

**ISGA Institut Supérieur d'Ingénierie & des Affaires**

##### Mémoire du Projet de Fin d’année

**Sous le thème :**

# Diagnostic des maladies pulmonaires, incluant le COVID-19, via analyse d'histogrammes et réseaux neuronaux sur radiographies

*Présenté par*

**Belmouss Badr Eddine - El Hamidi Maroua – Akarkab Maryam -**

**Ighidi Hakima - Ighidi Laila**

***Encadré par : M. Lahcen Aitibourek***

**Année universitaire 2023-2024**

**REMERCIEMENTS**

Nous souhaitons dédier ce projet de fin d'études à notre encadrant, Monsieur **Lahcen Aitibourek**. Son suivi attentif, ses conseils éclairés, et son soutien inébranlable ont été des atouts précieux tout au long de ce travail. Son expertise et sa patience ont été essentielles pour surmonter les défis rencontrés et nous permettre de progresser constamment. Nous le remercions sincèrement pour son dévouement et la confiance qu'il a placée en nous.

Nous tenons également à exprimer notre profonde gratitude pour la collaboration et l'engagement que chacun de nous a apporté à ce projet. Ensemble, nous avons réussi à mener à bien cette aventure académique, grâce à notre coopération, nos idées, et notre détermination commune.

**Résumé**

Le projet vise à développer et optimiser des réseaux neuronaux convolutifs (CNN) pour améliorer la détection des maladies pulmonaires, y compris le COVID-19, à partir d'histogrammes radiographiques. En utilisant des techniques avancées de prétraitement d'images, telles que la normalisation, la réduction du bruit, la segmentation des poumons et l'amélioration du contraste, nous améliorerons la qualité des données d'entrée. Nous explorerons différentes architectures de CNN, optimiserons les hyperparamètres et intégrerons des mécanismes d'attention et de régularisation pour éviter le surapprentissage.

Les modèles pré-entraînés et l'apprentissage par transfert seront également utilisés. Les résultats seront interprétés à l'aide de visualisations d'activations, de cartes de chaleur, et d'analyses d'erreurs pour garantir des prédictions explicables et compréhensibles. En fin de compte, cette recherche fournira des outils IA avancés pour une détection précoce et précise des maladies pulmonaires, améliorant ainsi les pratiques diagnostiques médicales.

Mots clés : apprentissage automatique, segmentation d’image, Intelligence artificielle, Modèles pré-entraînés.

# Table des figures

Figure 1.1 : Les poumons dans le système respiratoire Figure 1.2 : La radiographie thoracique

Figure 1.5 : Diagramme de cas utilisation.

Figure 2.1 : Nuage de mots des concepts et technologies utilisés dans ce projet.

Figure 2.2 : Intelligence artificielle, apprentissage automatique et science des données [1]. Figure 2.3 : Programme traditionnel et apprentissage automatique.

Figure 2.4 : Différents composants de l’intelligence artificielle. Figure 2.5 : Cycle de vie de l'apprentissage automatique.

Figure 2.6 : Image couleur, son repère et un extrait de pixels. Figure 2.7 : Image en niveaux de matrice à deux dimensions

Figure 2.8 : Image en niveaux RGB combinaison de rouge, vert et bleu. Figure 2.9 : Représentation des pixels

Figure 2.10 : histogramme tassé sur la gauche (sombre).

Figure 2.11 : Amélioration de la Qualité des Images avec Égalisation d'Histogramme du Contraste Figure 2.12 : Structure générale d’un réseau de neurones convolutifs

Figure 2.13 : Architecture CNN

Figure 3.1 : Quelques images de l’ensemble de données utilisées, (a) cas positifs au COVID-19, (b) Normale, (c) Viral Pneumonie

Figure 3.2 : Étapes de Prétraitement et d'Analyse d'Images Radiographiques Pulmonaires pour le Diagnostic du COVID-19 via un Réseau Neuronal Convolutif

Figure 3.3 : Répartitions de l’ensemble des images de la base de données par catégories

Figure 3.4 : Répartition des Données en Groupes d'Entraînement et de Test pour la Modélisation du Diagnostic COVID-19

Figure 3.5 : Amélioration de la Qualité des Images par Égalisation d'Histogramme pour une Répartition Optimale du Contraste

Figure 3.6 : Logo python

Figure 3.7 : Navigateur Anaconda

Figure 3.8 : L'environnement de développement scientifique Python Figure 3.9 : Logo Keras

Figure 3.10 : Logo Jupyter Notebook Figure 3.11 : Logo TensorFlow Figure 3.12 : Logo NumPy

Figure 3.13 : Logo Pandas Figure 3.14 : Logo Matplotlib Figure 3.15 : Logo Glob Figure 3.16 : Logo seaborn

Figure 4.1 : Résultat d’application de la fonction Binarypattern du sur une image Figure 4.2 : Résultat d’application de l’algorithme Canny du sur une image Figure 4.3 : Image origine et Image avec le filtre de Sobel XY

Figure 4.4 : Image origine et Image après l'application de la fonction he\_hsv

Figure 4.5 : Image origine et Image après la transformation par application de Canny Figure 4.6 : Histogramme et CDF d’une image origine.

Figure 4.7 : Histogramme et CDF d’une image après l’égalisation.

Figure 4.8 : Histogramme cumulatif avant égalisation et Histogramme cumulatif après égalisation Figure 4.9 : début de l’entrainement du model

Figure 4.10 : validation du model par 20% data Figure 4.11 : performances ConvNet rapides.

Figure 4.12 : la courbes de ROC CNN utiliser. Figure 4.13 : Premier interface de l’application. Figure 4.14 : Sélection d'une Image.

Figure 4.15 : Sélection d'une Image Covid-19. Figure 4.16 : Sélection d'une Image Normal.

Figure 4.17 : Sélection d'une Image Virus Pulmonaire. Figure 4.18 : Sélection d'une Image Inconnu.

# Liste des tableaux

Tableau 1.1 : Accueil Diagramme de Gantt Tableau 1.2 : Taches Diagramme de Gantt Tableau 1.3 : Diagramme de Gantt

Tableau 1.4 : Ressources Diagramme de Gantt

Tableau 3.1 : Caractéristique de matériel de programmation

Tableau 4.1 : résumé d'un modèle de réseau de neurones convolutifs (CNN) créé avec Keras

# Listings

Listing 4.1 – code traitement d'image utilisée pour la reconnaissance de motifs. Listing 4.2 – code d’application de l’algorithme Canny.

Listing 4.3 code d’application du filtre Sobel.

Listing 4.4 code de comparaison visuellement l'image originale avec les bords obtenus. Listing 4.3 code d’application du filtre Sobel.

Listing 4.4 code de comparaison visuellement l'image originale avec les bords obtenus.

Listing 4.5 code de création de figure de 10x10 pouces avec l’affichage d’image transformée par l'algorithme Canny.

Listing 4.6 code d’utilisation de OpenCV et Matplotlib pour afficher l'histogramme d'une image Listing 4.7 code histogramme et sa fonction de distribution cumulée.

Listing 4.8 code d’utilisant TensorFlow et Keras pour l’entrainement du modèle. Listing 4.9 code pour la classification d'images en utilisant TensorFlow et Keras.

Listing 4.10 code de sauvegarder le modèle entraîné dans deux formats différents HDF5 et Keras. Listing 4.11 code entraîne un modèle de réseau de neurones avec des callbacks.

Listing 4.12 code validation un modèle de réseau de neurones.

Listing 4.13 code Matplotlib pour visualiser les performances du modèle de réseau de neurones. Listing 4.14 code Matrice de confusion.

Listing 4.14 code de création d’une interface graphique.

Listing 4.15 exécute une prédiction COVID-19 à partir d'images radiographiques en utilisant Tkinter.

# Liste des abréviations

**ADN** : acide désoxyribonucléique

**IA** : Intelligence artificielle **TDM** : Tomodensitométrie **QU** : Qatar Université

**AI** : Intelligence Artificielle **ML** : Apprentissage Machine **DL** : Apprentissage Profond **IoT** : Internet des Objets

**API** : Interface de Programmation d’Application

**SDK** : Kit de Développement de Logiciel

**IDE** : Environnement de Développement Intégré

**CNN** : Réseaux Neuronaux Convolutifs (Convolutional Neural Networks)

**CT** : Tomodensitométrie (Computed Tomography)

**IRM** : Imagerie par Résonance Magnétique **DSE** : Dossiers de Santé Électroniques **HDF5** : Hierarchical Data Format version 5

**Keras** : Bibliothèque de Réseau de Neurones de Haut Niveau en Python

**COVID-19** : Coronavirus Disease 2019

**ReLU** : Rectified Linear Unit

**ROC** : Receiver Operating Characteristic

**AUC** : (Area Under Curve) Aire sous la courbe

# Table des matières

[Résumé iv](#_TOC_250005)

[Table des figures vii](#_TOC_250003)

[Liste des tableaux viii](#_TOC_250002)

[Listings ix](#_TOC_250001)

[Liste des abréviations x](#_TOC_250000)

[Introduction Générale 14](#_bookmark0)

[Chapitre 1 : DEFINITION ET CONCEPTS DE BASE 16](#_bookmark1)

* 1. [Introduction 16](#_bookmark2)
  2. [Détection des maladies pulmonaires 16](#_bookmark3)
     1. [Le Poumon 16](#_bookmark4)
     2. [Démonstration de Maladie Pulmonaire 19](#_bookmark5)
     3. [Planification des taches. 20](#_bookmark6)
     4. [Diagramme de cas utilisation 23](#_bookmark7)
  3. [Conclusion 23](#_bookmark8)

[Chapitre 2 : ETAT DE L’ART DES TECHNIQUE DE TRAITEMENT D’IMAGE](#_bookmark9)

[POUR LA DETECTION DES MALADIES PULMONAIRE 24](#_bookmark9)

* 1. [Introduction 24](#_bookmark10)
  2. [L’intelligence artificielle et l’apprentissage automatique 26](#_bookmark11)
  3. [Définition d’apprentissage automatique 27](#_bookmark12)
     1. [Déploiement d'un modèle d'apprentissage automatique 28](#_bookmark13)
     2. [Algorithmes d'apprentissage automatique utilisés dans cette thèse 29](#_bookmark14)
  4. [Définition d’apprentissage profond 31](#_bookmark15)
  5. [Technique actuelle sur le marché (traitement d’image) 33](#_bookmark16)
     1. [Rappel sur la notion image 33](#_bookmark17)
     2. [Définition de l'image 33](#_bookmark18)
     3. [Types d’images 34](#_bookmark19)
     4. [Caractéristiques de l'image 35](#_bookmark20)
     5. [Techniques de Prétraitement des Images : Histogrammes 36](#_bookmark21)
  6. [Réseaux de neurones 38](#_bookmark22)
     1. [Architecture CNN Utilisée Détection des Maladies Pulmonaires 40](#_bookmark23)
  7. [Conclusion 42](#_bookmark24)

[CHAPITRE 3 : PRESENTATION DU PROJET 43](#_bookmark25)

* 1. [Introduction 43](#_bookmark26)
  2. [Présentation du Project 43](#_bookmark27)
     1. [Objectifs d’application Énoncé du problème 44](#_bookmark28)
     2. [Acquisition de données 44](#_bookmark29)
     3. [Préparation des données 45](#_bookmark30)
     4. [Opération d’égalisation d’histogramme 46](#_bookmark31)
     5. [Importance de la détection des maladies pulmonaires 46](#_bookmark32)
  3. [Méthodologie et outil de travail 46](#_bookmark33)
     1. [Environnement du travail 47](#_bookmark34)
  4. [Conclusion 53](#_bookmark35)

[CHAPITRE 4 : CONCEPTION ET MISE EN PLACE D’APPLICATION DE](#_bookmark36) [DETECTION DES MALADIES PULMONAIRES BASÉE SUR LE TRAITEMENT](#_bookmark36) [DES IMAGES 54](#_bookmark36)

* 1. [Introduction 54](#_bookmark37)
  2. [Application du modèle sur la base de données 54](#_bookmark38)
     1. [Préparation des données. 54](#_bookmark39)
     2. [Prétraitement des images 54](#_bookmark40)
     3. [Entraînement et enregistrement d’un modèle de réseau de neurones convolutifs](#_bookmark41) [(CNN) pour la classification d'images. 62](#_bookmark41)
     4. [Prédiction avec le modèle 66](#_bookmark42)
     5. [Développement d’application 72](#_bookmark43)
  3. [Conclusion 78](#_bookmark44)

[Conclusion Générale du Projet 79](#_bookmark45)

[Bibliographie 80](#_bookmark46)

[Webographie 82](#_bookmark47)

# Introduction Générale

La santé joue un rôle fondamental dans le bien-être individuel et collectif, englobant le diagnostic, le traitement, la prévention des maladies et la promotion de modes de vie sains. Le système de santé, constitué d'organisations, de professionnels et d'établissements, fournit des services médicaux à la population. Les établissements de santé publique sont essentiels pour promouvoir et protéger la santé, prévenir la propagation des maladies et encourager des comportements sains.

Les technologies médicales, notamment la radiologie, permettent des diagnostics précis en visualisant l'intérieur du corps. Depuis deux ans, la pandémie de COVID-19 a profondément bouleversé la vie quotidienne. La propagation rapide du virus et ses conséquences dévastatrices, en particulier pour les personnes âgées ou souffrant de maladies pulmonaires et chroniques, ont exercé une pression considérable sur les établissements de santé et le personnel médical. Face à la demande croissante et à la pénurie de tests de dépistage, tels que les tests PCR et sérologiques, la radiographie pulmonaire est devenue une technique essentielle pour réduire l'incertitude.

Diverses technologies sont couramment utilisées pour diagnostiquer les maladies. Parmi celles-ci, on retrouve :

Les tests de laboratoire : Analysent des échantillons de sang, d'urine ou d'autres fluides corporels pour détecter la présence de maladies.

La radiologie : Utilise des outils d'imagerie tels que les rayons X, les scanners CT et les IRM pour visualiser l'intérieur du corps.

L'endoscopie : Insère une petite caméra dans le corps pour visualiser l'intérieur et diagnostiquer des conditions.

Les biopsies : Prélèvent un échantillon de tissu du corps pour l'examiner au microscope.

Les tests génétiques : Analyser l'ADN pour identifier des mutations ou variations génétiques associées à une condition particulière.

Les dossiers de santé électroniques (DSE) : Enregistrements numériques regroupant les antécédents médicaux d'une personne pour faciliter le diagnostic et le traitement.

L'intelligence artificielle (IA) et l'apprentissage automatique : Analyser les données médicales

pour assister le diagnostic et la planification des traitements.

Le traitement d'imagerie médicale a révolutionné la médecine grâce à des techniques de traitement d'image adaptées. L'apprentissage profond, utilisant notamment des réseaux de neurones convolutionnels (CNN), est largement sollicité pour détecter diverses pathologies telles que les tumeurs, les cancers de la prostate, les nodules pulmonaires et les cancers du sein.

De nombreux travaux de recherche ont été réalisés pour entraîner des modèles CNN afin de détecter le COVID-19 à partir d'images radiographiques pulmonaires. Cette étude se concentre sur l'utilisation de l'analyse d'images radiographiques pour détecter les maladies pulmonaires, y compris le COVID-19. Nous utilisons des réseaux neuronaux convolutifs et des histogrammes pour analyser les images et identifier les anomalies pulmonaires. Cette méthode combine intelligence artificielle et apprentissage automatique pour améliorer les diagnostics médicaux, avec pour objectif de fournir des outils plus efficaces et précis pour la détection et le traitement des maladies pulmonaires.

Dans ce projet, nous avons développé un modèle CNN léger à partir d'un apprentissage profond en utilisant la base de données COVID-QU de radiographies pulmonaires. Nous avons prétraité les images en égalisant leur histogramme pour équilibrer l'intensité, puis divisé l'ensemble de données en deux sous-ensembles, Train et Test. Enfin, nous avons implémenté notre architecture CNN pour détecter efficacement la présence du COVID-19 ou d'autres maladies pulmonaires à partir d'images radiographiques, afin d'aider le personnel médical à évaluer rapidement l'état des patients et à minimiser les dommages.

Tous ces éléments sont abordés dans ce projet à travers quatre chapitres principaux. Ce mémoire comprend les chapitres suivants :

Définition et concepts de base : Explique les notions fondamentales liées au traitement d'image médicale.

État de l’art des techniques de traitement d’image pour la détection des maladies pulmonaires : Analyse des méthodes actuelles utilisées pour identifier les maladies pulmonaires via l'imagerie.

Présentation du projet : Introduction des objectifs, du contexte et de la portée du projet.

Mise en place d’une méthode basée sur le Deep Learning pour le traitement des images : Développement d'une approche utilisant l'apprentissage profond pour analyser les images médicales.

## Chapitre 1 : DEFINITION ET CONCEPTS DE BASE

### Introduction

Les maladies pulmonaires représentent la deuxième cause de mortalité liée aux troubles respiratoires. La radiographie pulmonaire est une méthode privilégiée pour le diagnostic de ces maladies. L'imagerie joue un rôle important dans le processus de diagnostic pour les patients. Par conséquent, les techniques d'intelligence artificielle (IA) peuvent assister les radiologues dans l'interprétation de ces images. Dans ce chapitre, nous allons explorer cette maladie et les méthodes de détection, en particulier à travers l'imagerie médicale.

