérée de l'entrée. 21
- 2.12 Exemple d'apprentissage non supervisé 24
- 2.13 L'apprentissage par renforcement dans le dressage des chiens 26
- 2.14 RNN exemple 29
- 2.15 Une architecture CNN 30

2.16 Architecture du modèle LeNet	31
2.17 Architecture du modèle AlexNet	32
2.18 Architecture du modèle VGGNet	33
2.19 Couche Inception avec réduction de dimension	33
3.1 l'architecture générale	36
3.2 architecture de MP-CNN	37
3.3 Opération de prétraitement	39
3.4 Division de l'ensemble de données prétraitées	40
3.5 MP-CNN architecture	42
4.1 Google Colab	45
4.2 pycharm	46
4.3 python	46
4.4 tensorflow	47
4.5 MP-CNN paramètres	55
4.6 Capture d'écran de l'interface	57
4.7 accuracy	58
4.8 loss	59
4.9 matrice de confusion	61
4.10 matrice de confusion pour base de données de [27]	63

Liste des codes sources

- 4.1 redimensionnement de la base de données 49
- 4.2 Diviser la base de données avec un ratio 49
- 4.3 Importer des bibliothèques et des modules 50
- 4.4 Chargement de base de données 53
- 4.5 matrice de confusion 56

Liste des tableaux

2.1	Quelques fonctions d'activation	22
2.2	Comparaison des travaux relatifs	34
3.1	Les hyperparamètres utilisés dans le réseau de neurones à convolution	42
4.1	Diviser avec un ratio	50
4.2	Description des bibliothèques et modules utilisés	52
4.3	Chargement de base de données	53
4.4	Tableau comparatif des travaux connexes	62

Table des matières

Acknowledgements	i
Abstract	ii
Résumé	iii
Table des figures	vii
Liste des codes sources	vii
Liste des tableaux	viii
1 Introduction générale	1
2 Etat de l'art : Classification des maladies pulmonaires	5
2.1 Introduction	5
2.2 Notions d'anatomie pulmonaire	6
2.2.1 Cavité pleurale	6
2.2.2 Structure globale des poumons	7
2.2.3 Structure des lobes	8
2.2.4 Double vascularisation	9
2.3 Pathologies et maladies pulmonaires	10

2.4	présentation des images X-rays	11
2.4.1	Historique	11
2.4.2	Fonctionnement	13
2.4.3	Radiographies pulmonaires	13
2.5	Méthode de Classification d'images médicales	15
2.5.1	Méthodes de classification conventionnelles	16
2.5.2	Méthodes de classification basée sur l'apprentissage profond	27
2.6	Travaux relatifs et Synthèse	33
2.7	Conclusion	34
3	Conception de l'architecture pour la classification des maladies pulmonaires	35
3.1	Introduction	36
3.2	Architecture générale	36
3.3	Architecture détaillée	37
3.3.1	Préparation des données	37
3.3.1.1	Les bases de données utilisées	37
3.3.1.2	Prétraitement	38
3.3.1.3	Division de la base de données	39
3.3.2	<i>Training</i> le modele de l'apprentissage	40
3.3.3	la phase de test du modèle	43
3.4	Conclusion	44
4	Implémentation de l'architecture MP-CNN	45
4.1	Introduction	45

<i>TABLE DES MATIÈRES</i>	0
4.2 Environnements et outils de développement	45
4.3 Développement de le back-end	49
4.3.1 Préparation des données	49
4.3.2 Réalisation du modèle MP-CNN	50
4.3.2.1 Importer des bibliothèques et des modules	50
4.3.2.2 Initialisation des paramètres MP-CNN	53
4.3.2.3 Chargement de la base de données	53
4.3.2.4 Les parametres d'apprentissage du réseau	54
4.3.2.5 resultat de la prédiction	55
4.4 Présentation de l'interface	56
4.5 Résultats obtenus	57
4.5.1 Présentation des performances obtenu	57
4.5.2 Évaluation du modèle sur les données de test	60
5 Conclusion et perspectives	65
5.1 Conclusion	65
5.2 Perspectives	66
References	67

Chapitre 1

Introduction générale

Les radiographies médicales base sur les rayon-x sont des images généralement utilisées pour diagnostiquer certaines parties sensibles du corps humain telles que les os, la poitrine, les dents, le crâne, etc. Les experts médicaux utilisent cette technique depuis plusieurs décennies pour explorer et visualiser des fractures ou des anomalies dans les organes du corps [1] . Cela est dû au fait que les rayons X sont des outils de diagnostic très efficaces pour révéler les altérations pathologiques, en plus de ses caractéristiques non invasives et de considérations économiques. Les maladies thoraciques peuvent être montrées dans les images CXR sous la forme de cavitations, de consolidations, d'infiltrats, d'angles costophréniques émoussés et de petits nodules largement distribués. L'interprétation d'une radiographie pulmonaire permet de diagnostiquer de nombreuses affections et maladies telles que pleurésie, épanchement, pneumonie, bronchite, infiltration, nodule, atélectasie, péricardite, cardiomégalie, pneumothorax, fractures et bien d'autres [30] .

La classification des anomalies de la radiographie pulmonaire est considérée comme une tâche difficile pour les radiologues. Par conséquent, au cours des dernières décennies, des systèmes de diagnostic assisté par ordinateur (DAO) ont été développés pour extraire des informations

utiles des rayons X afin d'aider les médecins à avoir un aperçu quantitatif d'une radiographie. Cependant, ces systèmes DAO n'ont pas atteint un niveau de signification pour prendre des décisions sur le type de conditions de maladies dans une radiographie [22]. Ainsi, leur rôle a été laissé en tant que fonctionnalité de visualisation qui aide les médecins à prendre des décisions. Récemment, une classification précise des images a été réalisée par des systèmes basés sur l'apprentissage en profondeur. Ces réseaux profonds ont montré des précisions surhumaines dans l'exécution de telles tâches. Ce succès a motivé les chercheurs à appliquer ces réseaux sur des images médicales pour des tâches de classification de maladies et les résultats ont montré que les réseaux profonds peuvent efficacement extraire des caractéristiques utiles qui distinguent différentes classes d'images [18]. Les réseaux de neurones convolutifs ont été appliqués au diagnostic et à la classification de diverses images médicales en raison de leur pouvoir d'extraction de caractéristiques de différents niveaux à partir d'images.

Malgré divers avantages de la radiographie pulmonaire, l'interprétation de l'image X-ray a toujours été un enjeu majeur pour le médecin et le radiologue à cause des erreurs de diagnostic des différentes lésions pulmonaires. De plus, les chercheurs ont découvert que les poumons des patients présentant les symptômes du COVID-19 présentent certaines marques visuelles comme des opacités en verre dépoli - des taches sombres qui peuvent différencier les patients infectés par le COVID-19 de ceux qui ne le sont pas. [23]

Le patient COVID-19 peut développer divers symptômes et signes d'infection, dont la fièvre, la toux et une maladie respiratoire (comme la grippe). Dans les cas graves, l'infection peut provoquer une pneumonie, des difficultés respiratoires, une défaillance de plusieurs organes, et cela peut nous conduire à un faux examen. [27]

Dans ce contexte, l'objectif de ce travail est de fournir une classification des images radiologiques d'infections pulmonaires, et comment les méthodes Deep Learning ont le potentiel d'être l'approche plus nécessaire pour la tâche de classification, en présentant une application spécifique pour le diagnostic des radiographies pulmonaires et ceci en essayant de prendre en compte le plus de maladies possibles (les maladies connues précédemment ainsi que la maladie pulmonaire la plus récente comme le covid-19).

Cette mémoire est structurée comme suit :

Le chapitre (2) est un état de l'art qui commence par un bref aperçu des notions d'anatomie pulmonaire, les principes physiques des rayons x, De plus ce chapitre présente une description des approches d'apprentissage profond basées sur les réseaux de neurones ainsi qu'une synthèse de quelques travaux proposés dans la littérature relatifs aux objectifs lancés dans ce projet.

Le chapitre (3) décrit la conception de notre contribution qui correspond à une nouvelle architecture de réseau de neurones profonds pour la classification de plusieurs maladies (4 maladies au lieu par rapport aux travaux connexes) sur les images radiographiques est également décrit dans ce chapitre.

Le chapitre (4) décrit l'implémentation de notre architecture et les outils utilisés dans ce projet, et comment mettre en œuvre le système d'apprentissage en profondeur pour la classification des maladies pulmonaires. L'évaluation empirique et les résultats sont également présentés

dans ce chapitre.

En conclusion, nous résumons et passons en revue nos idées et résultats en donnant des améliorations possibles et quelques perspectives.

Chapitre 2

Etat de l'art : Classification des maladies pulmonaires

2.1 Introduction

La précision de l'analyse et de l'interprétation des images est cruciale pour l'établissement de meilleurs diagnostics. Bien que l'interprétation des images par des algorithmes d'apprentissage automatique conventionnels dépende principalement de caractéristiques élaborées par des experts, la vision par ordinateur est la meilleure application de l'apprentissage automatique. Aujourd'hui, dans tous les domaines, en particulier dans l'analyse d'images médicales, l'apprentissage profond a fait un grand pas en avant.

Dans ce chapitre nous présentons quelques éléments d'anatomie pulmonaire, les principes physiques des rayons x, la relation entre l'imagerie médicale et l'Artificial Intelligence, les méthodes de classification conventionnelles et les méthodes de classification basée sur l'apprentissage profond.

2.2 Notions d'anatomie pulmonaire

Il existe deux poumons, situés dans la poitrine, et plus particulièrement dans la cavité thoracique où ils occupent la majeure partie de la cavité thoracique. Les poumons gauche et droit sont séparés par le médiastin, situé au centre du médiastin, composé notamment du cœur. (Figure 2.1)[30]



FIGURE 2.1 – les poumons

2.2.1 Cavité pleurale

Chaque poumon est entouré d'une cavité pleurale (Figure 2.2), formée de deux couches de membranes : Un feuillet interne en contact avec les poumons s'appelle la plèvre pulmonaire. Un feuillet externe en contact avec la paroi thoracique est appelé plèvre pariétale. Cette cavité est composée de liquide séreux (exsudat), qui permet aux poumons de glisser. Le kit aide également à soutenir les poumons et à les empêcher de s'affaisser. [30]

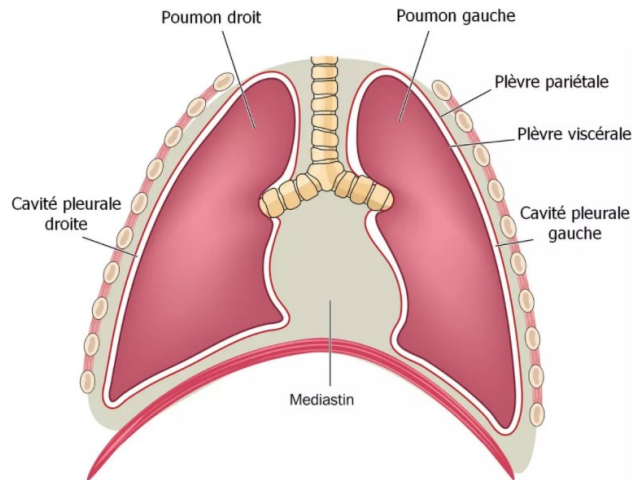


FIGURE 2.2 – Schéma de la plèvre

2.2.2 Structure globale des poumons

Les poumons droit et gauche sont reliés par les bronches et la trachée.

La trachée est les voies respiratoires du larynx. Elle passe entre les deux poumons dans la moitié supérieure et se divise en bronches gauche et droite. (Figure 2.3)

Chaque bronche s'adapte aux poumons. Dans les poumons, les bronches se divisent pour former des structures de plus en plus petites, jusqu'aux bronchioles terminales.

Forme pyramidale avec plusieurs faces dans les poumons :

- La surface extérieure, adjacente à la grille nervurée.
- La surface interne où la bronche pénètre et les vaisseaux sanguins circulent.
- La base, placée sur le diaphragme.