### Détection des maladies pulmonaires

##### Le Poumon

Les poumons sont situés dans la poitrine, de chaque côté du cœur. Ils sont divisés en plusieurs lobes, eux-mêmes subdivisés en segments. Le poumon gauche est composé de deux lobes tandis que le poumon droit en compte trois.

Figure 1.1: Les poumons dans le système respiratoire

Les bronches prolongent la trachée, permettant le passage de l'air inspiré et expiré par le nez et la bouche. Elles pénètrent dans les poumons et se ramifient progressivement pour former les bronchioles. Les bronchioles se terminent par les alvéoles pulmonaires, petites cavités où se produisent les échanges gazeux entre l'air respiré et le sang. Lors de l'inspiration, l'air passe par la trachée et se

répartit dans les bronches, les bronchioles, puis les alvéoles. L'oxygène contenu dans l'air traverse la paroi des alvéoles pour entrer dans le sang, qui le distribue ensuite à toutes les cellules du corps. Simultanément, le dioxyde de carbone produit par les cellules est transporté par le sang jusqu'aux poumons, traverse la paroi des alvéoles, passe dans les bronches, et est finalement expulsé par la trachée, le nez et la bouche lors de l'expiration.

###### Maladies pulmonaires

Les maladies pulmonaires regroupent diverses affections qui touchent les poumons et les voies respiratoires, entravant la capacité respiratoire et l'échange efficace des gaz. Parmi les maladies pulmonaires les plus courantes figurent l'asthme, la bronchite chronique, l'emphysème, la pneumonie, COVID 19 et les cancers pulmonaires. Ces maladies peuvent être causées par des infections, des inflammations, des allergies, des substances toxiques, ou des prédispositions génétiques. Elles se manifestent souvent par des symptômes tels que la toux, la dyspnée (difficulté à respirer), la douleur thoracique, et la fatigue, nécessitant des diagnostics précis et des traitements adaptés pour améliorer la qualité de vie des patients et prévenir les complications graves.

**Les causes du cancer du poumon**

Les maladies pulmonaires peuvent avoir diverses causes, allant des infections aux facteurs environnementaux et génétiques. Voici quelques-unes des principales causes :

* + Infections :

Bactériennes : Les infections bactériennes comme la pneumonie causée par Streptococcus pneumoniae.

Virales : Les infections virales comme la grippe et d'autres virus respiratoires. Fongiques : Les infections fongiques comme l'aspergillose.

* + Facteurs environnementaux :

Pollution de l'air : L'exposition prolongée à des niveaux élevés de pollution peut entraîner des maladies respiratoires chroniques.

Fumée de tabac : Le tabagisme actif et passif est une cause majeure de maladies pulmonaires telles que la bronchite chronique, l'emphysème et le cancer du poumon.

Exposition professionnelle : L'inhalation de poussières, de produits chimiques et d'amiante dans certains environnements de travail peut causer des maladies pulmonaires.

* + Allergies :

Pollens, acariens, moisissures : Les allergies respiratoires peuvent déclencher ou aggraver des conditions comme l'asthme.

* + Prédispositions génétiques :

Certaines maladies pulmonaires, comme la fibrose kystique, ont une composante génétique.

* + Maladies auto-immunes :

Des conditions comme la sarcoïdose et certaines formes de pneumopathies interstitielles peuvent être causées par des réponses immunitaires anormales.

* + Mode de vie et autres facteurs :

Sédentarité : Un manque d'activité physique peut contribuer à une mauvaise santé respiratoire. Obésité : Peut affecter la fonction pulmonaire et aggraver des maladies existantes comme l'apnée du sommeil.

* + Complications d'autres maladies :

Maladies cardiovasculaires : Les problèmes cardiaques peuvent avoir des répercussions sur la fonction pulmonaire.

* + Substances irritantes :

Produits chimiques : L'exposition à des produits chimiques irritants dans le ménage ou sur le lieu de travail peut endommager les poumons.

Comprendre ces causes est très importants pour la prévention, le diagnostic précoce, et la gestion efficace des maladies pulmonaires.

**Symptôme du cancer du poumon**

Les maladies pulmonaires peuvent présenter une variété de symptômes, souvent liés à des difficultés respiratoires et à des problèmes de circulation de l'air dans les poumons. Voici quelques symptômes courants associés aux maladies pulmonaires :

* + Toux persistante :

Une toux qui dure plusieurs semaines, souvent accompagnée de mucosités.

* + Dyspnée (difficulté à respirer) :

Essoufflement, même lors d'activités légères ou au repos.

* + Douleur thoracique :

Sensation de pression ou de douleur dans la poitrine, surtout lors de la respiration profonde ou de la toux.

* + Sifflements (wheezing) :

Bruits respiratoires sifflants, souvent audibles lors de l'expiration.

* + Fatigue :

Sensation de fatigue ou d'épuisement chronique, souvent due à une réduction de l'apport en

oxygène.

* + Expectorations :

Production de mucus épais, parfois teinté de sang.

* + Cyanose :

Coloration bleuâtre des lèvres, du bout des doigts et des orteils, indiquant un manque d'oxygène dans le sang.

* + Perte de poids :

Perte de poids involontaire et perte d'appétit.

* + Infections respiratoires fréquentes : Répétition de bronchites ou de pneumonies.
  + Difficulté à dormir :

Troubles du sommeil, souvent causés par une respiration difficile ou une toux nocturne.

Ces symptômes peuvent varier en intensité et en durée selon la maladie pulmonaire sous-jacente. Il est essentiel de consulter un professionnel de la santé pour un diagnostic précis et un traitement approprié si l'un de ces symptômes persiste.

**Les traitements des maladies des poumons**

Le traitement des maladies pulmonaires nécessite souvent une approche multidisciplinaire incluant des médicaments, des thérapies par inhalation, des interventions chirurgicales et des soins de soutien. Les progrès de la recherche et des technologies médicales continuent d'améliorer les options thérapeutiques disponibles, offrant ainsi de meilleures perspectives pour les patients souffrant de maladies pulmonaires.

##### Démonstration de Maladie Pulmonaire

Dans cette section, nous démontrons la détection des maladies pulmonaires à travers un ensemble de méthodes appliquées dans le domaine de la santé par exemple le diagnostic par l'Imagerie Médicale ou le diagnostic précoce des maladies pulmonaires qui est essentiel pour le traitement efficace et le contrôle de leur progression. Les principales techniques d'imagerie médicale utilisées comprennent la radiographie thoracique, la tomodensitométrie (TDM), l'IRM et l'échographie des poumons. On trouve aussi la radiographie thoracique qui utilise des rayons X pour évaluer les poumons et est l'outil de diagnostic le plus courant et le plus accessible pour surveiller la santé pulmonaire. Elle est largement utilisée pour diagnostiquer des conditions telles que la pneumonie, la tuberculose et le cancer du poumon. On distingue aussi des Vues de la Radiographie Thoracique :

Vue Frontale : Permet une évaluation globale de la cavité thoracique.

Vue Latérale : Fournit des informations complémentaires pour une analyse plus détaillée.

Ces deux vues sont souvent prises pour offrir une vision complète des poumons et aider à un diagnostic précis. Les radiographies thoraciques jouent un rôle clé dans la détection précoce et le suivi des maladies pulmonaires, facilitant ainsi une intervention rapide et appropriée.

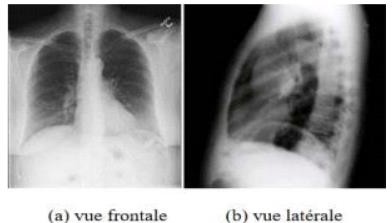


Figure 1.2: La radiographie thoracique

##### Planification des taches.

Pour créer un diagramme de Gantt pour ce projet, nous allons identifier les tâches principales et leur durée. Voici une liste des tâches et leur séquence logique, suivies d'une représentation textuelle du diagramme de Gantt.

**Maroua El Hamidi** : A contribué à l'élaboration du rapport et au prétraitement des images. Elle s'est occupée des recherches sur la littérature et les algorithmes possibles, de la recherche du dataset, ainsi que de son nettoyage et de sa préparation.

**Meryem Akarkab** : A participé à la conception et à la planification du projet, ainsi qu'au prétraitement des images.

**Hakima Ighidi :** a mené une recherche sur diverses techniques de traitement d'images, notamment l'égalisation d'histogramme, les filtres gaussiens et bilatéraux, ainsi que la convolution avec des noyaux gaussiens 1D et 2D. Elle a ensuite appliqué ces techniques dans des tâches pratiques pour mettre en œuvre les concepts étudiés.

**Laila Ighidi :** a réalisé des recherches théoriques sur les réseaux neuronaux convolutifs (CNN) et, dans la partie pratique, elle a conçu et implémenté un modèle de CNN pour résoudre des problèmes de reconnaissance et de traitement d'images.

**Bader Eddine Belmouss** : Développer un modèle CNN en TensorFlow pour classifier des images en trois catégories. Il a utilisé des données d'entraînement augmentées et valider le modèle. Créé une application GUI avec CustomTkinter pour prédire la présence de Covid-19. Afficher les résultats avec une barre de progression et gérer les erreurs. Ainsi que l’Optimisation de l'application pour améliorer les performances et l'interface.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Tableau 1.2 : Taches Diagramme de Gantt

###### Explication :

Le projet de fin d'études s'étend du 22 juillet au 31 août 2024, avec 16 tâches distinctes. Il commence par la définition du sujet et des objectifs, suivie par la recherche et la synthèse de la littérature. Ensuite, les sources de données sont identifiées et filtrées. Les données sont ensuite préparées et annotées, avant que les outils et technologies ne soient sélectionnés. Le prétraitement des données est effectué avant la sélection, l'entraînement et l'évaluation des modèles. Enfin, les performances sont analysées, les résultats visualisés, et le projet se conclut par la rédaction du rapport et la soutenance.

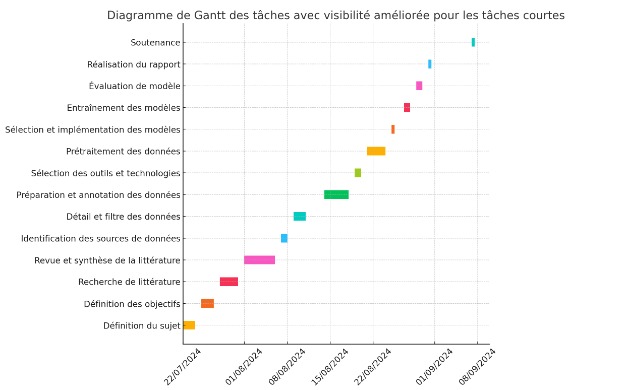


Tableau 1 .3 : Diagramme de Gantt

##### Diagramme de cas utilisation

Voici le diagramme de cas d'utilisation de notre application, décrivant les interactions principales entre l'utilisateur et le système de prédiction

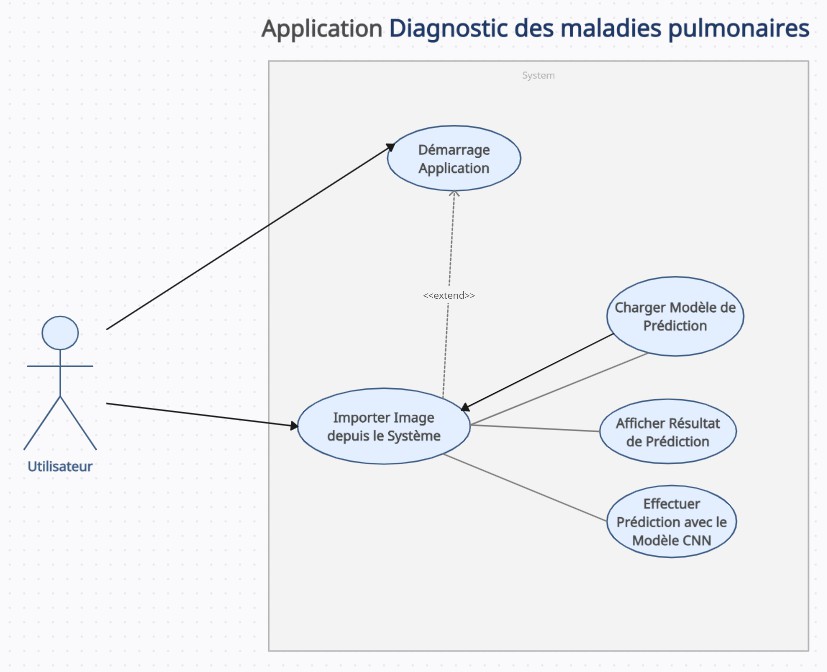


Figure 1.5 : Diagramme de cas utilisation.

Ces cas d'utilisation couvrent les principales fonctionnalités de l'application et les interactions attendues entre l'utilisateur et le système.

### Conclusion

Le diagnostic des maladies pulmonaires par imagerie médicale est essentiel pour une détection rapide et une gestion efficace. Les techniques telles que la radiographie thoracique, la TDM, l'IRM et l'échographie permettent de visualiser les structures pulmonaires et de détecter les anomalies. La radiographie thoracique, la plus utilisée, facilite la détection de maladies comme la pneumonie, la tuberculose et le cancer, améliorant les résultats pour les patients.

## Chapitre 2 : ETAT DE L’ART DES TECHNIQUE DE TRAITEMENT D’IMAGE POUR LA DETECTION DES MALADIES PULMONAIRE

### Introduction

Le traitement d'imagerie médicale, une branche de la vision par ordinateur, a révolutionné la médecine en intégrant des techniques d'imagerie adaptées, notamment l'apprentissage profond via les réseaux de neurones convolutifs (CNN). Ces derniers sont reconnus pour leur capacité à extraire des caractéristiques et à classifier les données, utilisés dans divers domaines médicaux tels que la détection de tumeurs cérébrales, de cancer de la prostate et du sein, de nodules pulmonaires.

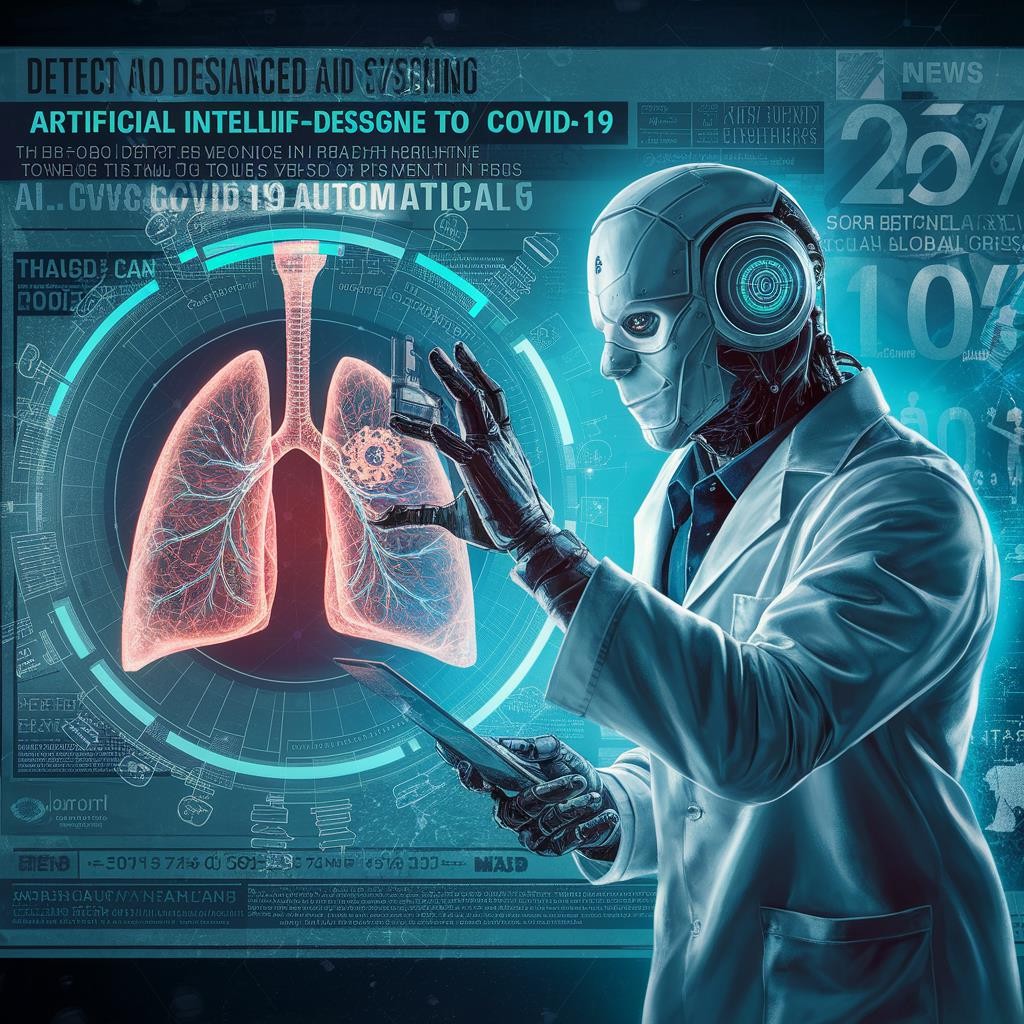


Figure 2.1 - Nuage de mots des concepts et technologies utilisés dans ce projet.

De nombreuses recherches ont été menées pour entraîner des modèles CNN à détecter les maladies pulmonaires ainsi que le COVID-19 à partir de radiographies pulmonaires. Notre approche repose sur une architecture CNN légère, utilisant une base de données COVID-QU. Après un prétraitement par égalisation d'histogramme pour améliorer la qualité des images, nous

avons divisé les données en ensembles d'entraînement et de test. Notre modèle vise à détecter efficacement la présence d'un virus pulmonaire ou COVID-19 sur des radiographies pulmonaires, facilitant ainsi l'évaluation médicale et la prise de décision rapide. L’expérience des machines se présente sous la forme de données. Les données utilisées pour enseigner aux machines sont appelées données de formation. L'apprentissage automatique bouleverse le modèle de programmation traditionnel (Figure 2.3). Un logiciel, constitué d'un ensemble d'instructions destinées à un ordinateur, convertit les signaux d'entrée en signaux de sortie en suivant des règles et des relations établies à l'avance. Les algorithmes d'apprentissage automatique, également appelés "apprenants", prennent à la fois l'entrée et la sortie connues (données d'apprentissage) pour déterminer un modèle pour le programme qui convertit l'entrée en sortie. Par exemple, de nombreuses organisations telles que les plateformes de médias sociaux, les sites d'évaluation ou les forums sont tenues de modérer les publications et de supprimer le contenu abusif. Comment enseigner aux machines à automatiser le filtrage des contenus abusifs ? Il est nécessaire de leur fournir des exemples de messages, en précisant clairement ceux qui sont considérés comme abusifs et ceux qui ne le sont pas. Les apprenants généraliseront un modèle utilise des mots spécifiques ou des séquences de mots pour déterminer si le message dans son ensemble est abusif ou non. Le modèle peut prendre être sous la forme d'un ensemble de règles. Une fois que les règles ou le modèle de science des données sont développés, les machines peuvent commencer à catégoriser la disposition de tout nouveau message.

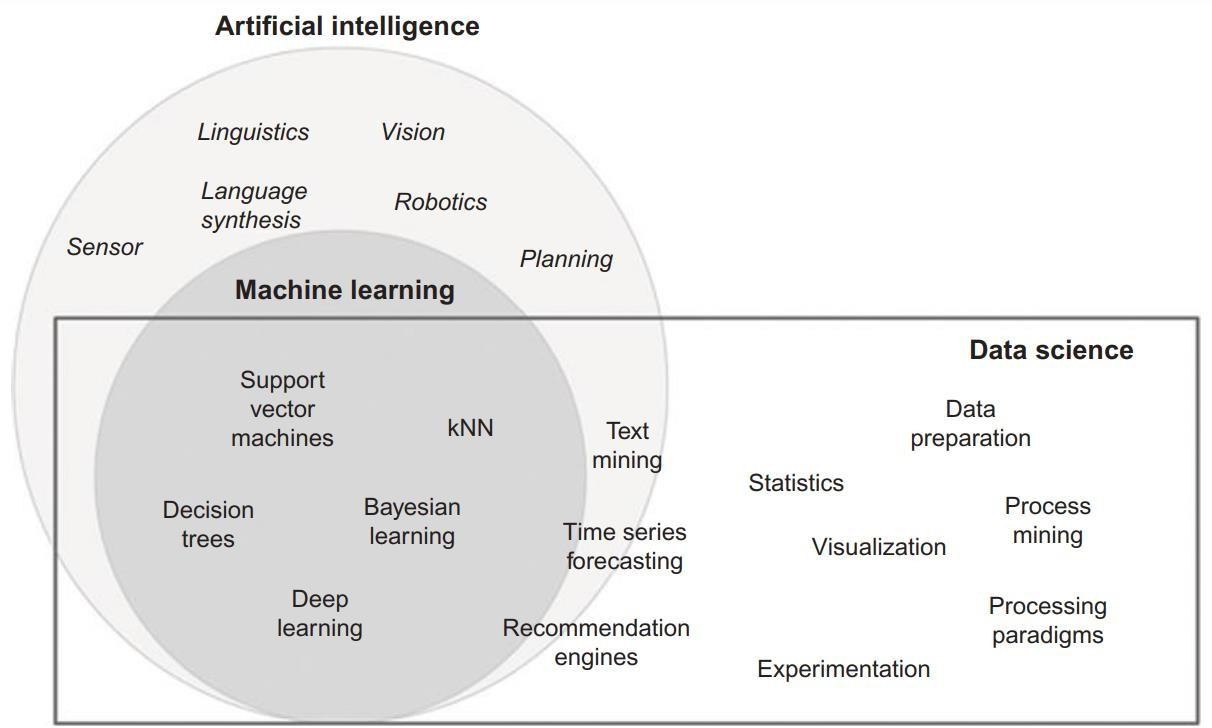
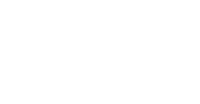
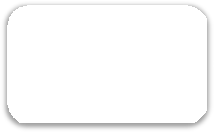


Figure 2.2 - Intelligence artificielle, apprentissage automatique et science des données.

La science des données intègre l'apprentissage automatique, l'intelligence artificielle et d'autres disciplines quantitatives pour extraire de la valeur des données. Elle trouve des applications dans divers domaines, y compris les recommandations personnalisées, la détection de la fraude et les prévisions de revenus, facilitant ainsi la prise de décisions fondées sur les données.

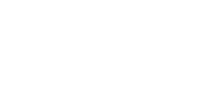
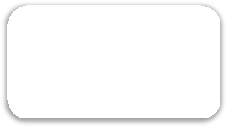


**Program**

**Output Y**

**Input X**

**Input X**



**Machine Learning**

**Représentative Model of The program**

Figure 2.3 - Programme traditionnel et apprentissage automatique.

Approches d'apprentissage en profondeur Il existe plusieurs approches différentes de l'apprentissage en profondeur, notamment :

Réseaux de neurones convolutifs (CNN) : ils sont utilisés pour les tâches de reconnaissance d'images et de vidéos, ainsi que pour le traitement du langage naturel. Les CNN sont conçus pour traiter des données avec une topologie en forme de grille, telle qu'une image.

##### L’intelligence artificielle et l’apprentissage automatique

L'intelligence artificielle (IA) consiste à créer des machines qui imitent le fonctionnement de l'intelligence humaine. Ces machines sont capables d'apprendre, de résoudre des problèmes et de prendre des décisions en fonction des données fournies. L'IA offre un large éventail d'applications, notamment dans les domaines des véhicules autonomes, du diagnostic médical, de la reconnaissance d'images et du traitement du langage naturel. Cependant, son développement soulève des questions éthiques, notamment sur le plan de l'emploi et des préoccupations concernant son utilisation potentielle à des fins malveillantes. Les systèmes d'IA peuvent intégrer divers composants, comme illustré dans la Figure 2.4.

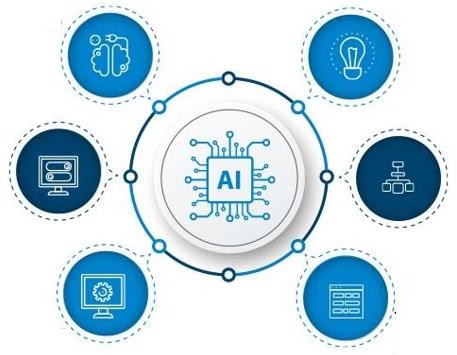
Certains des composants clés incluent :

* Algorithmes d'apprentissage automatique : ce sont des algorithmes qui permettent à un système d'apprendre à partir de données et d'améliorer ses performances au fil du temps.
* Représentation des connaissances et raisonnement : il s'agit d'un domaine de l'IA qui se concentre sur la manière de représenter les connaissances et de les utiliser pour prendre des décisions et résoudre des problèmes.
* Prise de décision : il s'agit d'un domaine de l'IA qui se concentre sur la conception d'algorithmes et de systèmes capables de prendre des décisions en fonction des informations disponibles.

En général, les différents composants de l’intelligence artificielle fonctionnent ensemble pour permettre à un système d'effectuer des tâches intelligentes, telles que la reconnaissance de visages sur des photos, la traduction de langues, etc.

Prise de Décision

Robotique



Apprentissage Automatique

Représentation Connaissances Raisonnement

Vision par Ordinateur

Traitement Langage Naturel (NLP)

Figure 2.4 - Différents composants de l’intelligence artificielle.

### Définition d’apprentissage automatique

L'apprentissage automatique, une branche de l'intelligence artificielle, consiste à permettre aux ordinateurs d'apprendre à partir de données sans programmation explicite. Son objectif est de créer des algorithmes capables de recevoir des données d'entrée et de les utiliser pour prédire ou

prendre des mesures spécifiques. Les algorithmes peuvent être supervisés, utilisant des données étiquetées, ou non supervisés, avec des données non structurées. L'apprentissage automatique utilise généralement des algorithmes et des modèles statistiques pour améliorer les performances au fil du temps, impliquant divers composants tels que le prétraitement des données, l'ingénierie des fonctionnalités, l'entraînement du modèle et l'évaluation.

L'apprentissage automatique permet aux ordinateurs d'apprendre à partir de données sans programmation explicite. Il repose sur l'utilisation d'algorithmes et de modèles statistiques pour améliorer les performances du système au fil du temps. Les composants essentiels de l'apprentissage automatique comprennent :

* Données : Les algorithmes d'apprentissage automatique requièrent une vaste quantité de données pour s'entraîner. Ces données servent à former le modèle et à faire des prédictions sur de nouvelles données.
* Modèle : Le modèle constitue le cœur d'un système d'apprentissage automatique. Il représente mathématiquement les relations et les schémas présents dans les données. Créé par l'algorithme d'apprentissage, le modèle est utilisé pour effectuer des prédictions sur de nouvelles données.
* Algorithme : Un algorithme d'apprentissage est un ensemble d'instructions guidant le modèle dans la découverte de schémas dans les données et leur utilisation pour effectuer des prédictions. Il existe divers types d'algorithmes, chacun ayant ses propres avantages et inconvénients.
* Évaluation : Une fois que le modèle a été formé et fait des prédictions, il est important d'évaluer ses performances pour s'assurer qu'il est précis et fiable. Cela se fait généralement à l'aide d'un ensemble de données de test que le modèle n'a jamais vu auparavant.

##### Déploiement d'un modèle d'apprentissage automatique

La création d'un modèle d'apprentissage automatique est un processus itératif qui implique la répétition de plusieurs étapes pour obtenir des résultats optimaux avant le déploiement. Une fois déployé, le modèle doit être régulièrement entretenu et ajusté pour s'adapter aux changements environnementaux. La Figure 2.5 présente les différentes étapes du cycle de vie d'un modèle d'apprentissage automatique.

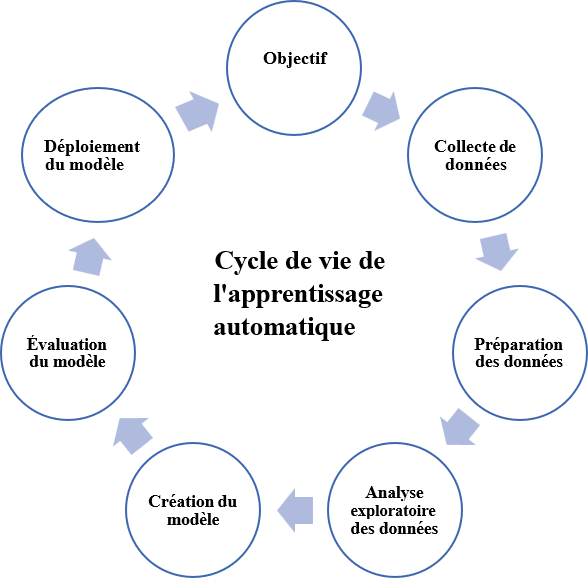


Figure 2.5 - Cycle de vie de l'apprentissage automatique.

La figure ci-dessus représente toutes les étapes impliquées dans le cycle de vie de l'apprentissage automatique requis pour le développement et la mise en œuvre des modèles d'apprentissage automatique. C'est un processus cyclique parce que le processus est continu et répétitif, suivez séquentiellement une étape après une autre étape.

##### Algorithmes d'apprentissage automatique utilisés dans cette thèse

Les algorithmes d'apprentissage automatique sont des algorithmes conçus pour apprendre à partir de données. Ils sont couramment utilisés pour construire des modèles prédictifs capables de faire des prédictions sur des événements futurs sur la base de données précédemment observées.

* Algorithmes de Convolution :

Utilisés pour extraire les caractéristiques des images en appliquant des filtres sur les images en entrée. Ces filtres sont spécifiés dans les couches Conv2D du modèle.

* Algorithme de Max Pooling :

Utilisé pour réduire la dimensionnalité des images en sélectionnant la valeur maximale dans chaque fenêtre de la grille. Cela aide à réduire le surajustement et à accélérer l'entraînement du modèle.

* Algorithme de Prétraitement des Images :

Utilisé avec l'objet Image Data Generator pour effectuer des opérations de prétraitement telles que la mise à l'échelle, le recadrage, la rotation, le zoom, etc., sur les images pendant l'entraînement du modèle.

* Algorithme gryscrole :

L'algorithme Gryscrole est une technique avancée utilisée dans le domaine du traitement d'image. Cet algorithme se distingue par sa capacité d'améliorer la clarté des images ou détecter les contours avec une précision élevée. Son application est particulièrement efficace pour l'analyse des radiographies médicales ou la restauration d'images endommagées démontrant ainsi son potentiel pour améliorer la qualité des diagnostics ou fournir des images plus nettes pour une meilleure analyse visuelle.

* Algorithme SOBEL :

L'algorithme Sobel est une technique couramment utilisée en traitement d'image pour détecter les contours. Il accentue les variations d'intensité, facilitant l'identification des bords et des lignes, notamment dans les images médicales et la vision par ordinateur.

* Algorithme de Sauvegarde et de Chargement de Modèle :

Utilisé pour sauvegarder le modèle entraîné au format HDF5 (.h5) et au format Keras (.keras), puis pour charger le modèle à partir de ces fichiers pour une utilisation ultérieure.

* Algorithme de Matrice de Confusion :

Utilisé pour évaluer les performances du modèle en comparant les prédictions du modèle avec les valeurs réelles et en les affichant sous forme de matrice.

* Algorithme Canny:

L'algorithme Canny est une méthode efficace pour la détection des contours dans les images. Il se distingue par sa capacité à identifier des bords avec précision tout en réduisant le bruit. Utilisé dans diverses applications, notamment l'imagerie médicale et la vision par ordinateur, l'algorithme Canny aide à produire des images claires et détaillées en détectant des contours nets et continus, améliorant ainsi l'analyse et l'interprétation des données visuelles.

* Autres algorithmes de visualisation :

Utilisés pour visualiser les performances du modèle à l'aide de graphiques, tels que les courbes d'apprentissage pour l'exactitude et la perte au fil des époques.

#### Définition d’apprentissage profond

L'apprentissage en profondeur est une branche de l'apprentissage automatique où des réseaux de neurones artificiels sont formés sur de grandes quantités de données. Appelé "profond" en raison du nombre de couches de neurones, il s'inspire de la structure cérébrale. Plutôt que d'être programmé avec des règles spécifiques, le réseau apprend en analysant des exemples, permettant ainsi l'apprentissage de modèles complexes pour la prise de décision.

Ce type d'apprentissage a été efficace dans diverses applications, telles que la reconnaissance d'images et de la parole, le traitement du langage naturel, et même des jeux comme les échecs. Il excelle dans la manipulation de données non structurées comme les images et le texte, le rendant précieux pour de nombreuses tâches nécessitant la compréhension de grandes quantités de données.

Le modèle est formé en ajustant les poids des neurones pour minimiser l'erreur entre les prédictions et les résultats corrects, à l'aide d'algorithmes comme la rétropropagation et la descente de gradient stochastique. Les CNN (réseaux de neurones convolutifs) sont particulièrement adaptés à la reconnaissance d'images, tandis que les RNN (réseaux de neurones récurrents) sont utilisés pour des données séquentielles telles que la traduction de langues et la reconnaissance vocale.

La popularité croissante de l'apprentissage en profondeur est due à la disponibilité de grandes quantités de données et à des avancées matérielles, facilitée par des bibliothèques open source comme TensorFlow et numpy pandas.

Il existe un ensemble de méthodes d'apprentissage profond, qui sont des techniques de machine learning basées sur des modèles de réseaux de neurones. Ces réseaux comportent des dizaines, voire des centaines de couches de neurones empilées, permettant une plus grande complexité dans l'établissement des règles et améliorant ainsi la précision et l'efficacité des modèles. Donc l'apprentissage en profondeur comprend plusieurs approches distinctes par exemples :

* Réseaux de neurones convolutifs (CNN) : Utilisés pour la reconnaissance d'images et de vidéos, ainsi que pour le traitement du langage naturel. Les CNN sont adaptés aux données en forme de grille, comme les images.
* Réseaux de neurones récurrents (RNN) : Appropriés pour les données séquentielles, comme la traduction de langues et la reconnaissance vocale. Les RNN traitent les données temporelles telles que les séries chronologiques.
* Réseaux antagonistes génératifs (GAN) : Utilisés pour créer de nouvelles données similaires à celles existantes. Les GAN combinent un réseau générateur qui crée des données et un réseau discriminateur qui distingue les données générées des données réelles.
* Auto-encodeurs : Employés pour la réduction de dimensionnalité et la détection d'anomalies. Ils apprennent à reconstruire les données d'entrée en créant une représentation compacte dans une couche intermédiaire.