Les poumons sont également composés de lobes, séparés par des scissures : deux pour le poumon gauche et trois pour le poumon droit. [30]

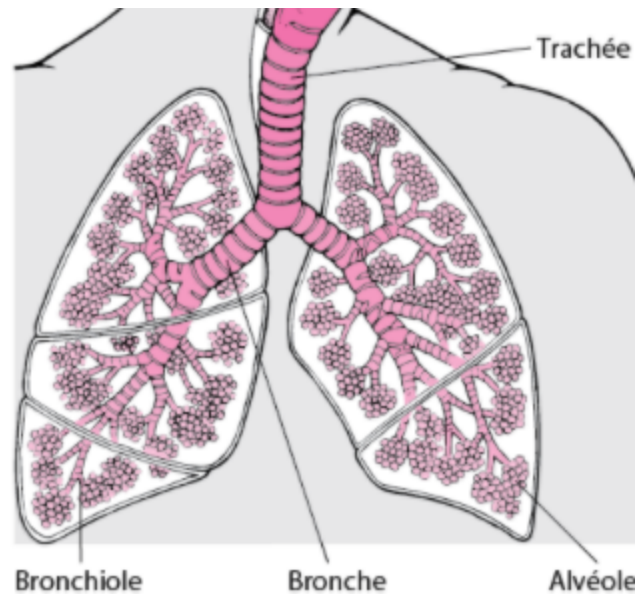


FIGURE 2.3 – Structure globale des poumons

2.2.3 Structure des lobes

Chaque lobe est composé de petits poumons et sa fonction est similaire à celle des petits poumons (Figure 2.4). Ils contiennent des branches des bronches et des artères et veines pulmonaires. Les extrémités des bronches sont appelées bronchioles et forment un sac : l'acinus. Cette dernière est constituée de multiples masses : les alvéoles. La paroi acinaire est très fine, en contact avec l'air dans les bronchioles et le réseau formé par les capillaires pulmonaires. [30]

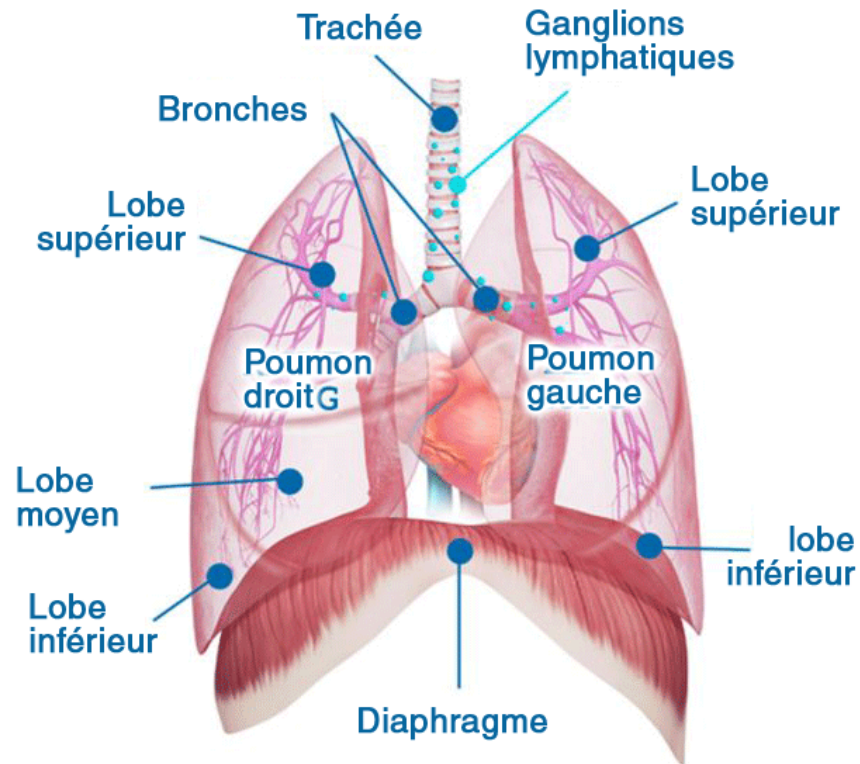


FIGURE 2.4 – Structure des lobes

2.2.4 Double vascularisation

Les poumons reçoivent une double vascularisation (Figure 2.5) :

- Vascularisation fonctionnelle formée par le réseau des artères et veines pulmonaires, permettant l'oxygénation du sang.
- L'apport de vaisseaux nutritifs composés d'artères et de veines bronchiques fournit les éléments de base pour le fonctionnement normal des poumons [30]

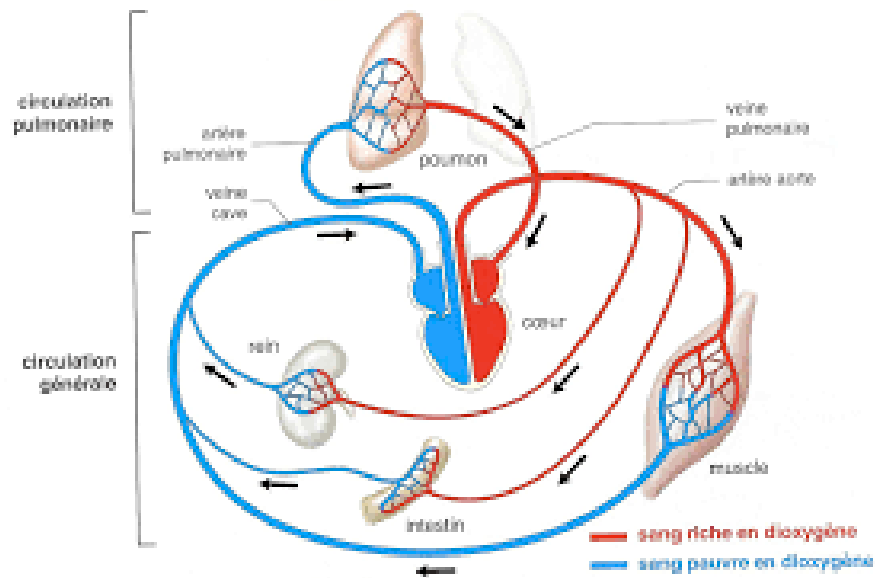


FIGURE 2.5 – modèle double circulation

2.3 Pathologies et maladies pulmonaires

Pneumothorax Cette condition pathologique est le passage d'air anormal dans l'espace entre la cavité pleurale, les poumons et la cage thoracique. Elle se manifeste par de fortes douleurs thoraciques, parfois accompagnées de difficultés respiratoires. [30]

Pneumonie Cette affection est une infection respiratoire aiguë qui affecte directement les poumons. Les alvéoles sont affectées et remplies de pus, provoquant des problèmes respiratoires. Les infections sont principalement causées par des bactéries, des virus ou des champignons. [30]

Tuberculose Cette maladie correspond à une infection bactérienne se retrouvant souvent au niveau des poumons. Les symptômes sont une toux chronique avec émissions de sang, des fièvres intenses avec sueurs nocturnes, ainsi qu'une perte de poids. [30]

Bronchite aiguë Cette affection est due à une infection des bronches, généralement une infection virale. Elle arrive souvent en hiver, provoquant toux et fièvre.[30]

Cancer du poumon Des cellules tumorales malignes peuvent se développer au niveau du poumon et des bronches. Ce type de cancer est l'un des plus répandus dans le monde.[30]

Exploration et examens Examen clinique : afin d'évaluer la pathologie, la respiration, les poumons et les symptômes du patient ont été analysés.

Examen d'imagerie médicale : une radiologie x-ray des poumons, un scanner thoracique, une IRM ou encore une scintigraphie pulmonaire peuvent être pratiqués pour confirmer un diagnostic.

2.4 présentation des images X-rays

2.4.1 Historique

C'est le physicien allemand Wilhelm Conrad Roentgen qui a découvert les rayons X le 8 novembre 1895, alors qu'il travaillait avec des tubes à rayons cathodiques (genre de cylindres sous vide et scellés dotés d'un canon à électrons à une extrémité et d'un écran conçu pour capter des images à l'autre). Roentgen avait alors remarqué que l'écran qu'il utilisait (une plaque photographique composée d'un produit chimique appelé « platinocyanure de baryum ») devenait fluorescent (émettait de la lumière), et le restait même après qu'il ait bloqué les rayons cathodiques. Il a baptisé ce nouveau type de radiation électromagnétique le « rayonnement X », parce

qu'il ne savait pas vraiment de quoi il s'agissait et qu'en mathématique, le « x » représente une valeur inconnue. On a plus tard raccourci le terme en utilisant l'expression.[34]

la Figure 2.6 présenter la première radiographie médicale.



FIGURE 2.6 – La première radiographie médicale

Il a aussi découvert que ces rayons pouvaient faire apparaître des images sur de la pellicule photographique. En reconnaissance de ses travaux novateurs, Roentgen a reçu de nombreuses distinctions, dont le premier prix Nobel de physique en 1901. L'unité de mesure qui a longtemps été utilisée pour démontrer le niveau de l'exposition aux rayons X (R, pour roentgen) a été nommée en son honneur. Aujourd'hui, nous utilisons des Coulombs par kilogramme (C/kg).

2.4.2 Fonctionnement

Les appareils de radiographie sont essentiellement des caméras spécialisées. Dans leur cas, au lieu de la lumière visible, ce sont des rayons X qui exposent une pellicule de façon à créer des images. Les tissus osseux, qui contiennent beaucoup de calcium, absorbent ces rayons, alors que les tissus mous, comme la peau et le gras, les laissent passer. Sur une radiographie, les zones sombres correspondent donc aux tissus moins compacts, alors que les zones plus claires représentent les endroits où les tissus sont plus denses ou épais. Voilà pourquoi les objets métalliques y apparaissent en blanc[32].

2.4.3 Radiographies pulmonaires

Un test radiologique thoracique est un test radiologique extrêmement normal et non pénétrant qui crée une image de la poitrine et des organes internes. Pour faire un test thoracique par faisceau X (Figure 2.7) . le thorax est rapidement présenté au rayonnement d'un appareil à rayon X et une image est créée sur un film ou dans un ordinateur informatisé . En fonction de son épaisseur, chaque organe à l'intérieur de la fosse thoracique retient des degrés de rayonnement fluctuants, créant des ombres distinctes sur le film. Les images des rayons X de la poitrine sont très contrastées avec juste la brillance ou le flou caractérisant les différentes structures. Par exemple, les os du diviseur thoracique (côtes et vertèbres) peuvent assimiler une plus grande quantité de rayonnement et, le long de ces lignes, semblent plus blancs sur le film.[20] Là encore, le tissu pulmonaire, qui est pour la plupart composé d'air, permettra à la majorité du rayonnement de passer, donnant au film une apparence plus sombre. Le cœur et l'aorte sembleront blanchâtres, mais généralement moins brillants que les os, qui sont plus denses. Les tests

de radiographie thoracique sont demandés par les médecins pour diverses raisons. De nombreuses conditions cliniques peuvent être évaluées par ce test radiologique de base. Une partie des conditions de base reconnues sur une radiographie thoracique comprend :

- Pneumonie,
- Hypertrophie du cœur,
- Lung mass,
- Fissures des côtes,
- Liquide autour du poumon (irradiation pleurale),
- Air autour du poumon (pneumothorax),
- Covid-19,
- Opacité pulmonaire,
- Tuberculose

Dans l'ensemble, un test de rayon X thoracique est un système simple, rapide, économique et modérément inoffensif avec un risque de rayonnement insignifiant. Il est en outre largement accessible.



FIGURE 2.7 – chest x-ray

2.5 Méthode de Classification d'images médicales

l'imagerie radiographique(imagerie par rayons X) est la modalité d'imagerie la plus courante dans la plupart des cabinets médicaux du monde entier. En raison du grand nombre de radiographies réalisées chaque jour, la radiographie est une cible de choix pour le développement et la mise en œuvre de solutions d'intelligence artificielle (IA) susceptibles d'accroître l'efficacité et d'améliorer la qualité. Des progrès significatifs ont été réalisés dans le domaine de l'IA grâce au développement d'approches d'apprentissage automatique, notamment l'apprentissage par représentation et l'apprentissage profond, à l'augmentation de la capacité de calcul et à l'accroissement des données disponibles pour l'entraînement des algorithmes. Ces progrès ont relancé les efforts des groupes de recherche universitaires et de l'industrie pour développer des solutions d'IA. De nombreux efforts se sont concentrés sur les radiographies conventionnelles

en raison de leur importance dans les pratiques radiologiques, des grandes quantités de données d'image disponibles pour l'entraînement des algorithmes et de leur simplicité en tant que représentation bidimensionnelle d'un objet tridimensionnel.[17]

2.5.1 Méthodes de classification conventionnelles

Machine learning (ML) est une branche de l'intelligence artificielle qui applique systématiquement des algorithmes pour synthétiser les relations sous-jacentes entre les données et les informations. Par exemple, les systèmes d'apprentissage automatique peuvent être entraînés sur des systèmes de reconnaissance automatique de la parole (comme le Siri de l'iPhone) pour convertir les informations acoustiques d'une séquence de données vocales en structure sémantique exprimée sous la forme d'une chaîne de mots. La ML est déjà largement utilisée dans la recherche sur le web, l'évaluation du crédit, les prévisions boursières, l'analyse des séquences génétiques, l'analyse du comportement, les coupons intelligents, le développement de médicaments, les prévisions météorologiques, l'analyse des big data et bien d'autres applications encore.[19].

La ML jouera un rôle décisif dans le développement d'une multitude d'innovations centrées sur l'utilisateur . En 1959, Arthur Samuel a décrit la ML comme le "domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés". Il concluait que la programmation des ordinateurs pour qu'ils apprennent par l'expérience devrait, à terme, éliminer la nécessité d'une grande partie de cet effort de programmation détaillée. Selon la définition de Tom M. Mitchell de la ML : "On dit qu'un programme informatique apprend de l'expérience E en ce qui concerne une certaine classe de tâches T et une mesure de performance P, si sa performance aux tâches dans T, telle que mesurée par P, s'améliore avec l'expérience E [19].

Méthodes d'apprentissage Les algorithmes d'apprentissage automatique sont organisés en taxonomie, en fonction du résultat souhaité de l'algorithme.