Ces approches ont obtenu des résultats impressionnants dans des domaines variés comme la reconnaissance d'images, la parole, le traitement du langage naturel et les jeux.

Il existe un vaste ensemble d'applications de l'apprentissage en profondeur, couvrant de nombreux domaines par exemples :

* Reconnaissance d'images et de vidéos : Les modèles peuvent classifier des images et vidéos, détecter des objets et générer des descriptions. Utilisés pour la reconnaissance faciale, les voitures autonomes et la vidéosurveillance.
* Traitement du langage naturel : Permet la compréhension et la génération du langage humain pour des tâches comme la traduction automatique et la génération de texte.
* Reconnaissance vocale : Transforme les mots parlés en texte écrit, utilisé dans la dictée et les assistants virtuels.
* Imagerie médicale : Analyse des images médicales pour détecter des anomalies, aidant au diagnostic.
* Systèmes de recommandation : Personnalise les recommandations basées sur les préférences utilisateur, utilisé pour la musique, les films et le commerce en ligne.
* Modélisation financière : Prédit les cours des actions et analyse les données financières pour la gestion des risques et la détection des fraudes.

Ces exemples illustrent l'adaptabilité des modèles d'apprentissage en profondeur à diverses tâches nécessitant une certaine intelligence.

Il existe un ensemble varié de types d'apprentissage profond, chacun adapté à des besoins spécifiques.

* Apprentissage supervisé : L'apprentissage supervisé est une technique d'apprentissage automatique dans laquelle un modèle est formé à partir de données d'entrée et de sortie étiquetées. Le modèle utilise ces données pour apprendre à faire des prédictions précises sur de nouvelles données. Par exemple, dans la classification d'images, des images étiquetées avec des étiquettes telles que "chien" ou "chat" sont utilisées pour entraîner un modèle à identifier les

animaux dans de nouvelles images.

* Apprentissage non supervisé : L'apprentissage non supervisé est une méthode d'apprentissage automatique où un algorithme est formé pour trouver des motifs ou des structures dans un ensemble de données sans l'aide d'étiquettes ou de réponses prédéfinies. Les algorithmes d'apprentissage non supervisé sont souvent utilisés pour explorer et analyser des données, détecter des anomalies, regrouper des données similaires et réduire la dimensionnalité des données.
* Apprentissage semi-supervisé (hybride) : L'apprentissage semi-supervisé hybride est une technique d'apprentissage automatique qui combine des données étiquetées et non étiquetées pour améliorer les performances de la classification ou de la prédiction. Cette approche permet de tirer parti des données non étiquetées pour renforcer la qualité de la modélisation et de l'apprentissage, tout en réduisant la nécessité d'étiqueter manuellement toutes les données d'entrée. Les modèles hybrides peuvent inclure des techniques telles que la classification par ensembles, la régularisation, la propagation de labels, etc.

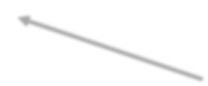
##### Technique actuelle sur le marché (traitement d’image)

##### Rappel sur la notion image

Une image réelle est produite à partir d'un signal continu bidimensionnel, tel que celui capté par une caméra, un scanner ou des capteurs d'imagerie. Ce signal continu doit être échantillonné et quantifié pour devenir une image numérique discrète, qui peut ensuite être stockée et traitée par des ordinateurs.

##### Définition de l'image

L'image peut être représentée par une fonction I(i, j), où i et j sont les coordonnées spatiales d'un point de l'image, et I est une fonction indiquant l'intensité ou le niveau de gris. Sous cette forme, l'image ne peut pas être directement utilisée par une machine, ce qui rend nécessaire sa numérisation. L'image numérique, également appelée scène, a un système de coordonnées différent de celui d'une fonction mathématique traditionnelle, avec une hauteur (H) et une largeur (W).



Ligne i

Figure 2.6 : Image couleur, son repère et un extrait de pixels.

* + 1. **Types d’images**

On distingue cinq types d’images :

Binaire : deux couleurs (arrière-plan et avant-plan) Niveaux de gris : variations d’une même teinte Couleur : couleurs "vraies"

Indexée : utilisation d'une palette de couleurs

À valeurs réelles : chaque pixel représente une valeur réelle

* + - * Image binaire :Une image binaire est une image MxN où chaque pixel peut uniquement être 0 (noir) ou 1 (blanc). Le niveau de gris est codé sur un bit. Avec Ng = 2, la valeur du pixel devient p(i, j) = 0 ou p(i, j) = 1.
      * Image en niveaux de gris (Monochrome) : Les images en niveaux de gris apportent des nuances dans le rendu. Cette nuance provient de la valeur des pixels, qui ne sera plus binaire, mais variera dans une plage (par exemple de 0 à 255). Plus la valeur d'un pixel est proche de zéro, plus le pixel sera sombre ; inversement, plus cette valeur est proche de 255, plus le pixel sera clair. La matrice représentant ce type d'image reste une matrice à deux dimensions, mais avec des valeurs variant dans une plage définie (par exemple entre 0 et 255).

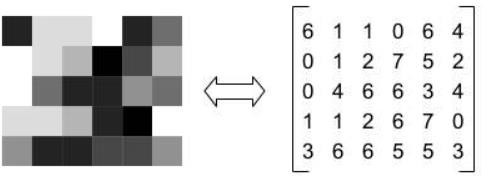


Figure 2.7 : Image en niveaux de matrice à deux dimensions

###### Image couleur (RGB)

Chaque pixel est défini par une combinaison de trois couleurs de base : rouge (R), vert (G) et bleu (B). Chacune de ces composantes est codée sur l'intervalle [0, 255], ce qui donne 255³ = 16 777 216 couleurs possibles. Pour encoder cette information de couleur pour chaque pixel, il est nécessaire d'utiliser 24 bits.



Figure 2.8 : Image en niveaux RGB combinaison de rouge, vert et bleu.

* Image indexée : La plupart des images n'utilisent pas les 16 millions de couleurs possibles pour des raisons de stockage et de gestion des fichiers. Elles utilisent une palette de couleurs représentant la liste des couleurs utilisées. Chaque pixel est représenté par un index qui renvoie à une couleur spécifique dans cette palette.
* Image à valeurs réelles : Pour certains calculs sur les images, il est souvent nécessaire de considérer les résultats comme des valeurs réelles, car ceux-ci ne peuvent pas toujours être des entiers. Ainsi, il est préférable de traiter à la fois l'image originale et l'image résultante comme des images à valeurs réelles. En général, une image à valeurs réelles est caractérisée par des niveaux de gris compris entre 0.0 et 1.0, exprimés en nombres réels.
  + 1. **Caractéristiques de l'image**

Les images possèdent diverses caractéristiques essentielles, permettant une analyse et une interprétation détaillées.

Pixel : À la base de toute image numérique se trouve le pixel, qui est le plus petit élément constitutif de l'image. La quantité d'information véhiculée par chaque pixel détermine les nuances entre les images monochromes et les images en couleur. Pour les images 3D, le terme utilisé est "voxel," représentant un volume élémentaire. Les images médicales, comme les images tomographiques axiales, utilisent ce type de représentation, étant construites à partir de plusieurs radiographies prises sous différents angles.



Figure 2.9 : Représentation des pixels

##### Techniques de Prétraitement des Images : Histogrammes

Avant d'introduire les données dans un réseau de neurones convolutifs (CNN), il est souvent avantageux de prétraiter les images pour améliorer les performances du modèle. L'une des techniques couramment utilisées à cet effet est l'analyse des histogrammes d'images. Cette méthode fournit une représentation de la distribution des niveaux de gris ou des couleurs dans une image, ce qui peut aider à uniformiser l'éclairage et à améliorer le contraste. En ajustant ces aspects, les caractéristiques importantes deviennent plus visibles et discernables pour le modèle, ce qui facilite l'apprentissage et la précision du CNN. Le prétraitement des images est donc une étape importante pour maximiser l'efficacité et la performance des réseaux de neurones convolutifs dans diverses applications.

En imageries numérique l’histogramme est une représentation graphique, souvent sous forme de diagramme en bâtons, des classes d'une variable statistique. Chaque classe est représentée par un rectangle dont la hauteur est proportionnelle à l'effectif de cette classe. Pour un histogramme d'image numérique, l'axe horizontal (abscisse) représente l'échelle des luminosités (valeurs allant de 0 à 255 en 8 bits), et l'axe vertical (ordonnée) indique le nombre de pixels pour chaque valeur de luminosité.

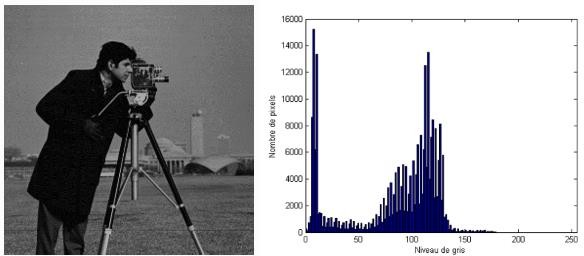


Figure 2.10 histogramme tassé sur la gauche (sombre).

Opération d’égalisation d’histogramme : Pour améliorer la qualité de l'ensemble des images, une opération d'égalisation d'histogramme a été appliquée afin d'obtenir une meilleure répartition du contraste sur toutes les images.

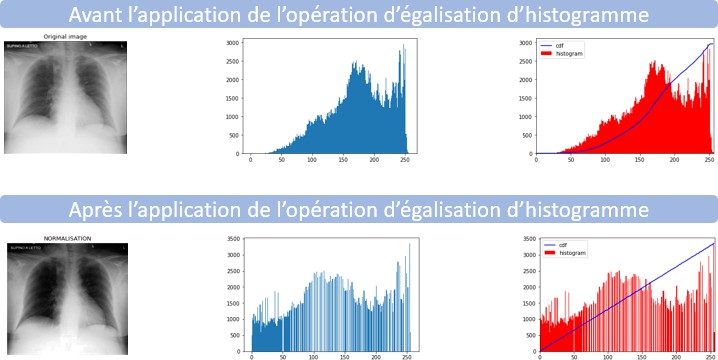


Figure 2.11 Amélioration de la Qualité des Images avec Égalisation d'Histogramme du Contraste

Homogénéité : L'homogénéité est une caractéristique locale qui correspond à l'uniformité d'une région dans une image. Une région est dite homogène si elle contient un ensemble de pixels ayant des caractéristiques similaires ou uniformes, telles que la variance du niveau de gris, la couleur, ou la texture.

Dimension **:** La dimension fait référence à la taille de l'image. Elle est représentée sous forme d'une matrice de valeurs numériques correspondant aux intensités lumineuses de chaque pixel. Le nombre total de pixels dans une image est obtenu en multipliant le nombre de lignes de la matrice par le nombre de colonnes.

Bruit **:** Le bruit désigne tout phénomène imprévisible perturbant le signal. Dans une image, cela se manifeste par une variation brusque de l'intensité d'un pixel par rapport à ses voisins. Le bruit provient souvent de l'éclairage et des dispositifs optiques et électroniques du capteur.

Luminance **:** La luminance indique le degré de luminosité des points de l'image. Elle est définie comme le quotient de l'intensité lumineuse d'une surface par son aire apparente, telle que perçue par un observateur distant. Le terme luminance remplace souvent le terme brillance, qui fait plutôt référence à l'éclat d'un objet.

Une bonne luminance se distingue par :

* Images lumineuses : Les images apparaissent claires et brillantes.
* Reproduction des couleurs : Lorsqu'elle éclaire des objets colorés, la luminance doit être ajustée pour garantir une reproduction fidèle des couleurs.
* Absence de parasites : Il ne doit pas y avoir de bruit ou d'interférences perturbant l'image.

Contraste : Le contraste est une propriété intrinsèque d'une image qui mesure la différence de luminosité entre deux régions claires et sombres. Si L1 et L2 représentent les niveaux de luminosité de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport :

C=(L1+L2)/(L1−L2)

Région : Une région dans une image se réfère à un ensemble de pixels connexes et homogènes. Un pixel appartient à une région spécifique s'il partage les mêmes caractéristiques que cette région (comme l'intensité moyenne ou le centre de gravité). Une région est délimitée par son contour, et un contour fermé définit une région.

Connexité : Lors de l'analyse d'une image, le déplacement à l'intérieur de celle-ci doit respecter des règles de voisinage. En général, on utilise deux types de voisinage : le voisinage à 4-connexité (comprenant 4 pixels voisins) et le voisinage à 8-connexité (comprenant 8 pixels voisins).

### Réseaux de neurones

Les réseaux de neurones convolutifs (CNN) ont trouvé de nombreuses applications dans le domaine médical, particulièrement dans la détection des maladies pulmonaires telles que la pneumonie et le COVID-19. En traitant des radiographies thoraciques, ces réseaux de neurones sont capables d'identifier et de différencier les caractéristiques visuelles associées à ces pathologies.

L'utilisation des CNN pour l'analyse des images médicales repose sur leur capacité à extraire des caractéristiques complexes à partir des données d'image. Ces caractéristiques, souvent indécelables à l'œil humain non expert, permettent de détecter des anomalies spécifiques aux maladies pulmonaires.

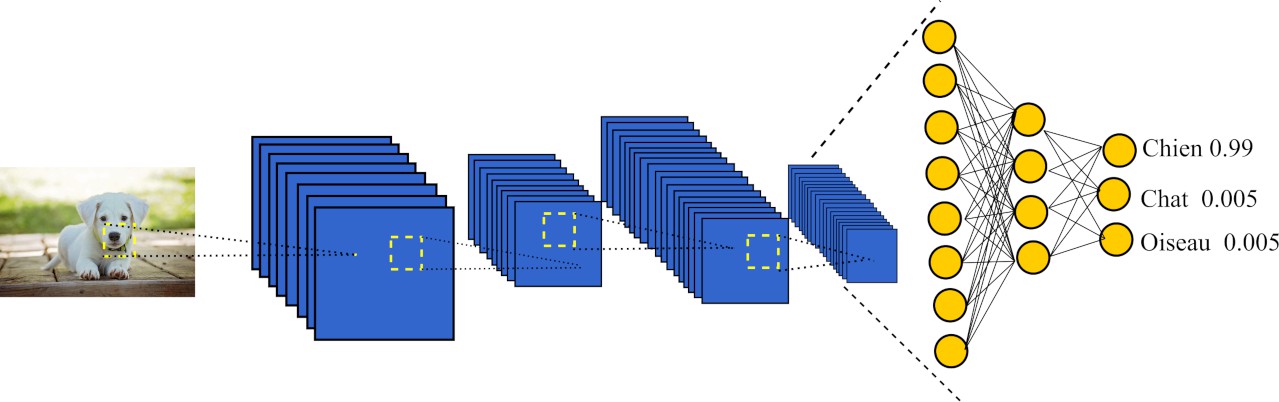


Figure 2-12 Structure générale d’un réseau de neurones convolutifs

Par exemple, dans le cas de la pneumonie, les CNN peuvent identifier des zones d'opacité dans les poumons, alors que pour le COVID-19, ils peuvent repérer des motifs de verre dépoli caractéristiques de la maladie.

L'efficacité des CNN dans le diagnostic médical automatisé repose sur leur capacité à apprendre à partir de grandes quantités de données. En entraînant les modèles sur des milliers, voire des millions, d'images radiographiques, les CNN développent une compréhension approfondie des variations visuelles qui indiquent la présence de maladies. Cette capacité d'apprentissage profond permet aux CNN de fournir des diagnostics précis et rapides, surpassant souvent les méthodes traditionnelles basées sur l'expertise humaine.

Outre leur précision, les CNN offrent également une grande rapidité dans le traitement des données, ce qui est essentiel dans les situations d'urgence médicale. En intégrant ces modèles dans les systèmes de santé, il est possible de réduire le temps nécessaire pour obtenir un diagnostic, permettant ainsi une intervention médicale plus rapide et efficace. De plus, l'utilisation des CNN peut contribuer à pallier le manque de radiologues qualifiés dans certaines régions, en fournissant une assistance diagnostique fiable.

En résumé, les réseaux de neurones convolutifs représentent un outil puissant pour le diagnostic automatisé des maladies pulmonaires. Leur capacité à analyser et à interpréter des radiographies

Thoraciques avec une grande précision et rapidité en fait une technologie précieuse pour améliorer les soins de santé et les résultats des patients.

##### Architecture CNN Utilisée Détection des Maladies Pulmonaires

Pour notre projet de détection des maladies pulmonaires à partir de radiographies, nous avons adopté une architecture spécifique de réseau de neurones convolutifs (CNN). Cette architecture se distingue par sa capacité à analyser et interpréter les images médicales en détectant les caractéristiques visuelles propres aux maladies pulmonaires. Voici une description détaillée de cette architecture :

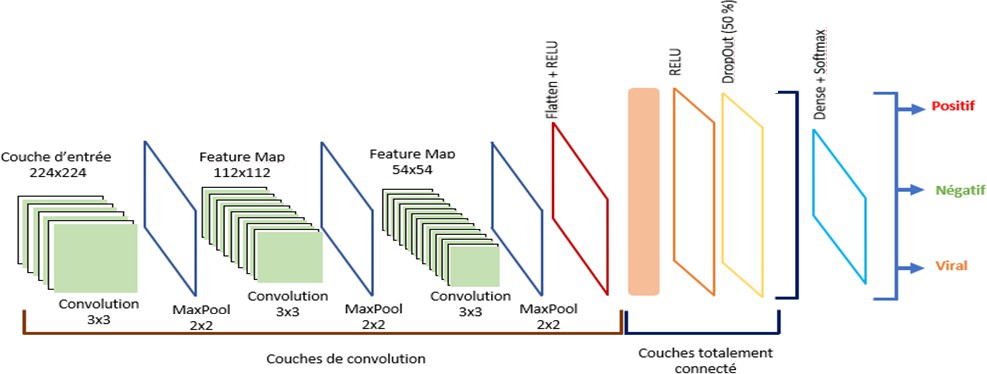
Elle comprend plusieurs couches convolutives pour extraire les caractéristiques, suivies de couches de pooling pour réduire la dimensionnalité. Ensuite, des couches entièrement connectées sont utilisées pour la classification finale. Des techniques de régularisation comme le dropout sont également intégrées pour éviter le surapprentissage. Cette combinaison permet au modèle de traiter efficacement les images radiographiques, améliorant ainsi la précision et la rapidité du diagnostic des maladies pulmonaires.

Figure 2-13 : Architecture CNN

* + - * Trois Couches de Convolution : Les couches de convolution sont utilisées pour extraire les caractéristiques locales des images. Chaque couche applique plusieurs filtres pour identifier différents motifs et structures. Dans notre modèle, nous avons 3 couches de convolution successives, chacune suivie d'une fonction d'activation non linéaire (ReLU) pour ajouter de la non-linéarité au modèle.
      * Trois Couches de MaxPooling : Après chaque couche de convolution, une couche de MaxPooling est appliquée 2x2 pour réduire la dimensionnalité des images et isoler les caractéristiques les plus pertinentes. Le MaxPooling sélectionne la valeur maximale dans une fenêtre de sous-échantillonnage, ce qui aide à diminuer le surapprentissage et à renforcer la robustesse du modèle.
      * Une Couche Flatten : La couche Flatten convertit les cartes de caractéristiques obtenues après les convolutions et les MaxPooling en un vecteur unidimensionnel. Cette transformation prépare les données pour les couches entièrement connectées suivantes.
      * Une Couche Totalement Connectée (Fully Connected Layer) : La couche totalement connectée prend le vecteur unidimensionnel et applique des poids et des biais pour générer une sortie de taille fixe. Cette couche est importante pour combiner les caractéristiques extraites et formuler des prédictions basées sur celles-ci.
      * Une Couche Dense : Enfin, nous employons une couche dense avec une fonction d'activation softmax (ou une autre fonction adaptée) pour classifier les images en différentes catégories de maladies pulmonaires (telles que COVID-19, pneumonie, etc.). Cette couche fournit les probabilités associées à chaque classe.

**Résumé de l'Architecture : L'architecture CNN que nous avons implémentée se compose donc de :**

* Trois couches de convolution
* Trois couches de MaxPool
* Une couche Flatten
* Une couche totalement connectée
* Une couche dense

Cette architecture a été choisie pour son efficacité à capturer et à interpréter les caractéristiques complexes présentes dans les radiographies pulmonaires, permettant ainsi une reconnaissance précise des maladies.

La reconnaissance des maladies pulmonaires par radiographie thoracique à l'aide de réseaux de neurones convolutifs (CNN) représente une avancée majeure dans le diagnostic médical assisté par ordinateur. Ce rapport explore une architecture CNN pour détecter des pathologies comme le COVID-

19 à partir de radiographies thoraciques. Notre modèle, avec ses couches de convolution, de MaxPooling, Flatten, totalement connectée et dense, a démontré une grande capacité à extraire et interpréter les caractéristiques complexes des images. Les CNN automatisent et standardisent le

diagnostic, réduisant la dépendance à l'expertise humaine et augmentant la rapidité et l'efficacité du dépistage. Finalement, l'application des CNN améliore significativement les pratiques diagnostiques actuelles, transformant potentiellement la détection et le traitement des maladies pulmonaires.

##### Conclusion

Ce chapitre a exploré les techniques de traitement d'image pour détecter les maladies pulmonaires, en soulignant l'importance des réseaux de neurones convolutifs (CNN). Le prétraitement, notamment l'égalisation d'histogramme, améliore la qualité des images, facilitant un diagnostic plus précis. Les caractéristiques des images, comme l'homogénéité, le bruit et la luminance, sont essentielles pour l'analyse médicale.

## CHAPITRE 3 : PRESENTATION DU PROJET

### Introduction

Ce chapitre examine l'utilisation des techniques d'imagerie médicale et des réseaux neuronaux convolutifs (CNN) pour la détection et la classification des maladies pulmonaires, notamment le COVID-19. En utilisant une base de données d'images médicales comprenant des radiographies thoraciques catégorisées en COVID, Normal et Virus, l'étude développe un modèle CNN. Ce modèle, après prétraitement par égalisation d'histogrammes, différencie efficacement les cas, aidant au diagnostic rapide et précis des patients.

### Présentation du Project

Ce projet explore l'application des techniques d'imagerie médicale et des réseaux neuronaux convolutifs (CNN) pour améliorer le diagnostic des maladies pulmonaires, avec un accent particulier sur le COVID-19 et d'autres infections virales. En utilisant une base de données d'images médicales, comprenant des radiographies thoraciques classées en trois catégories principales (COVID, Normal, Virus), nous avons développé un modèle d'apprentissage profond capable de différencier ces conditions.

L'émergence de la COVID-19, avec sa propagation rapide et ses effets graves, surtout chez les personnes âgées et celles atteintes de maladies chroniques, a incité les chercheurs à chercher des solutions efficaces. Les radiographies thoraciques sont devenues un outil essentiel, car le virus cible principalement les poumons, causant des problèmes respiratoires graves.

Le projet s'appuie sur l'apprentissage profond, en particulier les CNN, pour extraire et classifier des caractéristiques complexes des images médicales. Ces réseaux sont constitués de couches convolutionnelles et de pooling, permettant de traiter efficacement les données d'image. Les techniques CNN sont déjà utilisées dans diverses applications médicales comme la détection de tumeurs cérébrales, de cancer de la prostate, de nodules pulmonaires et de cancer du sein.

Pour ce projet, nous avons utilisé une base de données de radiographies thoraciques COVID, prétraitées par égalisation des histogrammes pour améliorer la qualité des images. Les données ont été divisées en ensembles d'entraînement et de test. Notre architecture CNN a démontré une capacité

élevée à classer correctement les images dans l'une des trois catégories, aidant ainsi le personnel médical à diagnostiquer rapidement et efficacement les patients, réduisant le temps de diagnostic.

##### Objectifs d’application Énoncé du problème

Comment améliorer la précision du diagnostic des maladies pulmonaires, et plus spécifiquement du COVID-19, en utilisant des techniques d'imagerie médicale et des réseaux neuronaux convolutifs pour différencier ces maladies de celles ayant des symptômes similaires, comme la grippe ?

##### Acquisition de données

La base de données des images radiographie pulmonaire COVID-QU, a été utilisée pour former notre modèle. Elle contient 1823 images réparties sur trois catégories positive au COVID-19, Normale, pneumonie virale. Comme suite quelques images de la base de données :



Figure 3.1 Quelques images de l’ensemble de données utilisées, (a) cas positifs au COVID-19, (b) Normale, (c) Viral Pneumonie

* + 1. **Préparation des données**

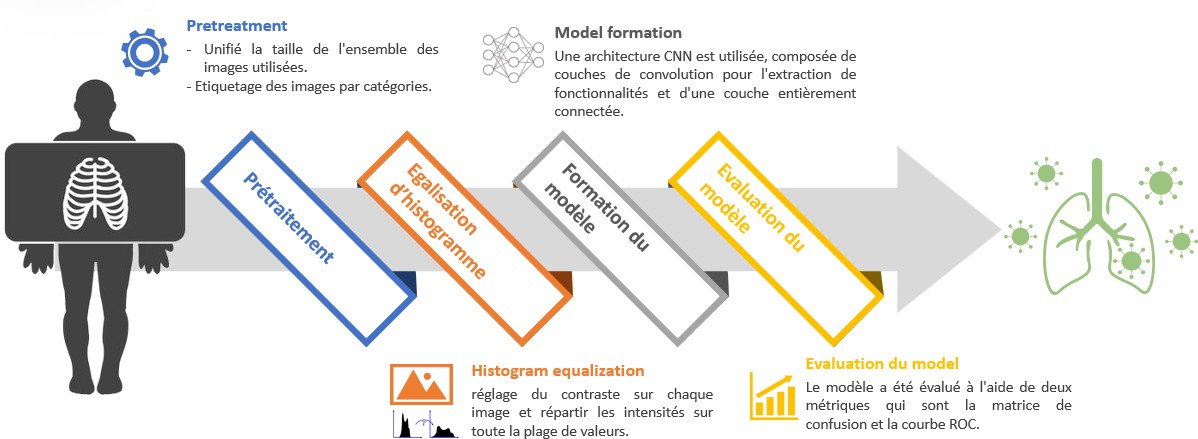


Figure 3 .2 Étapes de Prétraitement et d'Analyse d'Images Radiographiques Pulmonaires pour le Diagnostic du COVID-19 via un Réseau Neuronal Convolutif

L’ensemble des images de la base de données est réparti comme suit :

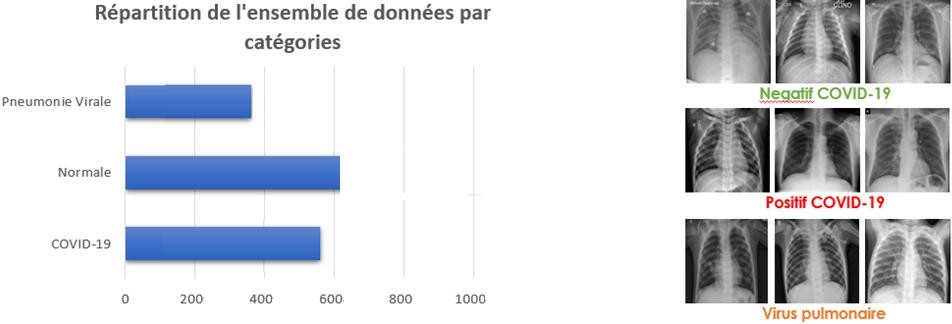


Figure 3.3. Répartitions de l’ensemble des images de la base de données par catégories

**Préparation du jeu de données utilisé**

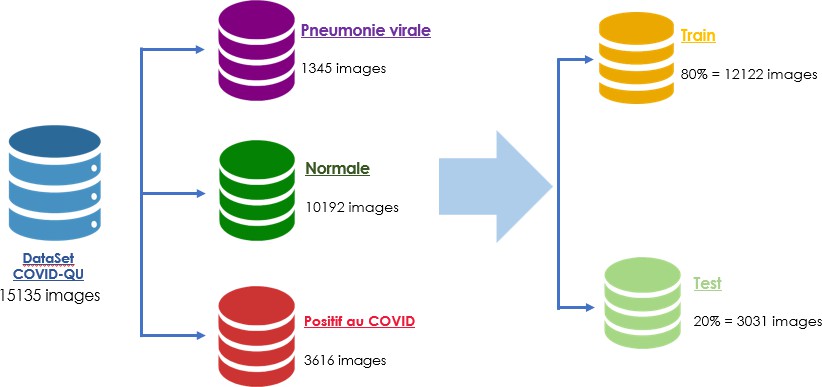


Figure 3 .4 Répartition des Données en Groupes d'Entraînement et de Test pour la Modélisation du Diagnostic COVID-19

##### Opération d’égalisation d’histogramme :

Pour améliorer la qualité de l'ensemble des images, une opération d'égalisation d'histogramme a été appliquée afin d'obtenir une meilleure répartition du contraste sur toutes les images.

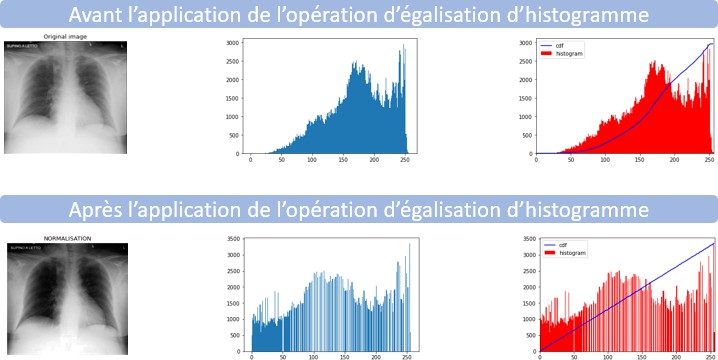


Figure 3 .5 Amélioration de la Qualité des Images par Égalisation d'Histogramme pour une Répartition Optimale du Contraste

##### Importance de la détection des maladies pulmonaires

Ce projet a pour objectif de fournir un outil de diagnostic assisté par ordinateur pour aider les professionnels de la santé dans l'interprétation des images radiographiques, avec un accent particulier sur la détection précoce des maladies respiratoires, y compris le COVID-19. En sensibilisant à l'utilisation de l'intelligence artificielle, le projet vise à améliorer la rapidité et la précision des diagnostics grâce à l'automatisation basée sur l'IA. Il offre une solution accessible et conviviale en utilisant des technologies modernes de développement logiciel. En exploitant efficacement les données médicales, le projet soutient la recherche et le développement de solutions médicales avancées. Il facilite également l'accès aux outils de diagnostic pour les établissements de santé et les professionnels médicaux. Cette proposition met en avant la conception d'une solution innovante combinant intelligence artificielle, développement logiciel et application médicale, visant à améliorer les capacités de diagnostic des maladies respiratoires. La méthode pourrait être étendue à d'autres pathologies pulmonaires, augmentant ainsi son utilité et sa fiabilité dans le domaine médical.

#### Méthodologie et outil de travail

Dans le chapitre précédent, nous avons abordé le Deep Learning et les réseaux de neurones convolutifs (CNN), en détaillant leurs architectures. Ce chapitre se concentre sur la mise en place d'un

système basé sur l'apprentissage profond pour la segmentation et la classification des images pulmonaires. Nous avons introduit des modèles CNN, entraînés pour classer les images de manière indépendante. De plus, nous décrirons notre ensemble de données, appelé dataset, ainsi que les outils et méthodes employés pour cette implémentation.

##### Environnement du travail

Dans cette partie, nous présentons les outils matériels et logiciels utilisés dans notre travail.

###### Environnement matériel :

Pour ce projet, nous avons utilisé du matériel doté de caractéristiques spécifiques, notamment des processeurs puissants, des GPU pour l'entraînement des modèles, et une grande capacité de stockage.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom de PC** | **Processeur** | **Mémoire (RAM)** | **Carte**  **Graphique** | **Système**  **D’exploitation** |
| HP Ordinateur portable HP 15s- fq2000nk (2U2M3EA) | Intel® Core™ i7- 1165G7 (jusqu’à 4,7 GHz avec la technologie Intel® Turbo Boost, mémoire cache L3  12 Mo, 4 cœurs) | Taille : 12 GO | Intel® Iris® XᵉIntégré | Microsoft Windows 11(64bits). |

Tableau 3 .1 : Caractéristique de matériel de programmation

###### Langage de programmation Python

Python est un langage de programmation de haut niveau interprété, conçu pour des usages généraux. Créé par Guido van Rossum et annoncé pour la première fois en 1991, Python est réputé pour sa lisibilité grâce à l'utilisation d'espaces significatifs. Il est l'un des langages open source les plus populaires, largement utilisé pour la gestion d'infrastructure, l'analyse de données et le développement de logiciels. Il existe deux versions principales : Python 2 et Python 3, la version actuelle, qui offre un interpréteur plus performant et un meilleur contrôle de la concurrence. Python 3 est compatible avec les systèmes Mac, Windows et Unix.



Figure 3.6 : Logo python

###### Navigateur Anaconda

Le Navigateur Anaconda est une interface graphique (GUI) incluse dans la distribution Anaconda. Il permet aux utilisateurs de lancer des applications, gérer les librairies conda, les environnements, et les canaux sans avoir besoin d'utiliser des lignes de commande. Le Navigateur peut aussi accéder aux librairies du Cloud Anaconda ou d'un dépôt Anaconda local pour les installer, les exécuter, et les mettre à jour. Il est disponible pour Windows, macOS et Linux.