Les types d'algorithmes les plus courants sont : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

A) Techniques d'apprentissage supervisé Lorsque la machine apprend la corrélation entre les étiquettes et les données, on parle d'apprentissage supervisé. Toutes les tâches de classification dépendent d'ensembles de données étiquetées ex : Détecter les visages, identifier les personnes dans les images, reconnaître les expressions faciales (en colère, joyeux). L'apprentissage profond est basé sur l'apprentissage supervisé.[11]

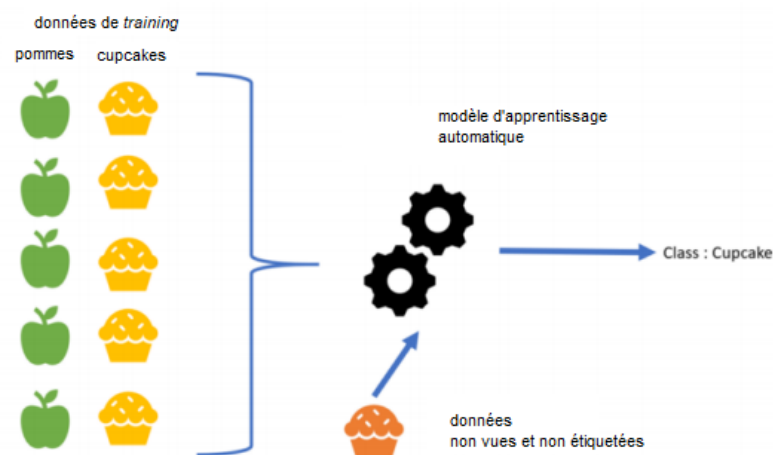


FIGURE 2.8 – Exemple d'apprentissage supervisé

selon la figure 2.8, la machine apprend qu'il y a deux classes et peut maintenant prédire la classe d'un nouvel objet.

Techniques d'apprentissage supervisé : Arbre de décision ,Réseaux de neurone ,Support Vector Machine, Régression logistique.[15]

(a) Arbre de décision Un arbre de décision est un schéma représentant les résultats possibles d'une série de choix interconnectés. Il permet à une personne ou une organisation d'évaluer différentes actions possibles en fonction de leur coût, leur probabilité et leurs bénéfices.[12]

1. avantages [12] :

- Le coût d'utilisation de l'arbre pour prédire des données diminue à chaque point de donnée supplémentaire.
- Ils fonctionnent aussi bien pour les données de catégorie que numériques.
- La modélisation des problèmes est possible avec plusieurs données de sortie.
- Ils utilisent un modèle de boîte blanche, ce qui rend les résultats faciles à expliquer.
- La fiabilité d'un arbre peut être testée et quantifiée.
- Ils tendent à être précis, même si les hypothèses des données source ne sont pas respectées.

2. inconvénients [12] :

- Lors de la gestion de données de catégorie comportant plusieurs niveaux, le gain d'information est biaisé en faveur des attributs disposant du plus de niveaux.
- Les calculs peuvent devenir compliqués lorsqu'une certaine incertitude est de mise et que de nombreux résultats sont liés entre eux.
- Les conjonctions entre les nœuds sont limitées à l'opérateur « ET », alors que les graphiques décisionnels permettent de connecter des nœuds avec l'opérateur « OU ».

(b) Support Vector Machine Une machine à vecteurs de support, est un algorithme d'apprentissage automatique supervisé qui peut être utilisé à des fins de classification et de régression. Les SVM sont plus généralement utilisés dans les situations de classification. [16]

1. Avantages [16] :

- Sa grande précision de prédiction
- Fonctionne bien sur de plus petits data sets
- Ils peuvent être plus efficace car ils utilisent un sous-ensemble de points d'entraînement.

2. inconvénients [16] :

- Ne convient pas aux grands ensembles de données, car le temps de formation avec les SVM peut être long.
- Moins efficace sur les ensembles de données contenant du bruit

(c) Réseaux de neurones artificiels est un modèle de calcul inspiré des neurones naturels. Les neurones naturels reçoivent des signaux via des synapses situées sur les dendrites du neurone. Lorsque les signaux reçus, le neurone est activé et émet un signal à travers l'axone. Ce signal pourrait être envoyé à une autre synapse et activer d'autres neurones. Un neurone biologique illustré à la figure 2.9 [5]

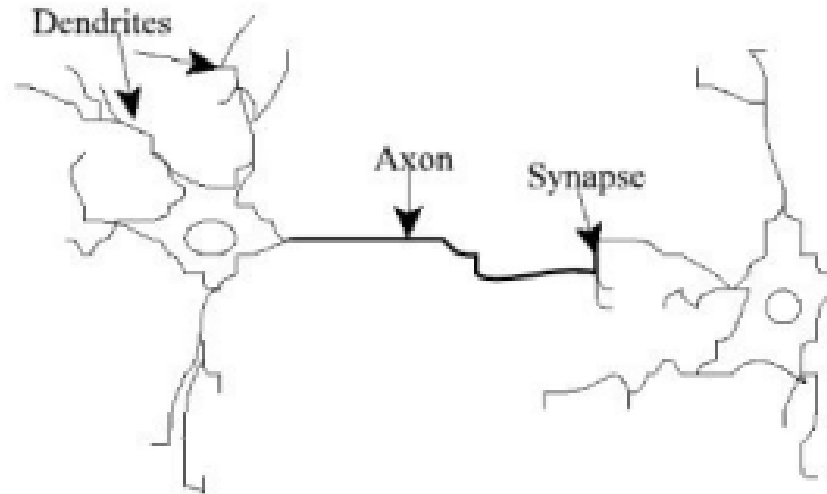


FIGURE 2.9 – un neurone biologique

Lors de la modélisation des neurones artificiels. Celles-ci consistent essentiellement en des entrées (comme les synapses), qui sont multipliées par des poids, puis calculées par une fonction mathématique qui active le neurone. Calculer la sortie de la fonction neuronale artificielle (parfois en fonction d'un certain seuil et il peut s'agir de l'identité) [5]. Modèle d'un neurone artificiel illustré sur la figure 2.10 et la figure 2.11.

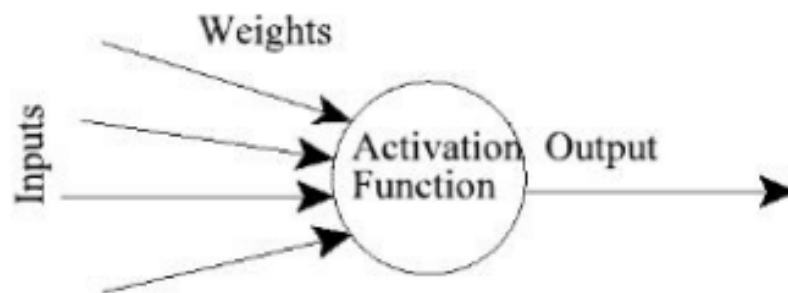


FIGURE 2.10 – Un neurone artificiel

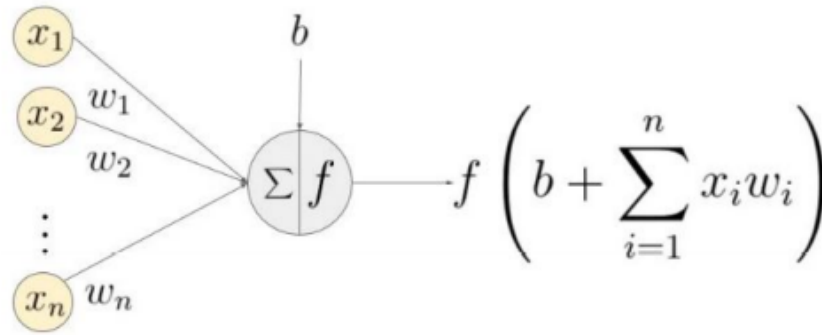


FIGURE 2.11 – Un exemple de neurone montrant la fonction d'activation f appliquée à la somme pondérée de l'entrée.

- $x_1 \dots x_n$ sont les entrées du neurone.
- $w_0 \dots w_n$ sont les poids.
- Le produit du poids et de l'entrée donne la force du signal.
- Sum :

$$\sum_{i=0}^n x_i w_i$$

Il y a diverses fonctions utilisées pour l'activation comme Sigmoid etc ... montrées dans le tableau 2.1.

TABLE 2.1 – Quelques fonctions d'activation

Name	Class
Hard Limit	$\alpha = 0$ $n < 0$
	$\alpha = 1$ $n \geq 0$
Symmetrical Hard Limit	$\alpha = -1$ $n < 0$
	$\alpha = +1$ $n \geq 0$
Linear	$\alpha = n$
Log-sigmoid	$\sigma(z) = \frac{1}{1 + e^{-z}}$
Rectified linear activation Function	$\text{Relu}(z) = \max(z, 0)$ $\text{Relu}'(x) = ($ $\quad 1 \text{ } x > 0$ $\quad 0 \text{ otherwise}$ $)$

(d) Régression logistique le modèle logistique est utilisé pour modéliser la probabilité d'une certaine classe ou d'un événement existant tel que réussite/échec, victoire/perd, vivant/mort ou sain/malade. Cela peut être étendu pour modéliser plusieurs classes d'événements tels que déterminer si une image contient un chat, un chien, un lion, etc. Chaque objet détecté dans l'image se verrait attribuer une probabilité comprise entre 0 et 1.[21]

1. Avantages

- La régression logistique est plus facile à mettre en œuvre, à interpréter et très efficace à entraîner.
- Elle est très rapide pour classer les enregistrements inconnus.
- Elle offre une bonne précision pour de nombreux ensembles de données simples et donne de bons résultats lorsque l'ensemble de données est linéairement séparable.

2. Inconvénients

- Si le nombre d'observations est inférieur au nombre de caractéristiques, la régression logistique ne doit pas être utilisée, sinon, elle peut conduire à un ajustement excessif.
- La principale limite de la régression logistique est la supposition de linéarité entre la variable dépendante et les variables indépendantes.
- Les problèmes non linéaires ne peuvent pas être résolus avec la régression logistique.

B) Techniques d'apprentissage non supervisé L'apprentissage est-il sans étiquettes. Une loi de l'apprentissage automatique est la suivante : plus un algorithme peut s'entraîner sur de données, plus il sera précis. Par conséquent, l'apprentissage non supervisé a le potentiel de produire des modèles très précis. L'apprentissage en profondeur ne nécessite pas d'étiquettes pour détecter les similitudes (clustering).[11]

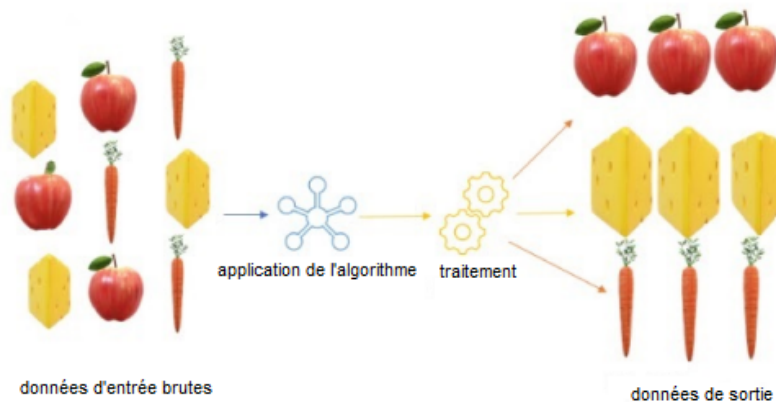


FIGURE 2.12 – Exemple d'apprentissage non supervisé

selon la figure 2.12, il y a trois classes non étiquetées, la machine les distinguera toutes les trois, mais il ne peut pas dire si un cluster donné est une pomme ou non, car il n'est pas étiqueté, mais toute nouvelle donnée s'intégrera automatiquement dans les clusters qui sont formés.

Techniques d'apprentissage non supervisé : la classification hiérarchique, l'algorithme k-moyennes (k-means), la classification basée sur une densité, la classification basée sur des modèles statistiques/probabilistes.

(a) K-means est une technique de classification par apprentissage automatique utilisée pour simplifier des ensembles de données volumineux en ensembles de données simples et plus petits. Des modèles distincts sont évalués et des ensembles de données similaires sont regroupés. [3]

1. Avantages [4] :

- L'algorithme de k-means est très populaire du fait qu'il est très facile à comprendre et à mettre en œuvre.
- Sa simplicité conceptuelle et sa rapidité

- Applicable à des données de grandes tailles, et aussi à tout type de données (mêmes textuelles), en choisissant une bonne notion de distance.

2. Inconvénients [4] :

- Le nombre de classe doit être fixé au départ,
- Le résultat dépend de tirage initial des centres des classes,
- Les clusters sont construits par rapports à des objets inexistantes (les milieux)

(b) algorithmes hiérarchiques Les méthodes dites descendantes partent d'une solution générale vers une autre plus spécifique. Les méthodes de cette catégorie démarrent avec un seul cluster contenant la totalité puis se divisent à chaque étape selon un critère jusqu'à l'obtention d'un ensemble de clusters différents.

1. avantages :[29] :

- flexibilité concernant le niveau de finesse de la classification.
- facilité de prise en compte de distances et d'indices de similarité de n'importe quel type.

2. inconvénients [29] :

- il est difficile de déterminer la coupure significative de l'arbre .
- la partition retenue a une étape dépend de celle obtenue à l'étape précédente .
- les algorithmes fournissent toujours des classes à partir de n'importe quelles données.

(c) DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) Il s'agit d'un algorithme fondé sur la densité dans la mesure qui s'appuie sur la densité estimée des clusters pour

effectuer le partitionnement. [9]

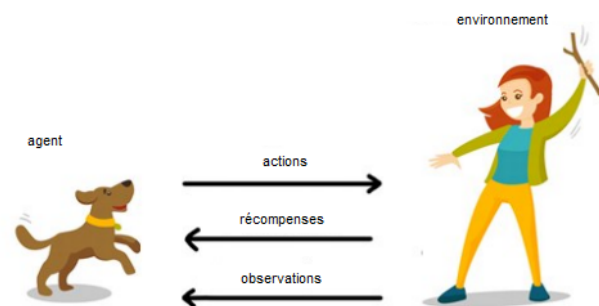
1. Avantages :

- DBSCAN ne nécessite pas que l'on spécifie le nombre de clusters au préalable.
- DBSCAN est performant avec des clusters de forme arbitraire.
- DBSCAN a une notion de bruit, et est robuste aux valeurs aberrantes.

2. Inconvénients :

- Le choix d'une valeur eps significative peut être difficile si les données ne sont pas bien comprises.
- DBSCAN n'est pas entièrement déterministe. C'est parce que l'algorithme commence avec un point aléatoire.

C) Techniques d'apprentissage par renforcement L'apprentissage par renforcement fait référence à des algorithmes axés sur les objectifs, qui apprennent à atteindre un objectif complexe (but) ou à maximiser le long d'une dimension particulière sur de nombreuses étapes. Il se réfère également à partir d'une ardoise vierge, et dans les bonnes conditions pour obtenir de meilleures performances ex : Prédictive Analytics (Régressions).[11]



3

FIGURE 2.13 – L'apprentissage par renforcement dans le dressage des chiens

Dans l'exemple de la figure 2.13, l'objectif de l'apprentissage par renforcement est d'apprendre au chien (agent) à accomplir une tâche dans un environnement, qui comprend l'entourage du chien et du dresseur. Tout d'abord, le dresseur donne un ordre ou une indication, que le chien observe (observation).

Le chien réagit alors en effectuant une action. Si l'action est proche du comportement souhaité, le dresseur fournira probablement une récompense, telle qu'une friandise ou un jouet; dans le cas contraire, aucune récompense ou une récompense négative sera fournie.

2.5.2 Méthodes de classification basée sur l'apprentissage profond

Deep learning (DL), nouveau domaine de recherche en apprentissage automatique, est un processus qui permet à l'ordinateur d'apprendre à effectuer des tâches naturelles pour le cerveau, comme la reconnaissance d'images. Actuellement, les méthodes d'apprentissage profond ont eu un impact profond sur les applications de vision par ordinateur et d'analyse d'images, telles que la classification, la segmentation et la complétion d'images, etc. L'apprentissage profond se concentre sur une catégorie spécifique d'apprentissage automatique appelée réseaux neuronaux artificiels, qui s'inspire de la fonctionnalité du cerveau humain. L'apprentissage profond moderne fournit un cadre très puissant pour l'apprentissage supervisé. En ajoutant plus de couches et plus d'unités dans une couche, un réseau profond peut représenter des fonctions de complexité croissante.

La plupart des tâches qui consistent à mettre en correspondance un vecteur d'entrée et un vecteur de sortie, et qui sont faciles à réaliser rapidement par une personne, peuvent être accomplies par l'apprentissage profond, avec des modèles suffisamment grands et des ensembles suffisamment grands d'exemples d'entraînement étiquetés.[10]

Approches d'apprentissage en profondeur Dans l'apprentissage en profondeur, un modèle informatique apprend à effectuer des tâches de classification directement à partir d'images, de texte ou de son. Ils peuvent atteindre une précision de pointe. Les modèles sont entraînés en utilisant un grand ensemble d'architectures de données étiquetées et de réseau neuronal avec de nombreuses couches.[10]

- La plupart des méthodes d'apprentissage en profondeur utilisent des architectures de réseau neuronal.
- Le terme « profond » fait généralement référence au nombre de couches cachées dans le réseau neuronal, où les réseaux neuronaux traditionnels ne contiennent que 2-3 couches cachées, tandis que les réseaux profonds peuvent en avoir jusqu'à 150.
- Les modèles d'apprentissage en profondeur sont formés à l'aide de grands ensembles de données étiquetées et d'architectures de réseau neuronal qui apprennent les fonctionnalités directement à partir des données sans avoir besoin d'extraction manuelle des fonctionnalités.[10]

I) Réseaux de neurones récurrents le RNN est l'une des architectures réseau fondamentales à partir desquelles d'autres architectures d'apprentissage en profondeur sont construites.(figure 2.14),il peut avoir des connexions qui alimentent les couches précédentes (ou la même couche), ce qui permet aux RNN de conserver la mémoire des entrées précédentes et des problèmes de modèle dans le temps et il peut être déplié dans le temps et formé avec une rétropropagation standard ou en utilisant une rétropropagation dans le temps (BPTT).[13]

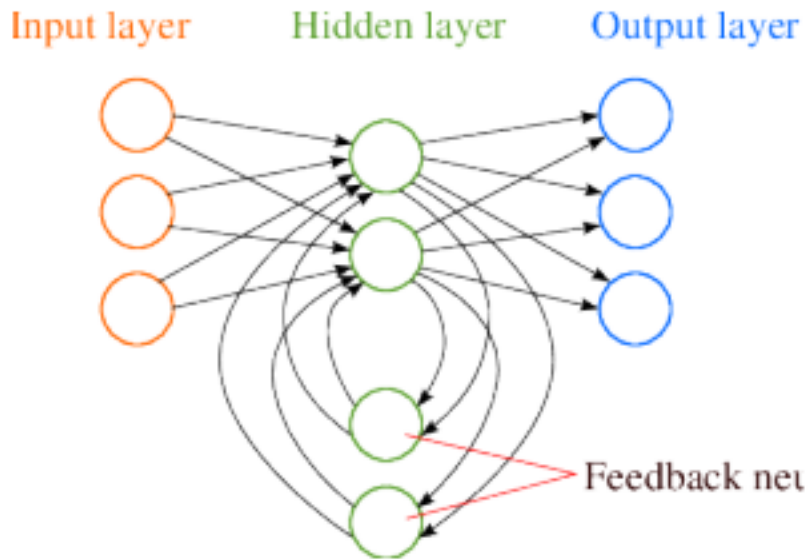


FIGURE 2.14 – RNN exemple

II) Les réseaux de neurones convolutionnels L'un des types les plus populaires de réseaux de neurones profonds est connu sous le nom de réseaux de neurones convolutifs (CNN ou ConvNet). [10]

Les CNN convolvent les entités apprises avec les données d'entrée et utilisent des couches convolutives 2D, ils éliminent le besoin d'extraction manuelle des caractéristiques pour classer les images, ils fonctionnent en extrayant des fonctionnalités directement à partir d'images. Ils sont appris pendant que le réseau s'entraîne sur une collection d'images.

L'extraction automatique des fonctionnalités rend les modèles d'apprentissage en profondeur très précis pour les tâches de vision par ordinateur telles que la classification d'objets.

Les couches sont organisées en 3 dimensions : largeur, hauteur et profondeur (trois canaux de couleur R, V, B). [2]

II.1) L'architecture de base de CNN

1. La partie Calques cachés / Extraction d'entités

Dans cette partie, le réseau effectuera une série de convolutions et d'opérations de regroupement jusqu'à ce que les entités soient détectées ou extraites.[2]

2. La partie Classification

Ici, les couches entièrement connectées serviront de classificateur au-dessus de ces caractéristiques extraites pour attribuer une probabilité pour que l'objet sur l'image soit ce que l'algorithme prédit qu'il est.[2]

et ca bien illustré dans la figure 2.15.

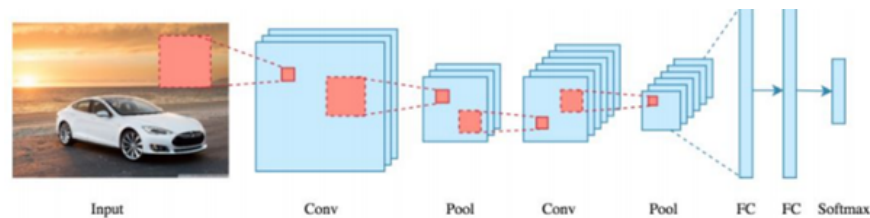


FIGURE 2.15 – Une architecture CNN

II.2) Architectures CNN populaires dans Deep Learning, un réseau de neurones convolutifs (CNN) est une classe de réseaux de neurones profonds, le plus souvent appliqué à l'analyse d'images visuelles. Les réseaux de neurones convolutifs sont des modèles pour la classification d'images, la segmentation, la détection d'objets et de nombreuses autres tâches de traitement d'images. Afin de nous lancer dans le domaine du traitement d'images ou d'améliorer la précision de prédiction des modèles CNN personnalisés, la connaissance de certaines des célèbres architectures CNN nous permettra d'atteindre le rythme dans ce monde compétitif.

Quelques exemples de tels modèles sont LeNet[26], AlexNet [28] et VGG Net [33]. D'autres alternatives et architectures avancées plus efficaces ont été proposées, notamment GoogLeNet [25], Residual Networks [24], DenseNet et FractalNet. Les éléments de construction de base (convolution et mise en commun) sont presque les mêmes dans ces architectures. Cependant, certaines différences topologiques sont observées dans les architectures modernes d'apprentissage en profondeur. Parmi les nombreuses architectures DCNN, AlexNet, VGG Net, GoogLeNet, DenseNet et FractalNet sont généralement considérés comme les architectures les plus populaires en raison de leurs performances de pointe sur différents benchmarks pour les tâches de reconnaissance d'objets.

II.2.1) L'architecture LeNet a été introduit par Yan LeCun pour la reconnaissance des chiffres.

La configuration de base du LeNet-5 est : 2 couches de convolutions (conv), 2 couches de sous-échantillonnage, 2 couches entièrement connectées et une couche de sortie avec la connexion gaussienne. Le nombre total de poids et de multiplication et d'accumulation (MAC) est de 431k et 2,3M respectivement (figure 2.16).[26]

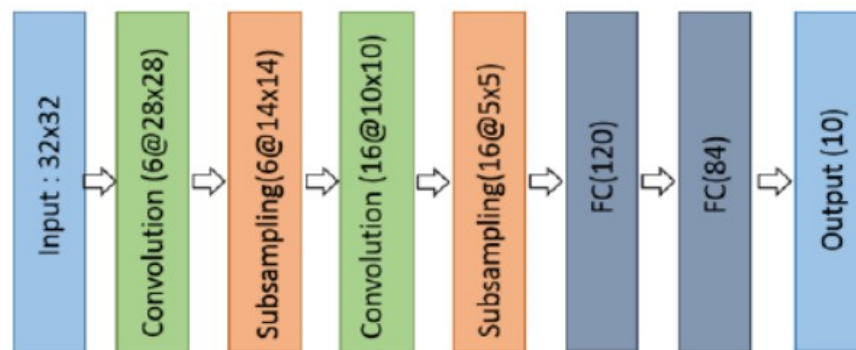


FIGURE 2.16 – Architecture du modèle LeNet

II.2.2) L'architecture AlexNet est illustrée dans la figure ci-dessous. La première couche convolutive effectue un pool de convolution (conv) max (MXP) avec normalisation de réponse locale (LRN) où 96 filtres récepteurs différents sont utilisés de taille 11×11 . Les opérations de pooling max sont effectuées avec des filtres 3×3 avec une taille de foulée de 2. Les mêmes opérations sont effectuées dans la deuxième couche avec des filtres 5×5 . Des filtres 3×3 sont utilisés dans les troisième, quatrième et cinquième couches convolutives avec respectivement 384, 384 et 296 cartes de caractéristiques. Deux couches entièrement connectées (FC) sont utilisées avec une suppression suivie d'une couche Softmax à la fin. AlexNet a été le premier à montrer que l'apprentissage en profondeur était efficace dans les tâches de vision par ordinateur (figure 2.17). [28]

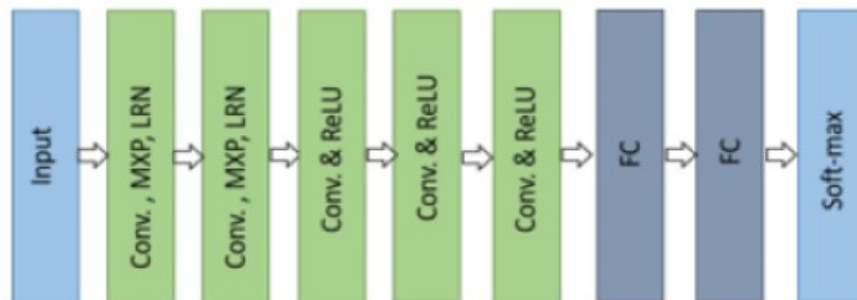


FIGURE 2.17 – Architecture du modèle AlexNet

II.2.3) L'architecture VGGNet se compose de deux couches convolutives qui utilisent toutes deux la fonction d'activation ReLU. Après la fonction d'activation, il y a une seule couche de pooling max et plusieurs couches entièrement connectées utilisant également une fonction d'activation ReLU. La dernière couche du modèle est une couche Softmax pour la classification (figure 2.18). [33]

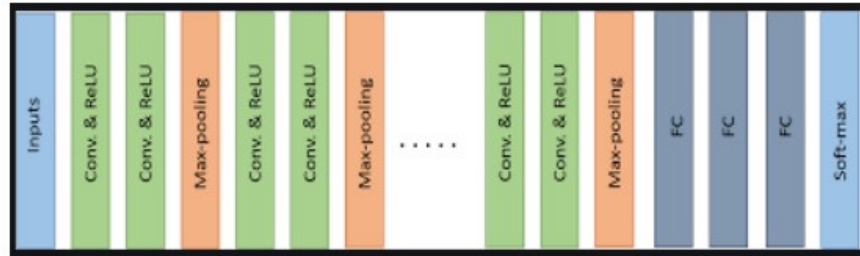


FIGURE 2.18 – Architecture du modèle VGGNet

II.2.4) L'architecture GoogLeNet a amélioré la précision de la reconnaissance de pointe en utilisant une pile de couches Inception (figure 2.19) qui avaient des champs réceptifs variables, qui ont été créés par différentes tailles de noyau. Ces noyaux permettaient de réduire la dimensionnalité avant les couches coûteuses en calcul. [25]

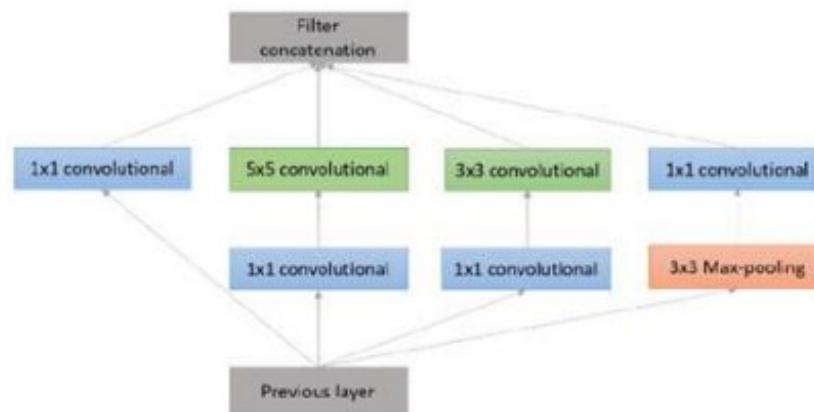


FIGURE 2.19 – Couche Inception avec réduction de dimension

2.6 Travaux relatifs et Synthèse

[31] : ce travail a proposé un modèle pour fournir des diagnostics précis pour la classification binaire (COVID vs No-COVID) et la classification multi-classes (COVID vs No-COVID vs Pneu-

monie). ce modèle a produit une précision de classification de 98,08 pour les classes binaires et de 87,02 pour les cas multi-classes.