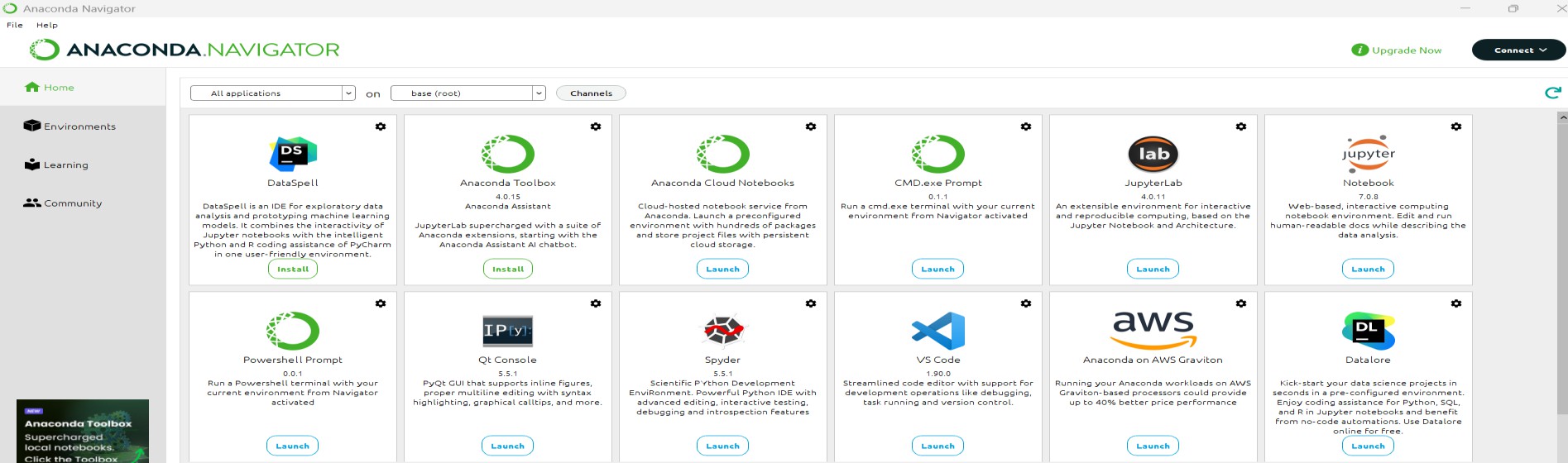


Figure 3.7 Navigateur Anaconda

###### Spyder

Spyder est un environnement de développement intégré (IDE) inclus dans la distribution Anaconda. Conçu spécifiquement pour les scientifiques, les ingénieurs et les analystes de données, Spyder offre une interface utilisateur riche et intuitive. Il permet d'éditer, d'analyser et de visualiser des données, ainsi que de gérer des librairies et des environnements conda sans utiliser de ligne de commande.

Spyder est compatible avec les systèmes Windows, macOS et Linux, et facilite l'accès aux librairies et aux outils scientifiques présents dans l'écosystème Python.



Figure 3.8 L'environnement de développement scientifique Python

###### Keras

Keras est une API de réseau de neurones écrite en Python. Cette bibliothèque open source fonctionne sur TensorFlow et permet de créer facilement des couches pour les réseaux de neurones ainsi que de mettre en place des architectures complexes. Elle a été développée pour faciliter l'expérimentation rapide, la recherche avancée et la production. Keras est compatible avec les versions de Python 2.7, 3.6, et autres.



Figure 3.9 Logo Keras

###### Jupyter Notebook

Jupyter Notebook est une application web open source qui permet de créer et de partager du code, des équations, des visualisations et du texte. Il combine le code, les commentaires, le contenu multimédia et les visualisations dans un document interactif appelé "notebook". Ce document interactif peut être partagé, réutilisé et modifié, facilitant la collaboration et l'itération rapide.



###### TensorFlow

Figure 3.10 Logo Jupyter Notebook

TensorFlow est une plateforme open source d'apprentissage automatique développée initialement par des chercheurs de Google. C'est une bibliothèque de calcul symbolique utilisant le flux de données et la programmation différentiable pour accomplir diverses tâches, notamment l'entraînement et l'inférence de réseaux neuronaux profonds. Elle permet aux développeurs de créer des applications d'apprentissage automatique en utilisant une variété d'outils, de bibliothèques et de ressources communautaires.



Figure 3.11: Logo TensorFlow

###### NumPy

NumPy (Numerical Python) est une bibliothèque Python essentielle pour le calcul numérique. Elle offre des fonctions puissantes pour manipuler des matrices et des tableaux multidimensionnels, permettant ainsi de réaliser des opérations mathématiques complexes de manière rapide et efficace. NumPy est largement utilisée pour ses capacités à surpasser les performances des listes Python standard, rendant les opérations sur de grandes quantités de données beaucoup plus rapides. Elle constitue une base incontournable pour de nombreuses applications scientifiques et d'analyse de données en Python.



Figure 3.12: Logo NumPy

###### Pandas

Pandas est un package Python dédié à la manipulation et à l'analyse de données. Son nom est dérivé de "Panel Data". Pandas introduit deux structures de données supplémentaires : Séries et DataFrame. Ces structures facilitent le travail avec des données étiquetées et relationnelles de manière simple et intuitive, rendant l'analyse et la manipulation des données beaucoup plus efficaces et conviviales. Pandas est un outil essentiel pour les scientifiques des données et les analystes travaillant en Python.



Figure 3.13: Logo Pandas

###### Matplotlib

Matplotlib est une bibliothèque Python utilisée pour créer des graphiques et visualiser des données. Elle permet de représenter visuellement les résultats de manière claire et concise. Idéale pour comparer des images et analyser des données, Matplotlib se combine efficacement avec d'autres bibliothèques scientifiques comme NumPy et SciPy. Elle est largement utilisée pour générer des visualisations diverses, facilitant l'interprétation et la communication des données analytiques.

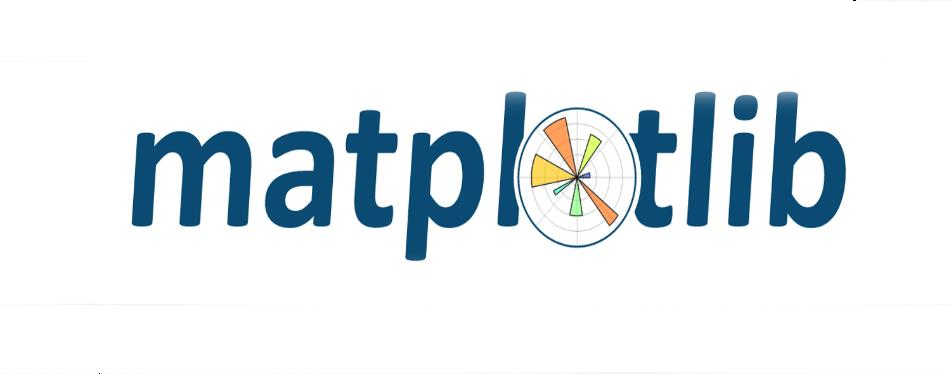


Figure 3.14: Logo Matplotlib

###### Glob

Glob est une bibliothèque Python utilisée pour trouver tous les chemins de fichiers correspondant à un motif spécifié, comme les expressions régulières. Elle permet de rechercher des fichiers et des dossiers dans le système de fichiers en utilisant des caractères génériques. Glob est particulièrement utile pour automatiser le traitement de fichiers en lot, facilitant ainsi la gestion et l'accès à de nombreux fichiers selon des critères spécifiques.



###### Seaborn

Figure 3.15: Logo Glob

Seaborn est une bibliothèque de visualisation construite sur Matplotlib, axée sur la création de graphiques statistiques standards avec un simple appel de fonction. Bien qu'elle offre moins d'options de personnalisation que Matplotlib, les attributs des graphiques générés avec Seaborn peuvent toujours être modifiés en utilisant des fonctions de Matplotlib. Cela permet de combiner la simplicité d'utilisation de Seaborn avec la flexibilité de Matplotlib pour des visualisations avancées.



Figure 3.16: Logo Glob

###### GanttProject

GanttProject est un logiciel libre de gestion de projet développé en Java, compatible avec divers systèmes d'exploitation tels que Windows, Linux et MacOS. Il permet de créer et d'éditer des diagrammes de Gantt. Cet outil facilite la planification de projets en offrant la possibilité de réaliser des diagrammes de Gantt, des diagrammes de ressources et des réseaux PERT.

#### Conclusion

Dans ce chapitre, nous avons présenté notre projet, ses objectifs, et la méthodologie utilisée, en détaillant les outils de travail. L'accent a été mis sur l'utilisation des réseaux neuronaux convolutifs (CNN) et de l'apprentissage profond pour segmenter et classifier les images pulmonaires, notamment pour détecter des maladies comme le COVID-19. Nous avons décrit notre ensemble de données et les outils matériels et logiciels employés, tels que Python, Jupyter Notebook, TensorFlow, Keras, NumPy, Pandas, Matplotlib, Seaborn, et Glob. De plus, un diagramme de Gantt a été utilisé pour planifier et suivre les différentes étapes du projet. Notre approche vise à améliorer la précision et la rapidité des diagnostics médicaux en exploitant l'intelligence artificielle et en offrant des solutions accessibles pour le secteur médical.

## CHAPITRE 4 : CONCEPTION ET MISE EN PLACE D’APPLICATION DE DETECTION DES MALADIES PULMONAIRES BASÉE SUR LE TRAITEMENT DES IMAGES

#### Introduction

Dans le chapitre précédent, nous avons abordé le Deep Learning et les réseaux de neurones convolutifs (CNN) ainsi que leurs architectures. Ce chapitre se concentre sur la mise en place d'un système basé sur l'apprentissage profond pour la segmentation et la classification d'images pulmonaires. Nous avons présenté des modèles CNN entraînés pour classifier les images. Nous décrivons également notre ensemble de données et les outils utilisés. Le processus inclut le prétraitement des images, la création et l'entraînement de notre modèle, suivi de tests réels avec une application Python Tkinter pour évaluer des cas concrets et afficher les résultats

#### Application du modèle sur la base de données

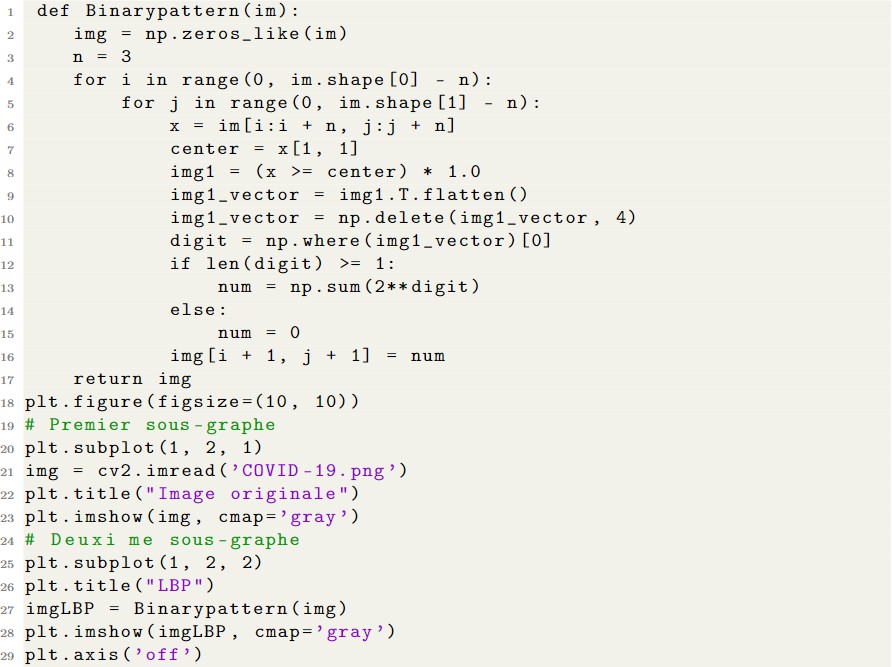
##### Préparation des données.

Dans notre étude, nous utilisons une base de données contenant 1 823 images radiographiques de taille 512 x 512, réparties en trois classes : 536 images de COVID, 668 images normales et 619 images de virus pulmonaires. Ces images sont utilisées pour la segmentation et la classification des maladies pulmonaires.

La base de données Ensemble de données COVID-QU a été obtenue à partir de la plateforme open source **Kaggle/dataset**, et permet de développer et tester notre modèle de détection basé sur l'apprentissage profond.

##### Prétraitement des images

Le "**Binary Pattern**" ou "modèle binaire" est une technique de traitement d'image utilisée pour la reconnaissance de motifs, l'analyse de textures et d'autres applications liées à l'image. L'un des exemples les plus connus est le "Local Binary Pattern" (LBP). Voici une explication détaillée :



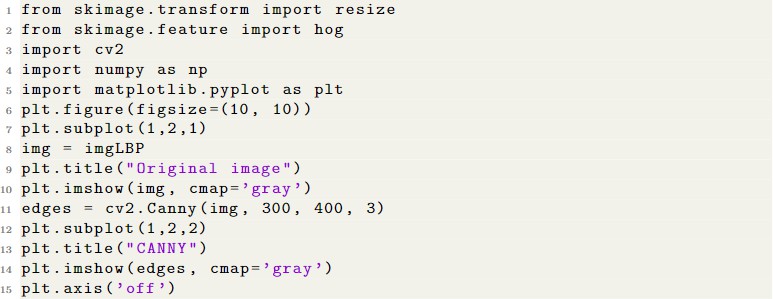
Listing 4.1 – code traitement d'image utilisée pour la reconnaissance de motifs.

Ce code utilise diverses bibliothèques pour traiter et afficher des images. Il définit une fonction **Binarypattern** qui génère un motif binaire local pour une image donnée. Ensuite, il lit une image nommée 'COVID-19.png', affiche l'image originale et le motif binaire local (LBP) en nuances de gris, en utilisant Matplotlib pour la visualisation dans une figure de taille 10x10 pouces.



**Code**

Figure 4.1: Résultat d’application de la fonction Binarypattern du sur une image



Listing 4.2 – code d’application de l’algorithme Canny.

Ce code redimensionne et traite une image pour détecter les contours en utilisant l'algorithme Canny. Après avoir importé les bibliothèques nécessaires, il affiche l'image originale et les contours détectés dans une figure Matplotlib. L'image est chargée et redimensionnée, puis les contours sont détectés et affichés en niveaux de gris.

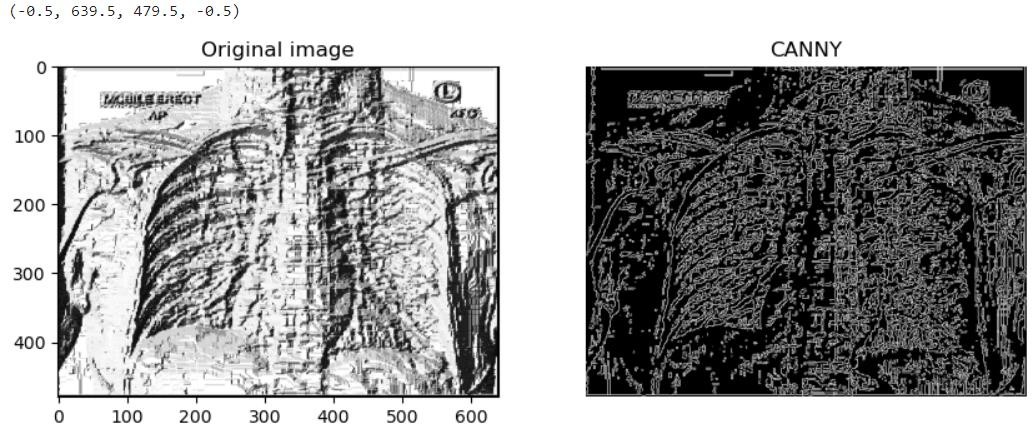
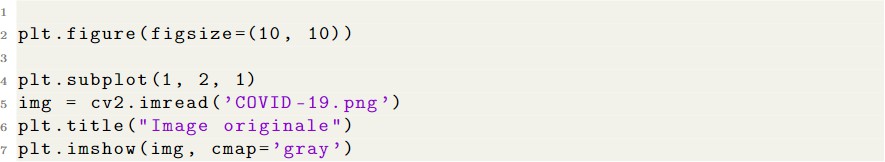


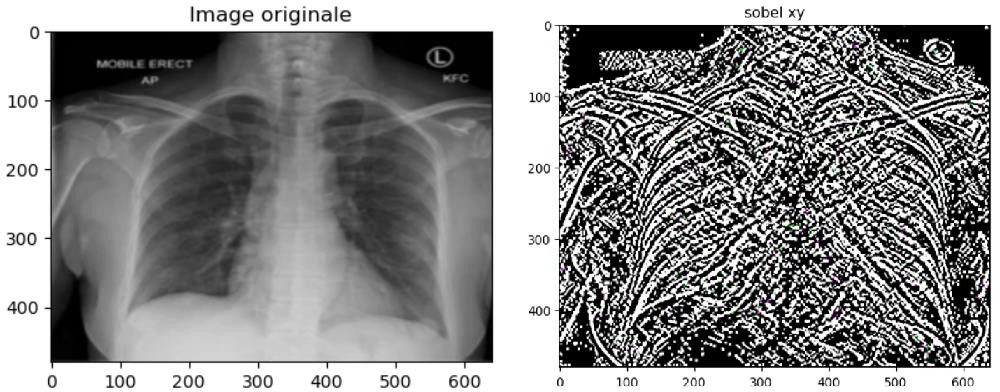
Figure 4.2: Résultat d’application de l’algorithme Canny du sur une image

###### Code



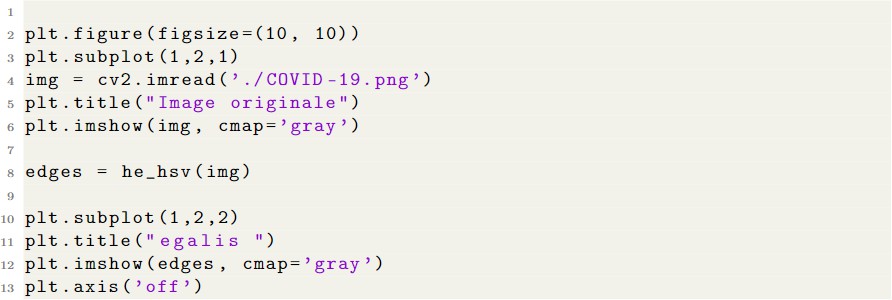
Listing 4.3 code d’application du filtre Sobel.