[35] :ce travail a développé une architecture personnalisée appelée architecture CovidNet pour la classification des patients Covid-19, des patients sains et des patients atteints de pneumonie. A démontré une précision de classification de 93.3 - un résultat qui surpasse les tests de laboratoire.

[27] :Ce travail a développé une architecture personnalisée appelée coroNet pour la classification des patients « COVID vs Pneumonia bacterial vs pneumonia viral vs normal » Ce modèle a produit Precision (90) Recall (89.92) Specificity (96.4) F-measure (89.8) Accuracy (89.6)

Dans le tableau 2.2, nous avons résumé une étude comparative entre plusieurs architectures de Deep Learning de la section ci-dessus des travaux associés.

TABLE 2.2 – Comparaison des travaux relatifs

	Article	Class	Accuracy	Sensitivity	Precision	specificity
DarkCovidNet	[31]	3	87,02	85.35	89.96	92.18
COVID Net	[35]	3	93.3	93.3	Nan	Nan
Coronet	[27]	4	89.6	Nan	90	96.4

2.7 Conclusion

Dans ce chapitre nous avons présenté quelques éléments d'anatomie pulmonaire ,ensuite, les principes physique des rayons x,ensuit ,les méthodes de classification conventionnelles et les méthodes de classification basée sur l'apprentissage profond, ainsi, quelques CNN populaires. Pour finir le chapitre avec les traveaux relatifs.

Chapitre 3

Conception de l'architecture pour la classification des maladies pulmonaires

3.1 Introduction

Dans ce chapitre, nous avons discuté de notre architecture de réseau de neurones convolutifs pour classer les maladies pulmonaires.

Tout d'abord, nous présentons notre architecture générale de notre modèle de classification basé sur les réseaux CNN. Ensuite dans les sections suivantes nous allons présenter le fonctionnement détaillé de notre modèle.

3.2 Architecture générale

Notre modèle de classification (figure 3.1) utilise en entrées différentes images Rayon-X et permet d'identifier plusieurs maladies pulmonaires (Covid-19, Pneumonie virale/bactérienne et la Tuberculose). Notre modèle que nous décrivons en détail dans la section suivante doit répondre aux différents soucis d'erreurs de diagnostic relatifs aux interprétations de radiographies thoraciques effectuées quotidiennement dans les hôpitaux sous diverses conditions médicales vitales des patients.

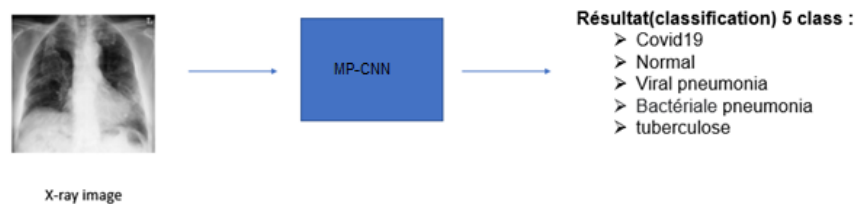


FIGURE 3.1 – l'architecture générale

3.3 Architecture détaillée

Afin de construire un modèle d'apprentissage en profondeur pour la classification avec CNN qui peut classer avec précision, nous proposons un MP-CNN présenté, dans la figure 3.2 .

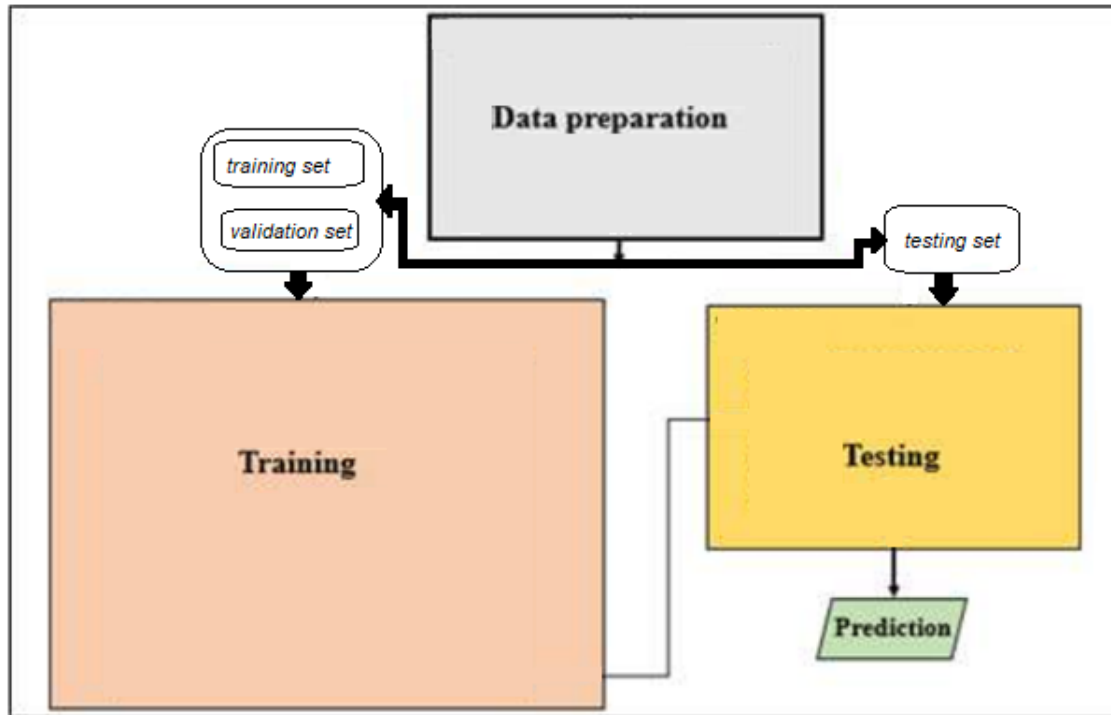


FIGURE 3.2 – architecture de MP-CNN

3.3.1 Préparation des données

La préparation des données est la première étape à appliquer avant leur utilisation au CNN. Elle est divisée en trois étapes.

3.3.1.1 Les bases de données utilisées

Nous avons utilisé trois bases de données publiques de kaggle qui concerne la covid-19, la pneumonie virale, la pneumonie bactérienne, la normale et la tuberculose. Nous avons utilisé

une image radiographique pour chaque cas de patient :

[7] COVID (3616 x-ray images), Viral Pneumonia (1345 x-ray images), Normal (4000 x-ray images).

[6] tuberculosis (3500 x-ray).

[8] COVID-19 (1281 x-ray images), Pneumonia-Bacterial (3001 x-ray images), Pneumonia-Viral (1656 x-ray images).

3.3.1.2 Prétraitement

Avant de fournir les données au CNN, il est nécessaire d'appliquer certaines techniques de traitement d'image afin de préparer les données et de construire un modèle d'apprentissage profond efficace. Ces techniques dépendent des bases de données utilisées. Dans notre travail, l'étape de prétraitement consiste en une opérations : le redimensionnement puisqu'on a utilisé différentes bases de données avec différents formats et ceci pour construire un modèle d'apprentissage puissant (Figure 3.3).

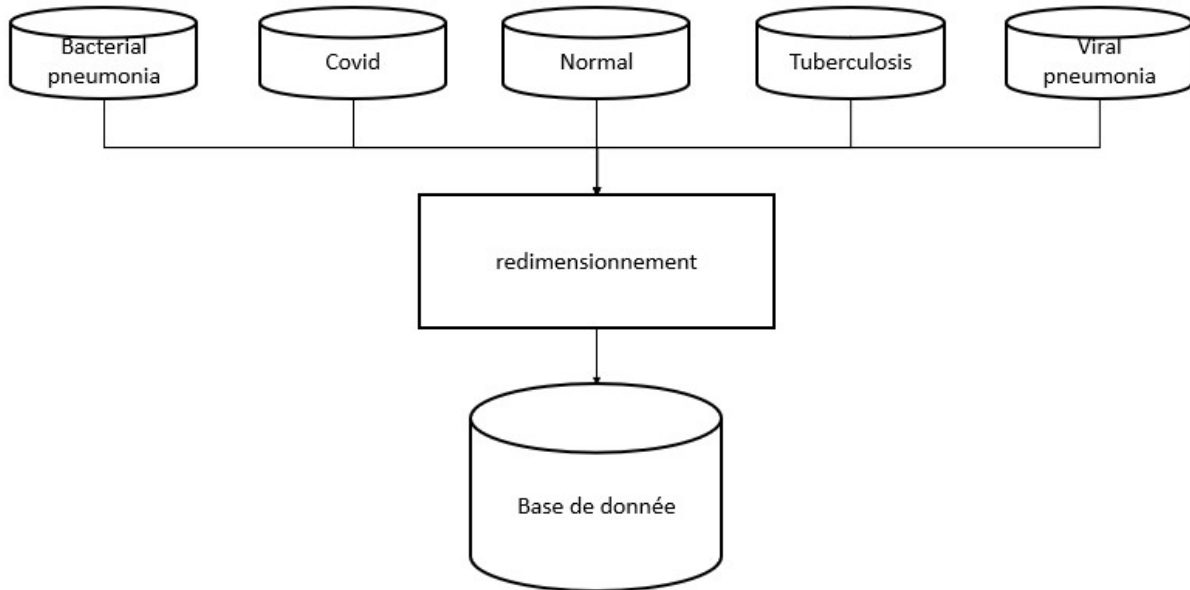


FIGURE 3.3 – Opération de prétraitement

3.3.1.3 Division de la base de données

La stratégie de fractionnement la plus fréquente de l'ensemble de données, en apprentissage profond, consiste à diviser l'ensemble de données en ensembles de formation, de validation et de test. Le rapport optimal d'échantillons distribués dans chaque ensemble varie pour chaque problème. Cependant, en règle générale, une répartition de 70% de train, 20% de validation et 10% de division de test est couramment utilisée (figure 3.4).

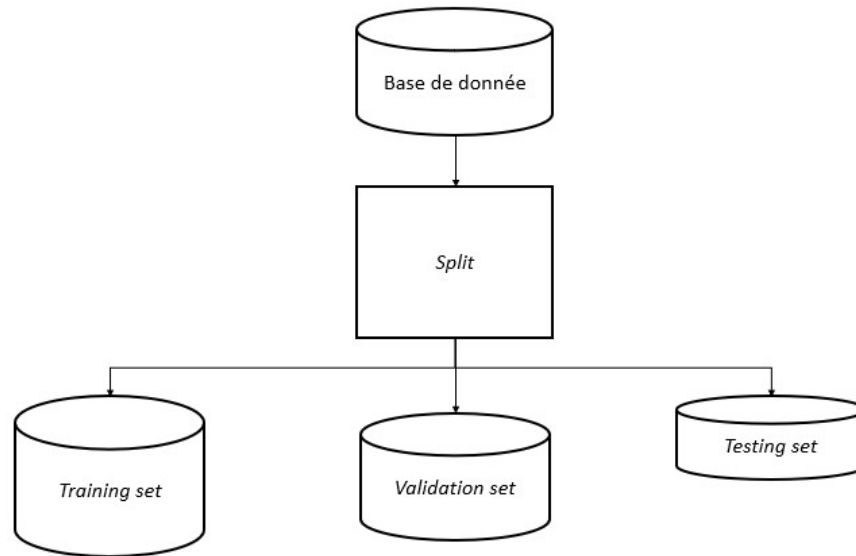


FIGURE 3.4 – Division de l'ensemble de données prétraitées

Pour concevoir un réseau approprié tel que la taille d'image d'entrée, le nombre de couches et les attributs convolutifs, nous pouvons essayer les réseaux populaires et performants (par exemple les réseaux AlexNet) avec le framework DL open-source populaire. Cependant, trouver la configuration de réseau la plus réussie pour un problème donné peut être un défi difficile étant donné le nombre total de configurations possibles dont on pourrait se prévaloir et également la quantité de temps concomitante pour la formation et le test du réseau.

3.3.2 *Training* le modèle de l'apprentissage

La stratégie d'apprentissage utilise une approche de descente de gradient stochastique, avec une taille de lot fixe : (a) une série de correctifs d'image corrigés moyens sont introduits dans le réseau sur une série d'époques, (b) une dérivée d'erreur calculée, et (c) rétro-propagée à travers le réseau en mettant à jour les poids du réseau. Le taux d'apprentissage est recuit au cours du

temps de sorte qu'un minimum local soit atteint. Les poids appris résultants (c'est-à-dire le modèle) sont stockés pour être utilisés plus tard au moment du test.

L'objectif de la composante de train est de déduire une fonction qui peut bien mapper les images d'entrée à leurs étiquettes appropriées (pneumonie bactérienne, covid, normal, tuberculose ou pneumonie virale) en utilisant l'ensemble de training. Le processus d'apprentissage est basé sur un réseau neuronal convolutif qui est un réseau neuronal profond d'apprentissage supervisé, et sa structure(Figure 3.5) comprend des couches de convolution(layers of convolution), des couches de mise en commun(layers of pooling) et des couches entièrement connectées, parmi lesquelles la couche de convolution et la couche de mise en commun sont des éléments essentiels pour réaliser l'extraction de caractéristiques du MP-CNN. La structure est d'abord agencée en alternance par couche de convolution et couche de regroupement pour réaliser l'extraction et la cartographie des caractéristiques locales à partir de l'ensemble d'apprentissage, puis successivement agencée par plusieurs couches entièrement connectées. La dernière couche du réseau classe ces cartes d'entités avec une méthode soft-max et donne en sortie des scores correspondant à la «probabilité» que chaque image appartienne à chacune des classes apprises. L'initialisation des paramètres CNN a un impact important sur la précision globale de la prédiction. Nous avons initialisé empiriquement les hyper-paramètres et les avons affinés au cours du processus de training . Les détails des hyperparamètres utilisés dans cette étude sont présentés dans le tableau 3.1

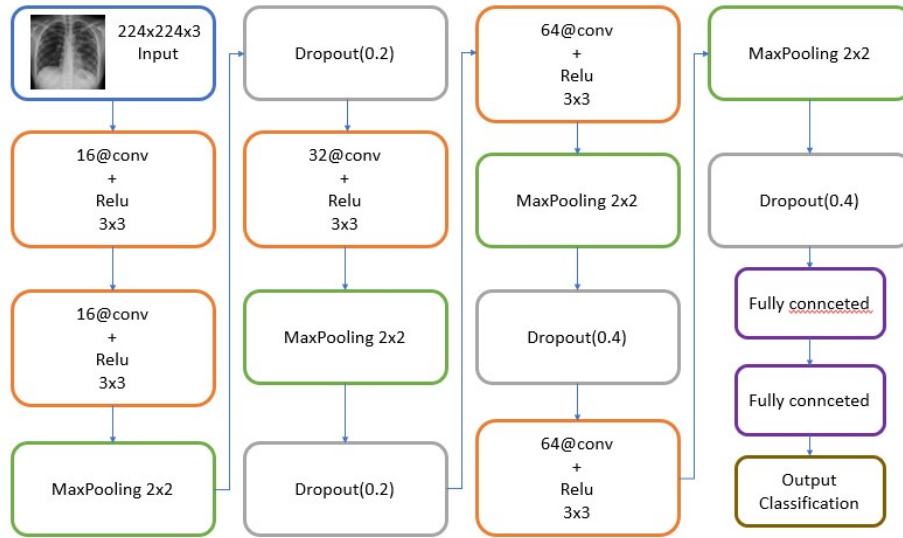


FIGURE 3.5 – MP-CNN architecture

TABLE 3.1 – Les hyperparamètres utilisés dans le réseau de neurones à convolution

	Paramètres	Hyperparamètres
Convolutional layer	noyaux	Taille des noyaux, nombre de noyaux, stride, padding, fonction d'activation
Pooling layer	—	Méthodes de Pooling , taille du filtre, stride, padding
Fully connected layer	Nombre de poids, fonction d'activation	Nombre de poids, fonction d'activation

Un ensemble de validation est utilisé pour tester notre modèle formé de manière intermédiaire . La précision ou l'erreur de la sortie d'apprentissage de l'ensemble de validation a été utilisée comme paramètre de rétroaction pour régler les hyper-paramètres, y compris le nombre

de cartes d'entités, la taille du noyau des couches convolutives et la taille du filtre des couches de regroupement dans le réseau. . Tout au long des cycles de training-validation, nous avons réglé les hyper-paramètres et mis à jour l'ensemble de données de training jusqu'à ce que la précision de l'apprentissage atteigne un niveau satisfaisant et par la suite on l'obtient notre architecture CNN.

3.3.3 la phase de test du modèle

En soumettant une image au réseau, de la même taille utilisée du training, nous obtenons une prédiction de classe à partir du modèle appris.

Un ensemble de test utilisé une seule fois à la fin du projet afin d'évaluer les performances du modèle final qui est affiné et sélectionné sur le processus d'apprentissage avec des ensembles d'apprentissage et de validation.

Les prédictions d'ensemble sur l'ensemble de tests ont été analysées pour les vrais positifs (V P), les vrais négatifs (V N), les faux positifs (F P) et les faux négatifs (F N). Ces paramètres ont été utilisés pour calculer une variété de mesures de performance, notamment " *precision* " (PR), " *sensitivity* " (SE), " *accuracy* " (ACC), " *F1-score* " (F1), " *specificity* " (SP) selon les équations suivantes :

$$1) PR = \frac{TP}{TP + FP}$$

$$2) SE = \frac{TP}{TP + FN}$$

$$3) ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$4) SP = \frac{TN}{TN + FP}$$

$$5) F1 = \frac{2TP}{2TP + FP + FN}$$

3.4 Conclusion

Dans ce chapitre, nous avons présenté notre méthodologie proposée pour concevoir un modèle de classification, en utilisant le Deep Learning avec un réseau de neurones convolutifs (CNN), est bien détaillé. Les environnements et outils de développement utilisés dans la réalisation de la méthodologie proposée seront décrits dans le chapitre suivant

Chapitre 4

Implémentation de l'architecture MP-CNN

4.1 Introduction

Dans ce chapitre, nous présentons le développement d'une conception détaillée qui apporte notre méthodologie pour concevoir un système d'apprentissage en profondeur pour la classification des maladies pulmonaires avec CNN (MP-CNN), proposé au chapitre 3.

4.2 Environnements et outils de développement

Google Colab La formation d'un modèle d'apprentissage en profondeur peut nécessiter une charge de travail importante du processeur / processeur graphique, c'est pourquoi nous avons utilisé la plate-forme cloud Google Colab pour cette tâche. Collaboratory est un projet de recherche Google créé pour aider à



FIGURE 4.1 – Google Colab

diffuser l'enseignement et la recherche sur l'apprentissage automatique. Il s'agit d'un environnement de notebook Jupyter qui ne nécessite aucune configuration pour s'utiliser et s'exécute

entièrement sur le cloud.

PyCharm PyCharm est un environnement de développement intégré utilisé pour programmer en Python . Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Développé par l'entreprise tchèque JetBrains, c'est un logiciel

multi-plateforme qui fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache.



FIGURE 4.2 – pycharm

Python Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs.

Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la

réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.



FIGURE 4.3 – python

TensorFlow TensorFlow est un framework de programmation pour les calculs numériques open source Publié par Google en novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning et donc les réseaux de neurones. Son nom est notamment inspiré du fait que



FIGURE 4.4 – tensorflow

les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow : Gmail, Google Photos, Reconnaissance de voix.

Keras Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur Google. En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow. Chollet a expliqué que Keras a été conçue comme une interface plutôt que comme un cadre d'apprentissage end to end. Il présente un ensemble d'abstractions de niveau supérieur et plus intuitif qui facilitent la configuration des réseaux neuronaux indépendamment de la bibliothèque informatique de backend. Microsoft travaille également à ajouter un backend CNTK à Keras aussi.

NumPy NumPy est une bibliothèque Python utilisée pour travailler avec des tableaux. Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices. NumPy a été créé en 2005 par Travis Oliphant. C'est un projet open source et vous pouvez l'utiliser librement.

Matplotlib Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques⁵. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD. Sa version stable actuelle (la 2.0.1 en 2017) est compatible avec la version 3 de Python.

PyQt PyQt est un module qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en python. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code python d'interfaces graphiques.

CSS Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

4.3 Développement de le back-end

Afin de mettre en œuvre notre modèle CNN d'apprentissage en profondeur, nous avons utilisé le langage Python avec des API et des bibliothèques bien connues travaillant sur l'apprentissage en profondeur (MP-CNN). Keras est l'une des bibliothèques Python les plus puissantes et les plus faciles à utiliser pour développer et évaluer des modèles d'apprentissage en profondeur ; Il enveloppe l'efficace bibliothèque de calcul numérique TensorFlow. Les étapes suivantes présentent notre implémentation du modèle CNN d'apprentissage en profondeur avec Python et Keras.

4.3.1 Préparation des données

Après avoir recherché et téléchargé l'ensemble de données, tout d'abord, nous avons appliqué un redimensionnement sur chaque image de l'ensemble de données (listing 4.1) pour obtenir un ensemble de données qui est divisé en trois ensembles distincts : training, validation et test, en utilisant une répartition de 60% de training, 30% de validation et 10% test. Le rapport d'échantillons distribués dans chaque ensemble est déterminé par des expériences. Nous avons appliqué `split folders.ratio`, en tant que module Python, tel qu'il est présenté dans la listing 4.2 et décrit dans le tableau 4.1.

```
1 f = r'/content/all data/Tuberculosis'  
2 for file in os.listdir(f):  
3     f_img = f+"/"+file  
4     img = Image.open(f_img)  
5     img = img.resize((224,224))  
6     img.save(f_img)  
7
```

Listing 4.1 – redimensionnement de la base de données

1

```

2 import splitfolders
3
4 input_folder="/content/all_data"
5 output="/content/datasetout1"
6 splitfolders.ratio(input_folder , output , seed=1337 , ratio=(.6 , .3 , .1))

```

Listing 4.2 – Diviser la base de données avec un ratio

TABLE 4.1 – Diviser avec un ratio

fonction	Arguments
Split folder.ratio()	<ul style="list-style-type: none"> — input folder : chemin d'accès à l'ensemble de données — seed : définir la valeur de départ pour mélanger les éléments, la valeur par défaut est 1337. — ratio : pourcentage de division (.6,.3,.1) — Output folder :chemin vers l'ensemble de données final

4.3.2 Réalisation du modèle MP-CNN

4.3.2.1 Importer des bibliothèques et des modules

Pour construire des réseaux de neurones profonds avec Keras, nous importons d'abord les différentes bibliothèques et modules tels qu'ils sont présentés dans le Listing 4.3 et décrits dans le Tableau 4.2 suivant.

```

1 import keras , os
2 from keras.models import Sequential
3 from keras.layers import Dense , Conv2D , MaxPool2D , Flatten , Dropout
4 from keras.preprocessing.image import ImageDataGenerator

```

```
5 import numpy as np
6 from keras.optimizers import Adam
7 from keras.callbacks import ModelCheckpoint, EarlyStopping
```

Listing 4.3 – Importer des bibliothèques et des modules

TABLE 4.2 – Description des bibliothèques et modules utilisés

Bibliothèques/modules	Descriptions
Numpy	Est la bibliothèque de base pour le calcul scientifique en Python. Il fournit un objet tableau multidimensionnel hautes performances et des outils pour travailler avec ces tableaux.
ImageDataGenerator	Est utilisé pour charger notre train, valider les ensembles de données en mémoire et générer des lots de tensor . données d'image avec augmentation des données en temps réel.
ImageDataGenerator	Pour créer des modèles d'apprentissage en profondeur où une instance de la classe Sequential est créée et des couches de modèle y sont créées et ajoutées. Un modèle séquentiel est approprié pour une pile simple de couches où chaque couche a exactement un tenseur d'entrée et un tenseur de sortie.
Dense	Est utilisé pour instancier une couche dense.
Dropout	Applique un décrochage à l'entrée. L'abandon consiste à définir de manière aléatoire un taux de fraction d'unités d'entrée sur 0 à chaque mise à jour pendant le temps d'entraînement, ce qui permet d'éviter le surapprentissage
Activation	Permet d'ajouter une fonction d'activation à la séquence de couches.
Flatten	Convertit la carte d'entités regroupées en une seule colonne qui est transmise à la couche entièrement connectée.
Conv2D	Crée un noyau de convolution qui est convolué avec l'entrée de couche pour produire un tenseur de sorties
Maxpooling2D	Sous-échantillonné la représentation en entrée en prenant la valeur maximale sur la fenêtre définie par la taille du pool pour chaque dimension le long de l'axe des caractéristiques
Adam	Optimiseur qui implémente l'algorithme Adam. L'optimisation d'Adam est une méthode de descente de gradient stochastique basée sur l'estimation adaptative des moments du premier et du second ordre
ModelCheckpoint	Rappel pour enregistrer le modèle Keras ou les poids du modèle à une certaine fréquence

4.3.2.2 Initialisation des paramètres MP-CNN

Avant de pouvoir commencer à entraîner le réseau, nous devons initialiser ses paramètres.