Ce code crée une figure de 10x10 pouces avec deux sous-graphes pour afficher une image originale et son résultat après application du filtre **Sobel** pour la détection des contours horizontaux.



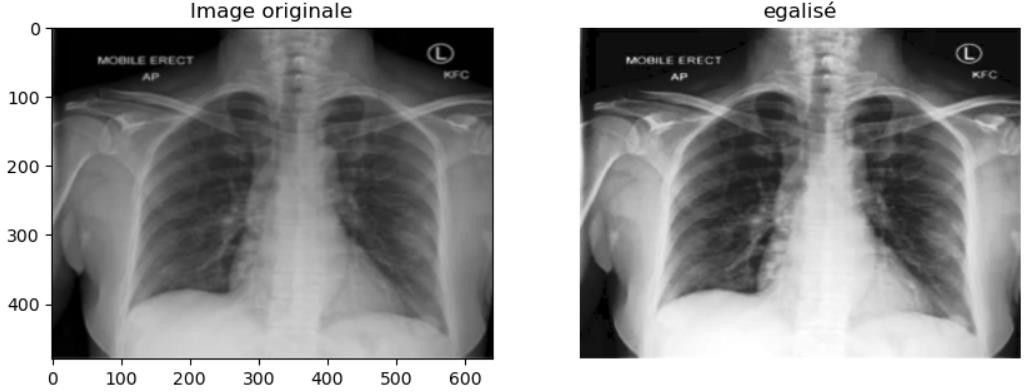
###### Code

Figure 4.3: Image origine et Image avec le filtre de Sobel XY



Listing 4.4 code de comparaison visuellement l'image originale avec les bords obtenus

Ce code permet de comparer visuellement l'image originale avec les bords obtenus après l'application de la fonction he\_hsv, facilitant ainsi l'analyse des résultats.



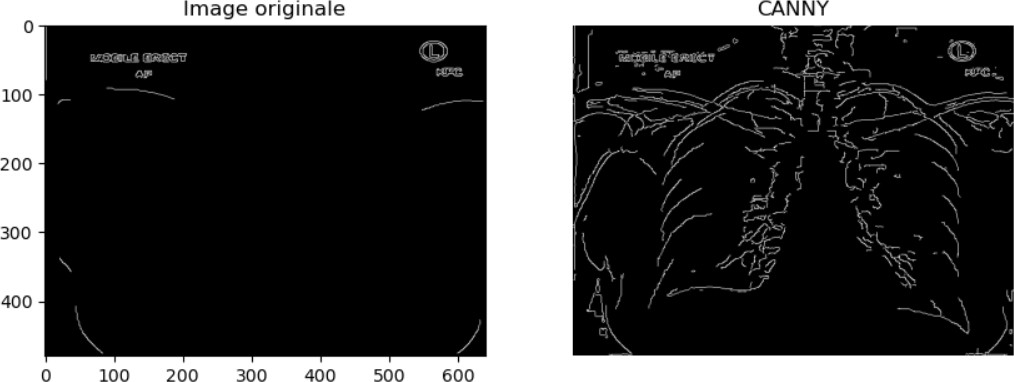
###### Code

Figure 4.4: Image origine et Image après l'application de la fonction he\_hsv



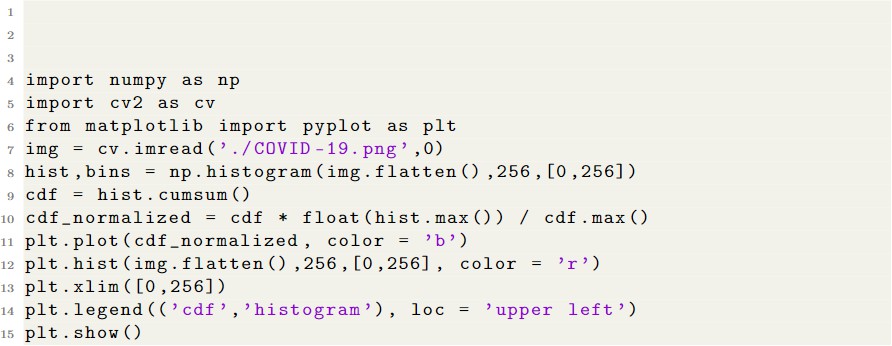
Listing 4.5 code de création de figure de 10x10 pouces avec l’affichage d’ image transformée par l'algorithme Canny

Ce code crée une figure de 10x10 pouces avec deux sous-graphes pour afficher une image originale transformée par l'algorithme Canny et une autre application de Canny pour la détection des bords.



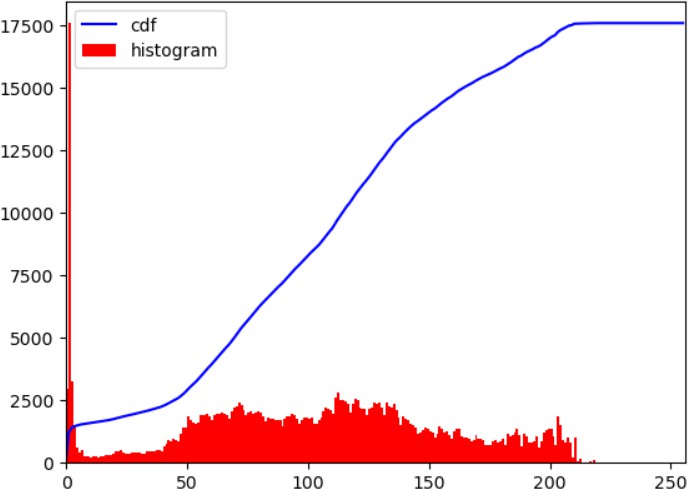
###### Code

Figure 4.5: Image origine et Image après la transformation par application de Canny



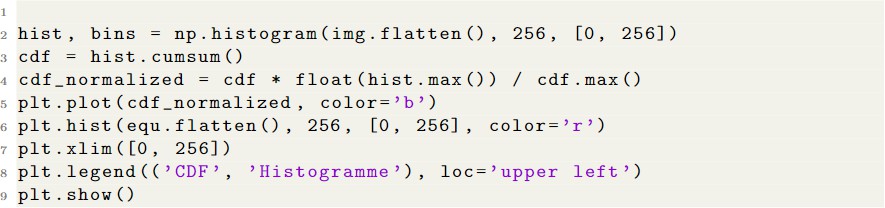
Listing 4.6 code d’utilisation de OpenCV et Matplotlib pour afficher l'histogramme d'une image

Ce code utilise OpenCV et Matplotlib pour afficher l'histogramme d'une image en niveaux de gris et sa fonction de distribution cumulée (CDF).



###### Code

Figure 4.6: Histogramme et CDF d’une image origine.



Listing 4.7 code histogramme et sa fonction de distribution cumulée

Ce code charge une image, calcule et affiche son histogramme et sa fonction de distribution cumulée (CDF) en utilisant Matplotlib. L'histogramme montre la distribution des niveaux de gris, tandis que la CDF montre la distribution cumulative des valeurs de pixels. La figure résultante permet de visualiser et de comparer ces deux distributions.

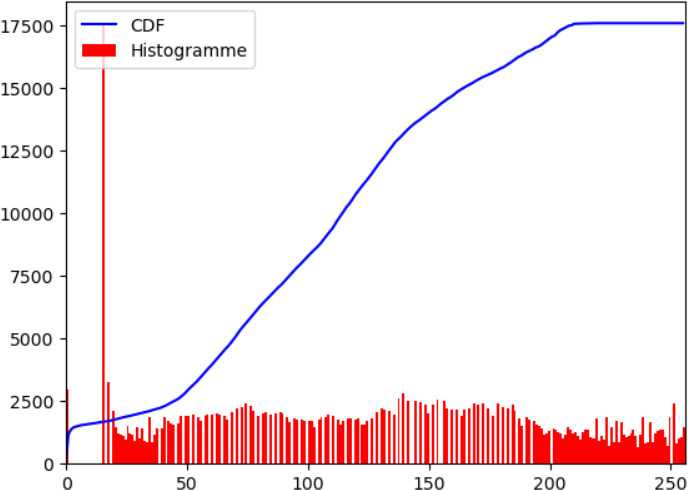


Figure 4.7: Histogramme et CDF d’une image après l’égalisation.

Résultat final du traitement de notre image :

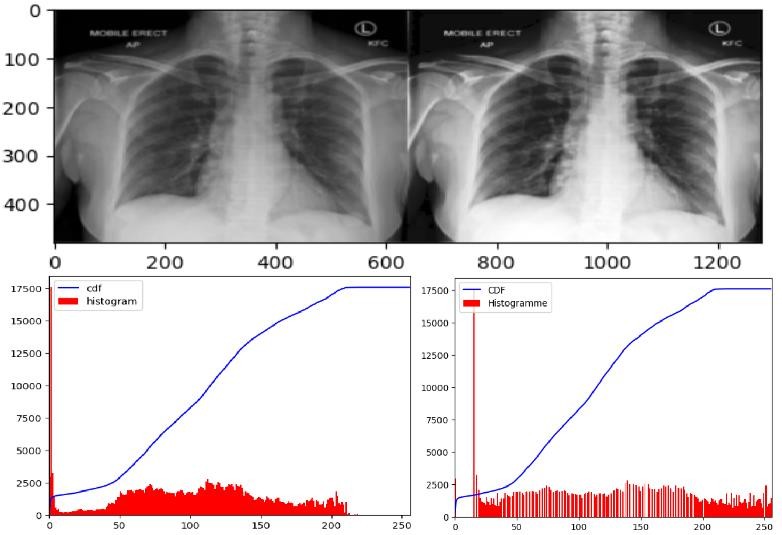


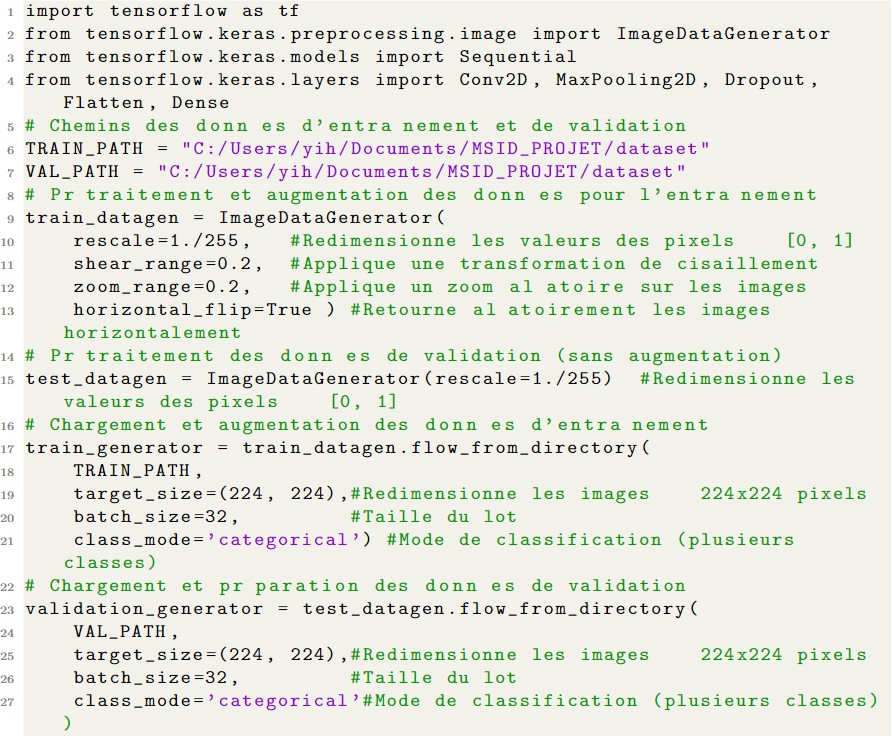
Figure 4.8: Histogramme cumulatif avant égalisation et Histogramme cumulatif après égalisation

Nous avons appliqué l'opération d'égalisation d'histogramme sur une image radiographique pulmonaire. En regardant l'histogramme cumulatif avant et après égalisation, nous pouvons voir que l'intensité a été distribuée sur toute l'image.

###### Entraînement et enregistrement d’un modèle de réseau de neurones convolutifs (CNN) pour la classification d'images.

Après la phase de prétraitement, nous avons formé deux ensembles de données : "Train" pour l'apprentissage du modèle et "Test" pour sa validation. Ensuite, nous avons développé une architecture CNN pour générer notre modèle.

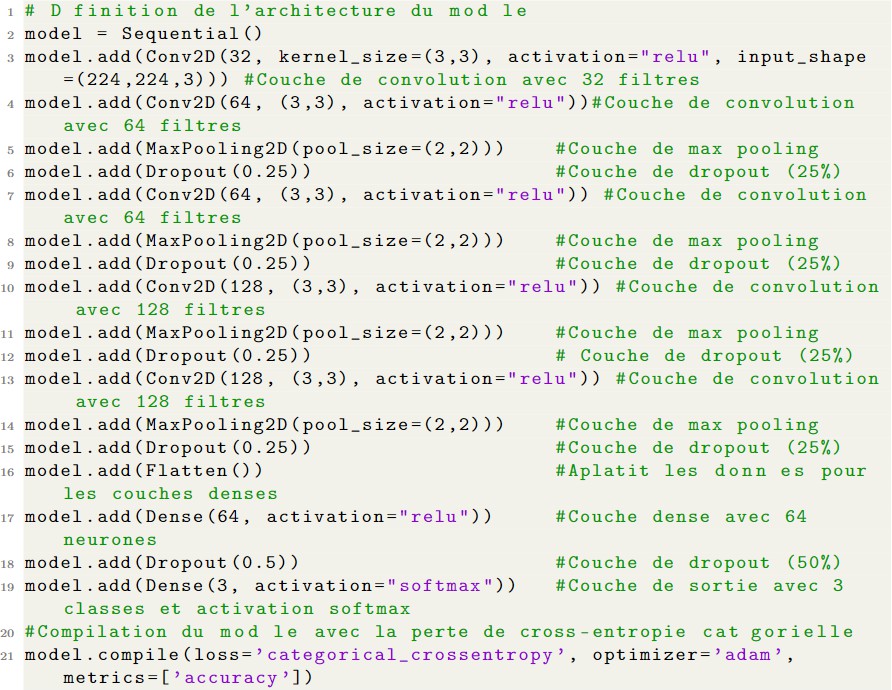
###### Code



Listing 4.8 code d’utilisant TensorFlow et Keras pour l’entrainement du modèle

Ce code prétraite et augmente les images pour l'entraînement d'un modèle de réseau de neurones convolutifs (CNN) en utilisant TensorFlow et Keras. Les images sont redimensionnées, et divers effets comme le cisaillement, le zoom et le retournement horizontal sont appliqués pour l'augmentation des données d'entraînement. Les images de validation sont également redimensionnées sans augmentation.

###### Code de l’architecture du modèle



Listing 4.9 code pour la classification d'images en utilisant TensorFlow et Keras

Ce code définit, entraîne, affiche le résumé et enregistre un modèle de réseau de neurones convolutifs (CNN) pour la classification d'images en utilisant TensorFlow et Keras.

###### Définition de l'architecture du modèle

Convolutional Layers (Conv2D) : Ces couches extraient les caractéristiques de l'image. Les filtres (32, 64, 128) appliquent des convolutions pour capturer des motifs à différentes échelles.

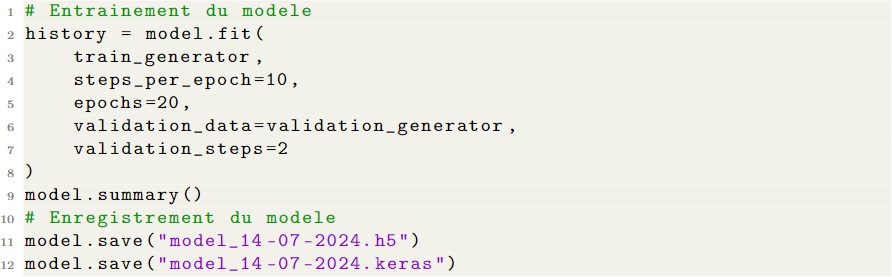
MaxPooling Layers (MaxPooling2D) : Ces couches réduisent la dimension des caractéristiques pour diminuer la complexité computationnelle et le surapprentissage.

Dropout Layers (Dropout) : Ces couches préviennent le surapprentissage en désactivant aléatoirement une fraction (25% ou 50%) des neurones pendant l'entraînement.

Flatten Layer (Flatten) : Cette couche aplatie les données pour les rendre compatibles avec les couches denses.

Dense Layers (Dense) : Ces couches sont entièrement connectées, avec 64 neurones pour l'extraction des caractéristiques et une couche finale de 3 neurones pour la classification avec l'activation softmax. Compilation : Le modèle est compilé avec la perte categorical\_crossentropy et l'optimiseur adam.

###### Code Entraînement du modèle



Listing 4.10 code de sauvegarder le modèle entraîné dans deux formats différents : HDF5 et Keras.