Les détails des hyperparamètres, utilisés dans cette étude, sont décrits dans le tableau 4.3.

Hyper-parameters	Descriptions	valeurs
largeur de l'image, hauteur de l'image	Taille de l'image d'entrée	224x224
batch size	est un hyperparamètre qui définit le nombre d'échantillons à traiter avant de mettre à jour les paramètres du modèle interne	<ul style="list-style-type: none"> — 50 pour train — 30 pour val
Class num	Le nombre des classes	5

TABLE 4.3 – Chargement de base de données

4.3.2.3 Chargement de la base de données

Nous avons utilisé la classe `ImageDataGenerator` pour charger nos ensembles de données de formation et de validation en mémoire. Dans cette étape, la taille cible, la taille du lot, les données de normalisation et le mode de classe sont définis. Ensuite, un itérateur est nécessaire pour obtenir progressivement des images pour un seul ensemble de données. Pour cela, il faut appeler la fonction `flow from directory()` et spécifier le répertoire de l'ensemble de données, comme le répertoire `train`, `test` ou `validation`. Vous trouverez plus de détails dans le Listing 4.4.

```
1 trdata = ImageDataGenerator()
```

```
2 traindata = trdata.flow_from_directory(directory="/content/datasetout1/  
   train",target_size=(224,224),batch_size = 50 )  
3 tsdata = ImageDataGenerator()  
4 testdata = tsdata.flow_from_directory(directory="/content/datasetout1/val"  
   , target_size=(224,224),batch_size = 30 )
```

Listing 4.4 – Chargement de base de données

4.3.2.4 Les paramètres d'apprentissage du réseau

Lors de la conception et de la configuration de notre modèle d'apprentissage en profondeur (CNN), nous avons tenté diverses configurations pour les réseaux des personnes et en utilisant des heuristiques. nous présentons les paramètres de l'architecture proposée pour CNN (Figure 4.5) .

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 224, 224, 16)       448
conv2d_1 (Conv2D)           (None, 224, 224, 16)       2320
max_pooling2d (MaxPooling2D) (None, 112, 112, 16)       0
dropout (Dropout)           (None, 112, 112, 16)       0
conv2d_2 (Conv2D)           (None, 112, 112, 32)       4640
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 32)       0
dropout_1 (Dropout)         (None, 56, 56, 32)       0
conv2d_3 (Conv2D)           (None, 56, 56, 64)        18496
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)       0
dropout_2 (Dropout)         (None, 28, 28, 64)       0
conv2d_4 (Conv2D)           (None, 28, 28, 64)        36928
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 64)       0
dropout_3 (Dropout)         (None, 14, 14, 64)       0
flatten (Flatten)           (None, 12544)              0
dense (Dense)               (None, 256)                3211520
dense_1 (Dense)             (None, 128)                32896
dense_2 (Dense)             (None, 5)                  645
-----
Total params: 3,307,893
Trainable params: 3,307,893
Non-trainable params: 0
-----

```

FIGURE 4.5 – MP-CNN paramètres

4.3.2.5 resultat de la prédiction

Les prédictions sur un modèle entraîné visent à faire une prédiction sur une image radiographique.

L'évaluation du modèle sur les données de test "test" est effectuée comme suit :

1. Comme pour la formation, nous utilisons ImageDataGenrator et la fonction flow from direc-

tory() pour passer notre ensemble de données de test.

2. Nous utilisons *load-model* pour charger le modèle formé.

3. nous utilisons le générateur *predict-generator* pour générer des prédictions 'prediction' à partir des échantillons d'entrée du générateur de données.

4. La matrice de confusion est utilisée pour évaluer la performance du classificateur (listing 4.5).

```
1 import numpy as np
2 from sklearn.metrics import accuracy_score, f1_score, precision_score,
   confusion_matrix
3 from mlxtend.plotting import plot_confusion_matrix
4 from sklearn.model_selection import StratifiedKFold
5 print("=====TEST RESULTS=====")
6 test_datagen = ImageDataGenerator()
7 test_generator = test_datagen.flow_from_directory(
8     'coronet/four_classes/train',
9     target_size=(224, 224),
10    batch_size=32,
11    class_mode=None,
12    shuffle=False)
13 predictions = model.predict(test_generator, verbose=1)
14 yPredictions = np.argmax(predictions, axis=1)
15 true_classes = test_generator.classes
16
17
18 CM = confusion_matrix(true_classes, yPredictions)
19 import matplotlib.pyplot as plt
20
21
22 fig, ax = plot_confusion_matrix(conf_mat=CM, figsize=(5, 5))
23 plt.show()
```

Listing 4.5 – matrice de confusion

4.4 Présentation de l'interface

Dans la partie interface du système, nous créons un design simple qui peut être facilement utilisé par le médecin, avec des couleurs et une structure simples.

Pour créer notre application de bureau, nous avons utilisé pyqt5 pour construire la structure de l'application, CSS (*Cascading Style Sheets*) nous donne un grand avantage sur la conception et le style de notre application pour une meilleure apparence et une meilleure organisation.

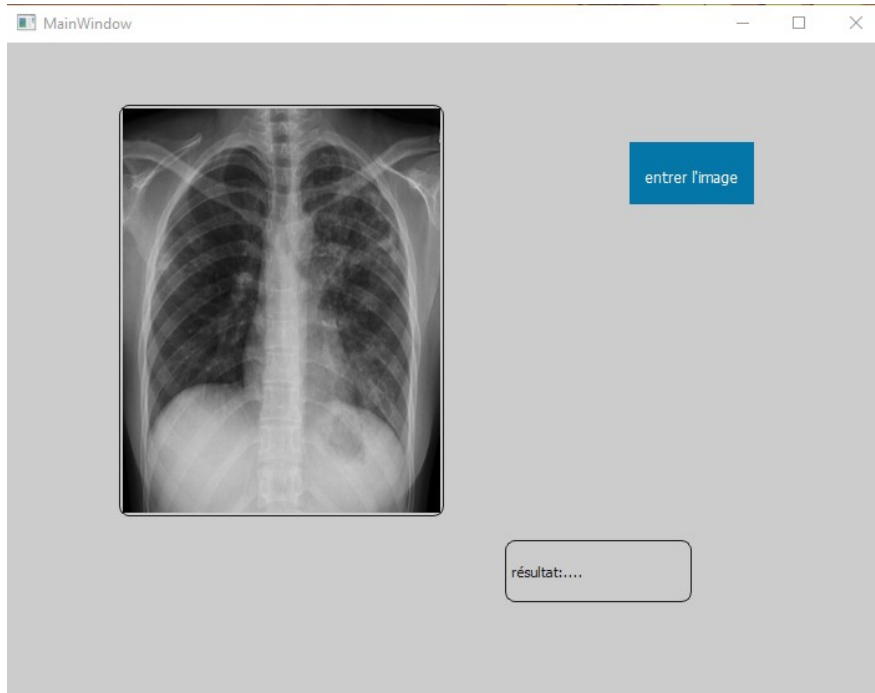


FIGURE 4.6 – Capture d'écran de l'interface

4.5 Résultats obtenus

4.5.1 Présentation des performances obtenu

Pour visualiser les performances de notre CNN de deep learning au fil du temps pendant *training*, nous avons créé :

- un graphe de "*accuracy*" sur l'ensemble de données *train* « *acc* » sur les époques d'entraînement (Figure 4.7).
- un graphe de "*loss*" sur l'ensemble de données *train* « *loss* » au cours des époques d'en-

entraînement(Figure 4.8).

- un graphe de "val-accuracy" sur l'ensemble de données de validation 'Val acc' sur les époques d'entraînement(Figure 4.7).
- un graphe de "val-loss" sur l'ensemble de données de validation «val-loss» sur les époques d'entraînement(Figure 4.8).

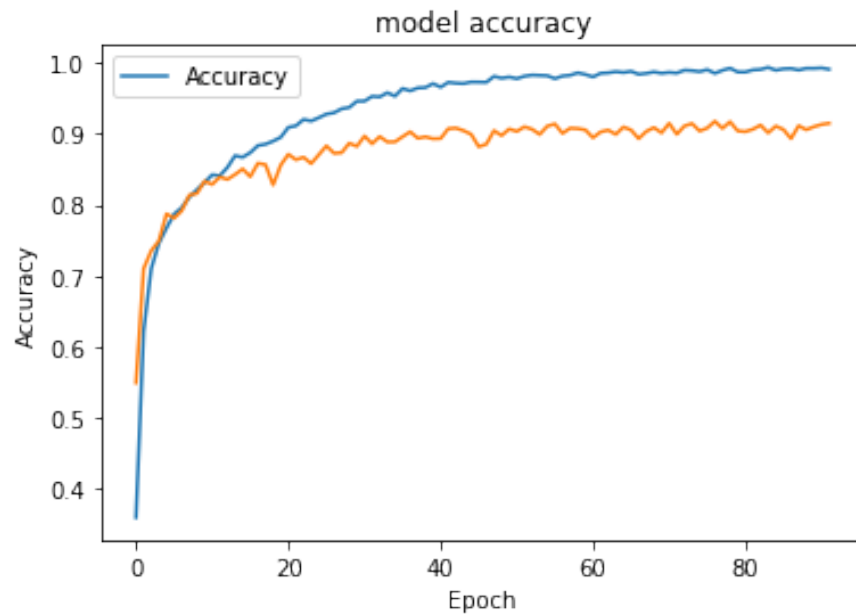


FIGURE 4.7 – accuracy

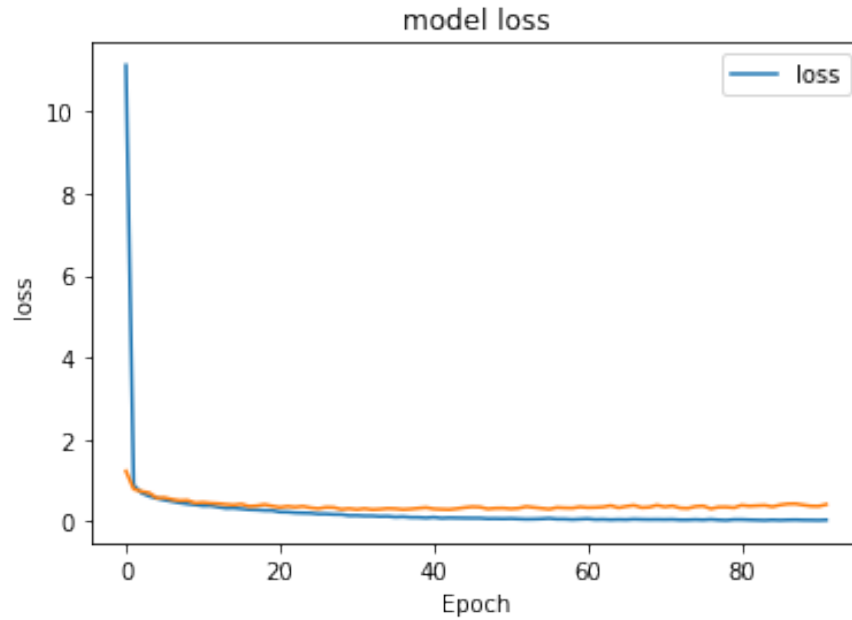


FIGURE 4.8 – loss

Dans l'apprentissage en profondeur, le point de contrôle sont les poids stockés du modèle lorsqu'il y a une amélioration de la précision de la classification sur l'ensemble de données de validation, ces poids peuvent être utilisés pour faire des prédictions telles quelles, ou utilisés comme base pour la formation continue. Dans notre cas, la meilleure valeur de validation accuracy(val-accuracy) est dans l'époque numéro 77 . A cette époque, les valeurs sont :

- Training loss 'loss' is 4.84% (Figure 4.9)
- Training accuracy 'accuracy' 98.36% (Figure 4.7)
- Validation loss 'Val loss 30.36% (Figure 4.10)
- Validation accuracy 'Val accuracy' 91.774% (Figure 4.8)

4.5.2 Évaluation du modèle sur les données de test

L'évaluation du MP-CNN sur les données de test 'test', vise à estimer sa précision de généralisation. Une matrice de confusion est une technique de mesure des performances utilisée pour évaluer les performances du modèle, construit dans la phase d'apprentissage, sur un ensemble de tests, en comparant les classes réelles et prédites. Il fournit une information plus détaillée des classifications correctes et incorrectes pour chaque classe. La matrice de confusion pour la prédiction de précision de 1843 images de l'ensemble de test (appartiennent à 5 classes) est présentée dans la figure 4.9 .

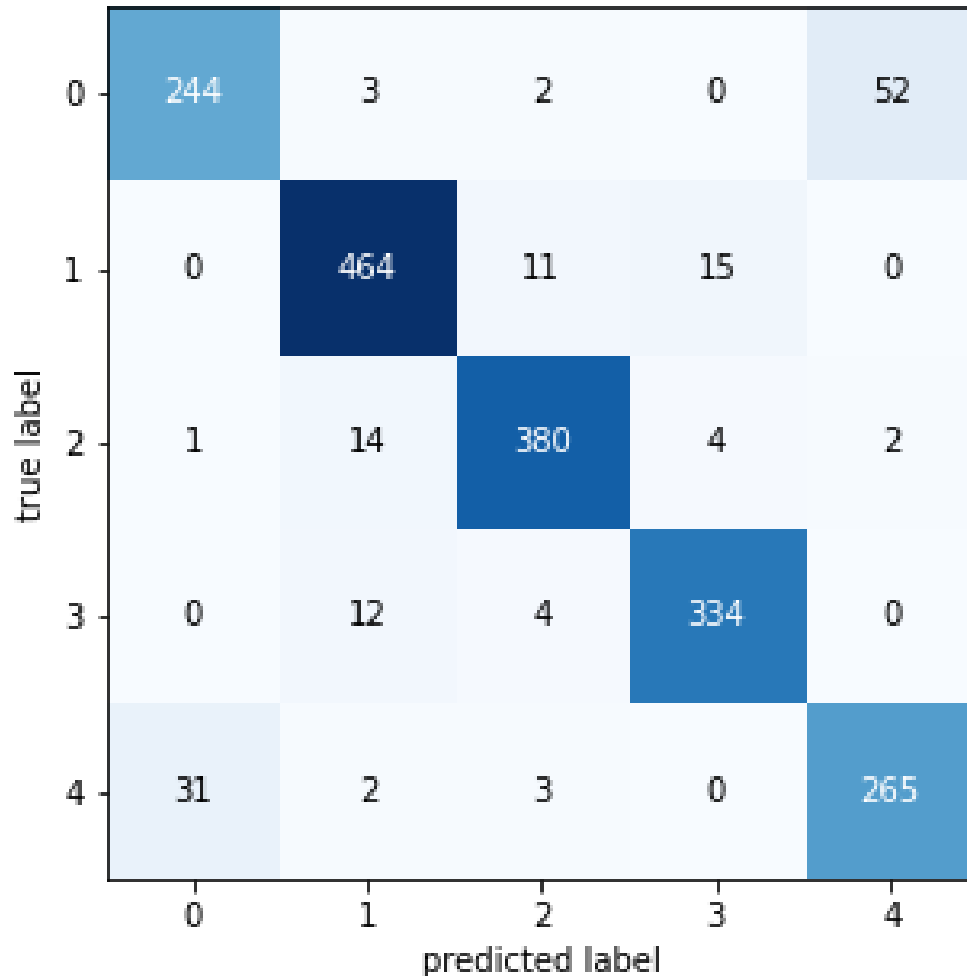


FIGURE 4.9 – matrice de confusion

(label 0="Bacterial pneumonia"-label 1="Covid"-label 2="normal"-label 3="tuberculosis"-label 4="Viral pneumonia")

Chaque ligne de la matrice de confusion représente les instances d'une classe réelle et chaque colonne représente les instances d'une classe prédite. Les éléments diagonaux représentent le nombre de points pour lesquels l'étiquette prédite est égale à l'étiquette vraie, tandis que tout ce qui se trouve en dehors de la diagonale a été mal étiqueté par le système.

Notre système MP-CNN peut prédire parfaitement : 244 images de pneumonie virale, 464 de

Covid, 380 de normale, 334 de tuberculose et 265 de pneumonie bactérienne.

Nous notons que le modèle fait beaucoup d'erreurs (52 fois en classant la pneumoni bactérienne et en la classant comme pneumonie viral,31 fois en classant la pneumoni viral et en la classant comme pneumonie bactérienne) en raison de leur similitude.

Les autres résultats du classement sont presque très bons.

- L'accuracy de model est : 91,53% .
- Precision de model est : 91.56% .
- F1 score de model est : 91.51% .

TABLE 4.4 – Tableau comparatif des travaux connexes

	Article	Class	Accuracy	Precision	specificity	f1 score
DarkCovidNet	[31]	3	87,02%	89.96%	92.18%	Nan
COVID Net	[35]	3	93.3%	Nan	Nan	Nan
Coronet	[27]	4	89.6 %	90%	96.4%	89.8%
MP-Net	ce tra- vail	5	91.53 %	91.56%	Nan	91.51%

Après la phase de test on a essayé de télécharger la base de données de l'article [27] (la base de données contient bacterial pneumonia,covid,normal,viral pneumonia) et essayer de la classer avec mon modèle et voici les résultats du classement (figure 4.10) :

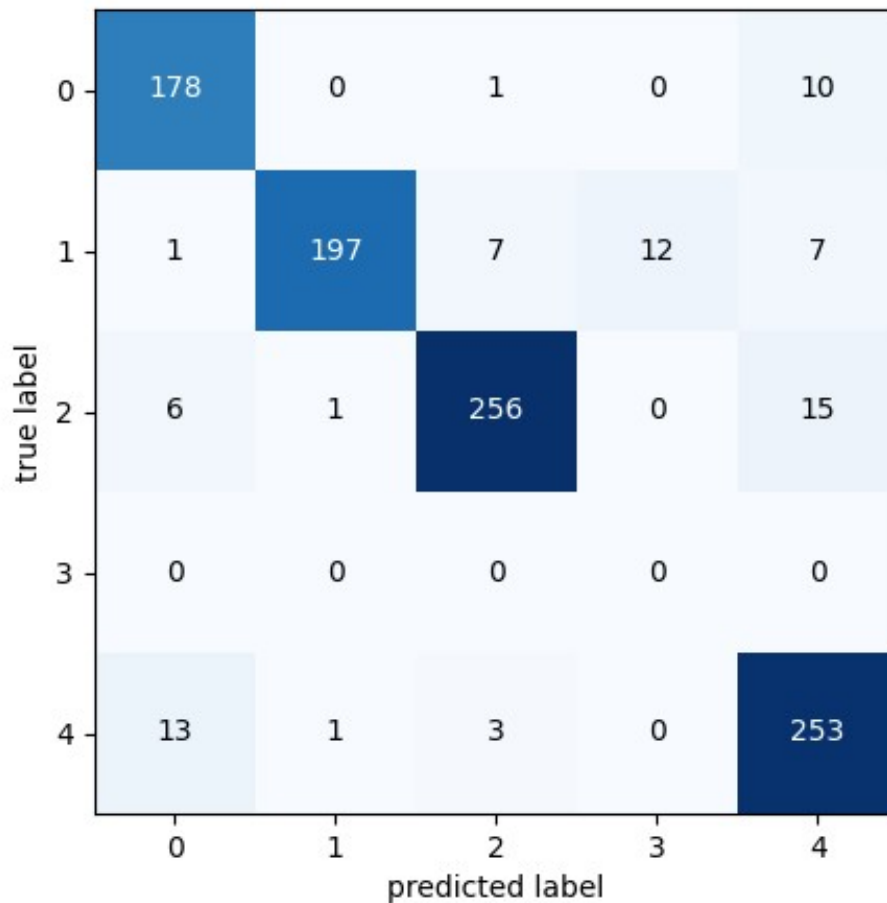


FIGURE 4.10 – matrice de confusion pour base de données de [27]

- L'accuracy est : 91.98% .
- Precision est : 93.43% .
- F1 score est : 92.59% .

Nous avons essayé de comparer les résultats que nous avons obtenus avec des travaux récents. Comme le montre le tableau 4.4, on montre qu'on a obtenu de meilleure .

le résultat de la classification dépend non seulement de la méthode de classification utilisée, mais aussi de la base de données et le traitement utilisée , du nombre d'images et du processus suivi. La bibliothèque d'images est sans aucun doute un facteur important affectant les résultats

de la classification.

Chapitre 5

Conclusion et perspectives

5.1 Conclusion

Dans ce travail, nous avons étudié l'utilisation des réseaux de neurones profonds, en particulier des réseaux de neurones convolutifs, pour le problème de la classification des images pulmonaires radiographiques.

Le modèle CNN d'apprentissage en profondeur proposé pour classer le modèle de la maladie pulmonaire (MP-CNN) excelle en apprenant les caractéristiques pertinentes directement à partir de l'image radiographique.

Après avoir entraîné avec succès le modèle, "accuracy" du test est calculée :

pour 1843 images appartenant à 5 classes nous avons atteint 91,53 % "testing accuracy" 91,56% precision.

De cette évaluation, nous pouvons conclure que :

- À mesure que le nombre d'échantillons augmente, les performances du modèle MP-CNN augmentent considérablement.
- L'erreur de classification entre la pneumonie virale et la classe de pneumonie bacté-

rienne peut être due aux similitudes entre elles.

5.2 Perspectives

Cependant, cette recherche pourrait être améliorée sur certains points que nous considérons comme des perspectives pour nos travaux.

- ajouter d'autres types de maladies pulmonaires au modèle MP-CNN.
- faire un modèle pour classifier les pourcentages de covid-19.
- Utilisation d'un autre ensemble de données pour la formation, la validation et les tests.
- Construire des modèles cnn personnalisés pour chaque "maladie et cas normaux" et après les avoir classifiés à partir du premier modèle qui est le modèle général, nous passons l'image pour le modèle personnalisé pour confirmer notre résultat et minimiser le défaut .

Bibliographie

- [1] X-rays. <https://medlineplus.gov/xrays.html>.
- [2] An intuitive guide to convolutional neural networks. <https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/>.
- [3] Algorithme k-moyennes. <http://www.ieee.ma/uaesb/pdf/Algo-k-Moyennes.pdf>, .
- [4] k-means. <https://analyticsinsights.io/k-means/>, .
- [5] Carlos gershenson. “artificial neural networks for beginners”. [https://datajobs.com/data-science-repo/Neural-Net-\[Carlos-Gershenson\].pdf](https://datajobs.com/data-science-repo/Neural-Net-[Carlos-Gershenson].pdf).
- [6] Tuberculosis (tb) chest x-ray database. <https://www.kaggle.com/tawsifurrahman/tuberculosis-tb-chest-xray-dataset>.
- [7] Covid-19 radiography database. <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.
- [8] Curated chest x-ray image dataset for covid-19. <https://www.kaggle.com/unaisait/curated-chest-xray-image-dataset-for-covid19?select=Curated+X-Ray+Dataset>.

-
- [9] Density-based spatial clustering of applications with noise (dbscan). <https://ml-explained.com/blog/dbscan-explained>.
- [10] What is deep learning? <https://www.mathworks.com/discovery/deep-learning.html>.
- [11] A beginner's guide to important topics in ai, machine learning, and deep learning. <http://www.cs.kumamoto-u.ac.jp/epslab/ICinPS/Lecture-2.pdf>.
- [12] Qu'est-ce qu'un arbre de décision? <https://www.lucidchart.com/pages/fr/arbre-de-decision>.
- [13] M. tim jones. deep learning architectures. <https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/>.
- [14] Classification non supervisée. https://veroniquetremblay.github.io/analyse_de_donnees_et_apprentissage_statistique_en_R/classification-non-supervisee.html.
- [15] Classification superviséeaperçu de quelques méthodes avec le logiciel r. <https://math.univ-angers.fr/~labatte/enseignement%20UFR/master%20MIM/classificationsupervisee.pdf>.
- [16] Support vector machines. <https://analyticsinsights.io/les-svm-support-vector-machine/>.
- [17] Scott J Adams, Robert DE Henderson, Xin Yi, and Paul Babyn. Artificial intelligence solutions for analysis of x-ray images. *Canadian Association of Radiologists Journal*, page 0846537120941671, 2020.

-
- [18] Kazuto Ashizawa, Takayuki Ishida, Heber MacMahon, Carl J Vyborny, Shigehiko Katsuragawa, and Kunio Doi. Artificial neural networks in chest radiography : application to the differential diagnosis of interstitial lung disease. *Academic radiology*, 6(1) :2–9, 1999.
- [19] Mariette Awad and Rahul Khanna. *Efficient learning machines : theories, concepts, and applications for engineers and system designers*. Springer nature, 2015.
- [20] Sylvain Caron and Louise Samson. La radiographie pulmonaire.
- [21] Bradley Efron. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352) :892–898, 1975.
- [22] Ali A El-Solh, Chiu-Bin Hsiao, Susan Goodnough, Joseph Serghani, and Brydon JB Grant. Predicting active pulmonary tuberculosis using an artificial neural network. *Chest*, 116(4) : 968–973, 1999.
- [23] Yicheng Fang, Huangqi Zhang, Jicheng Xie, Minjie Lin, Lingjun Ying, Peipei Pang, and Wenbin Ji. Sensitivity of chest ct for covid-19 : comparison to rt-pcr. *Radiology*, 296(2) :E115–E117, 2020.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [25] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

- [26] Mohammed Kayed, Ahmed Anter, and Hadeer Mohamed. Classification of garments from fashion mnist dataset using cnn lenet-5 architecture. In *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pages 238–243. IEEE, 2020.
- [27] Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasar Bhat. Coronet : A deep neural network for detection and diagnosis of covid-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, 196 :105581, 2020.
- [28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25 : 1097–1105, 2012.
- [29] Jean-Pierre Nakache and Josiane Confais. *Approche pragmatique de la classification : arbres hiérarchiques, partitionnements*. Editions Technip, 2004.
- [30] Quentin Nicard. Poumons anatomie. <https://www.passeportsante.net/fr/parties-corps/Fiche.aspx?doc=poumons>., 2016.
- [31] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine*, 121 :103792, 2020.
- [32] NJ Peacock, MG O’Mullane, R Barnsley, and M Tarbutt. Anticipated x-ray and vuv spectroscopic data from iterwith appropriate diagnostic instrumentation. *Canadian Journal of Physics*, 86(1) :277–284, 2008.

- [33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [34] R Stumm Von Bordwehr. A history of x-ray absorption fine structure. In *Annales de Physique*, volume 14, pages 377–465. EDP Sciences, 1989.
- [35] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net : A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1) :1–12, 2020.