* train\_generator : Générateur de données d'entraînement.
* steps\_per\_epoch : Nombre d'étapes (batches) à réaliser par époque.
* epochs : Nombre total d'époques pour l'entraînement.
* validation\_data : Générateur de données de validation.
* validation\_steps : Nombre d'étapes de validation par époque.

Et à la fin on va sauvegarder le modèle entraîné dans deux formats différents : HDF5 et Keras. Séparation des données en ensembles d'entraînement, de validation et de test

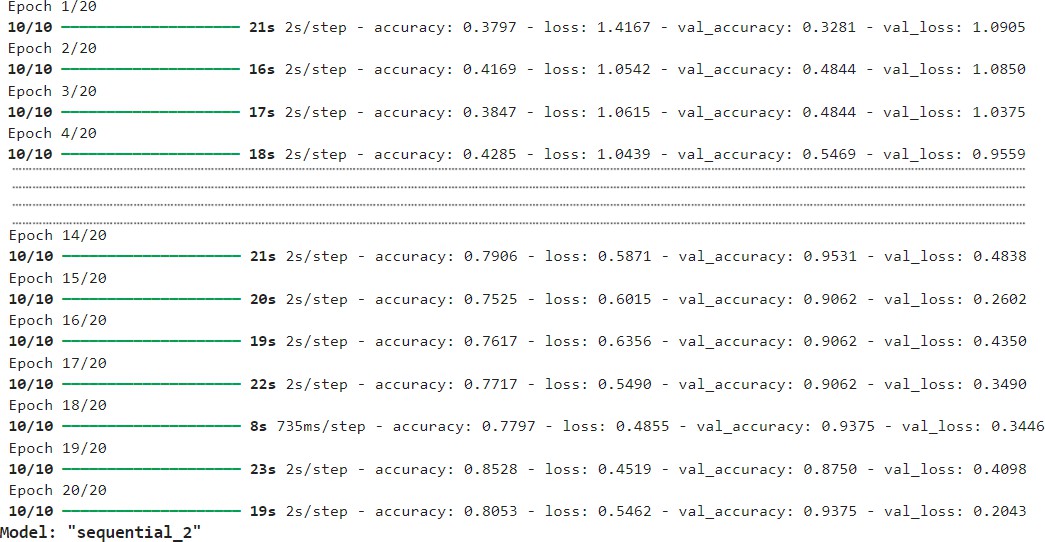


Figure 4.9: début de l’entrainement du model

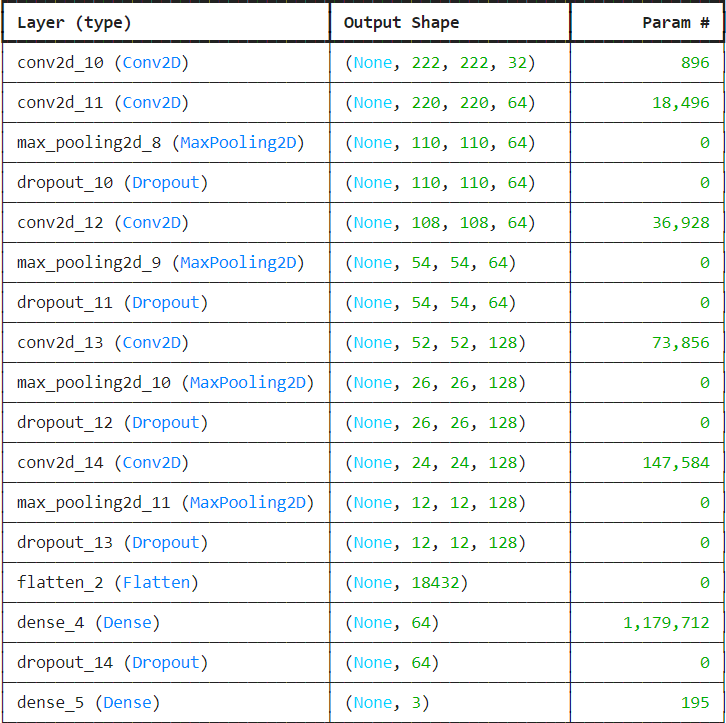


Tableau 4.1 résumé d'un modèle de réseau de neurones convolutifs (CNN) créé avec Keras

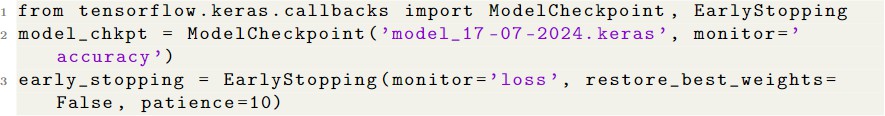
Ce résultat présente le résumé d'un modèle de réseau de neurones convolutifs (CNN) créé avec Keras. Chaque couche du modèle est détaillée avec sa forme de sortie (Output Shape) et le nombre de paramètres (Param #).

Le modèle est composé de plusieurs couches convolutives suivies de couches de max-pooling et de dropout, avec des couches denses pour la classification finale. Le nombre total de paramètres entraîne le modèle à détecter les caractéristiques spécifiques des images et à les classer en trois catégories.

##### Prédiction avec le modèle

Après l’enregistrement de notre modèle CNN préalablement entraîné On va charger le modèle et effectuer des prédictions sur de nouvelles images et affiche les résultats, en indiquant les probabilités de classification telles que COVID-19, normal, ou autre virus pulmonaire.

###### Code



Listing 4.11 code entraîne un modèle de réseau de neurones avec des callbacks.

Ce code entraîne un modèle de réseau de neurones avec des callbacks pour sauvegarder le modèle et arrêter l'entraînement de manière précoce si certaines conditions sont remplies.

* + - * ModelCheckpoint Objectif : Sauvegarder le modèle à chaque epoch si la performance surveillée (ici, la précision) s'améliore.

Paramètres :

filepath='model\_29-06-2024.keras' : Chemin et nom du fichier où le modèle sera sauvegardé. monitor='accuracy' : Métrique à surveiller pour décider de la sauvegarde du modèle.

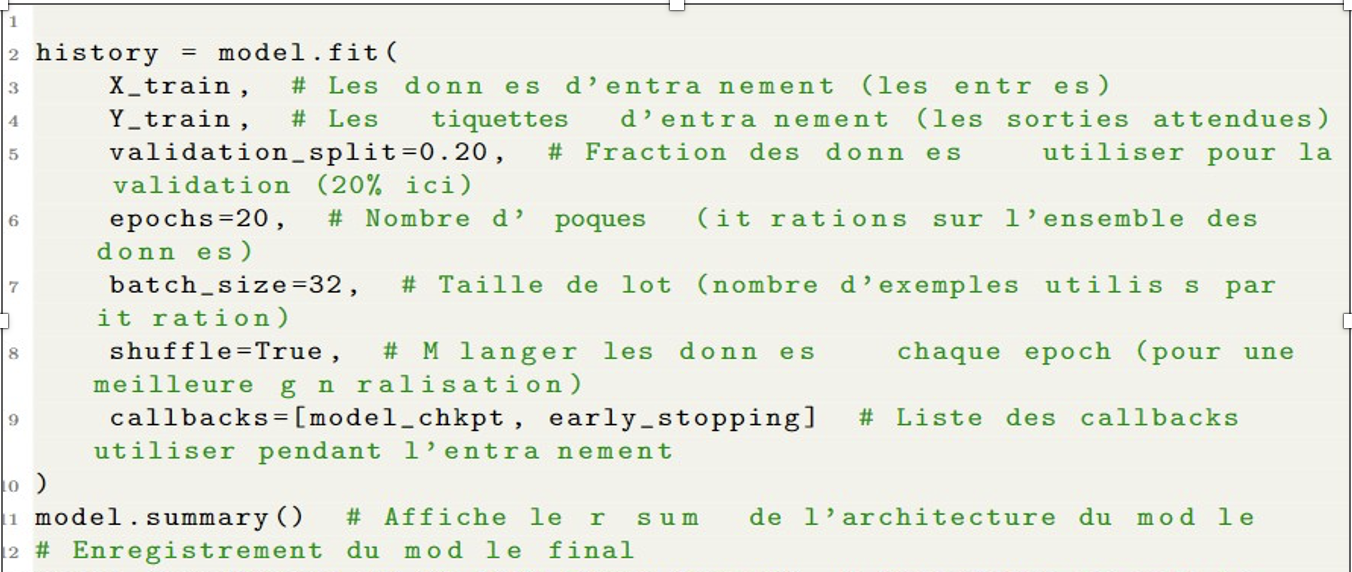
* + - * EarlyStopping Objectif : Arrêter l'entraînement prématurément si la performance du modèle ne s'améliore plus après un certain nombre d'epochs (patience).

Paramètres :

monitor='loss' : Métrique à surveiller pour décider de l'arrêt de l'entraînement. restore\_best\_weights=False : Ne pas restaurer les poids du meilleur modèle rencontré durant l'entraînement.

patience=10 : Nombre d'epochs sans amélioration après lesquelles l'entraînement sera arrêté.

Code



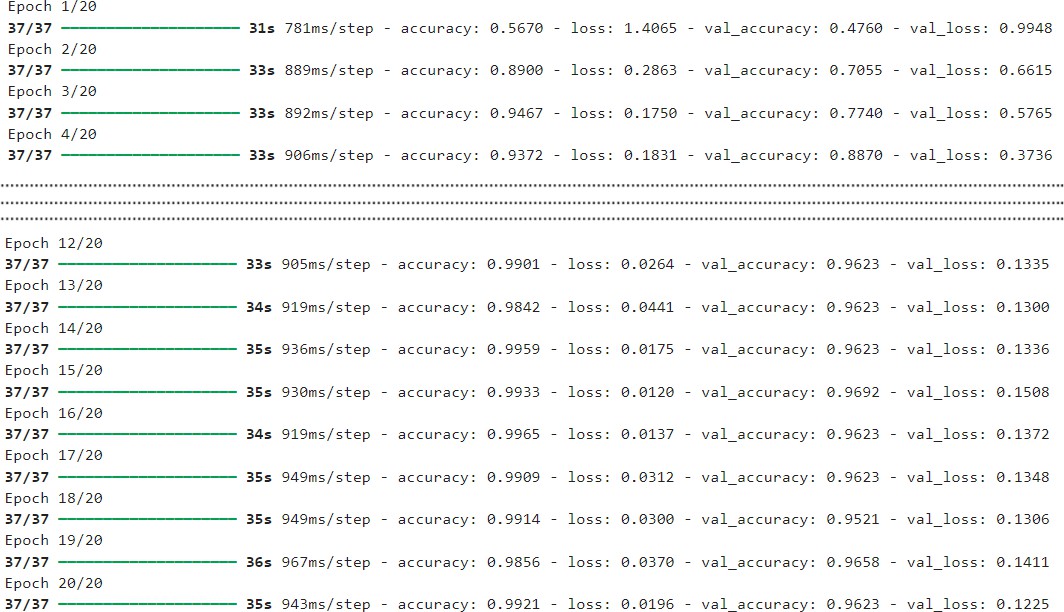
Listing 4.12 code validation un modèle de réseau de neurones.

Ce code entraîne un modèle de réseau de neurones sur des données d'entraînement avec une fraction des données utilisée pour la validation. Il utilise des callbacks pour sauvegarder le modèle lorsque la précision s'améliore et pour arrêter l'entraînement prématurément si la perte ne s'améliore plus après 10 epochs, permettant ainsi une gestion efficace du processus d'entraînement et évitant le surapprentissage.

###### Paramètres :

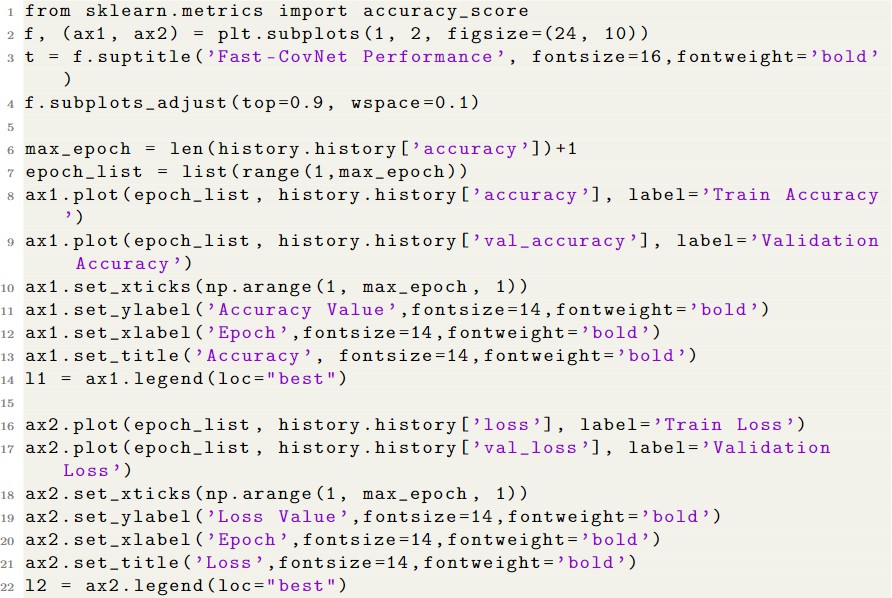
* + - * X\_train : Les données d'entraînement (entrées).
      * Y\_train : Les étiquettes d'entraînement (sorties attendues).
      * validation\_split=0.20 : Fraction des données à utiliser pour la validation (20% des données d'entraînement).
      * epochs=20 : Nombre total d'epochs pour l'entraînement.
      * batch\_size=32 : Nombre d'exemples par lot pour chaque itération.
      * shuffle=True : Mélanger les données à chaque epoch pour améliorer la généralisation.
      * callbacks=[model\_chkpt, early\_stopping]: Liste des callbacks à utiliser pendant l'entraînement : model\_chkpt : Sauvegarde du modèle si la précision s'améliore.

early\_stopping : Arrêt prématuré de l'entraînement si la perte ne s'améliore plus après 10 epochs.



**Code**

Figure 4.10: validation du model par 20% data



Listing 4.13 code Matplotlib pour visualiser les performances du modèle de réseau de neurones.

Ce code utilise Matplotlib pour visualiser les performances du modèle de réseau de neurones pendant l'entraînement, en affichant la précision et la perte pour les données d'entraînement et de validation. Deux sous-graphes sont créés : l'un pour la précision (Accuracy) et l'autre pour la perte (Loss). Les courbes montrent comment ces métriques évoluent pour les données d'entraînement et de validation au fil des epochs, aidant à évaluer la convergence et la généralisation du modèle.

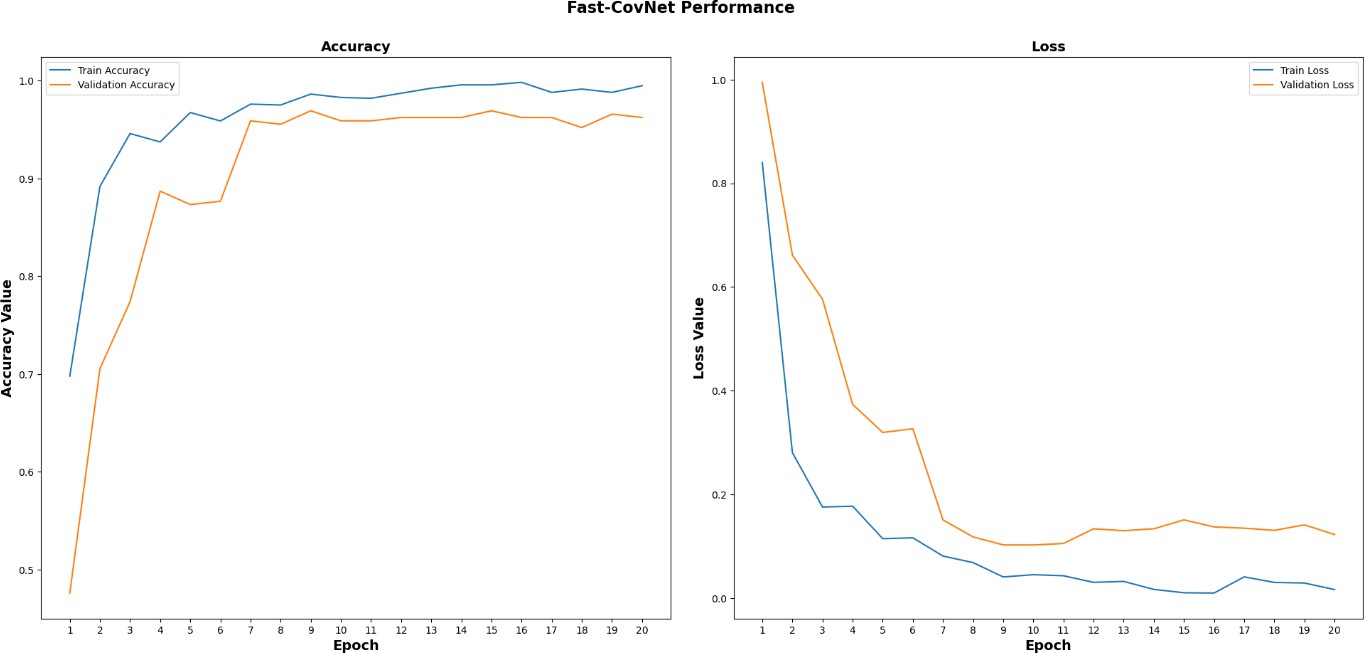
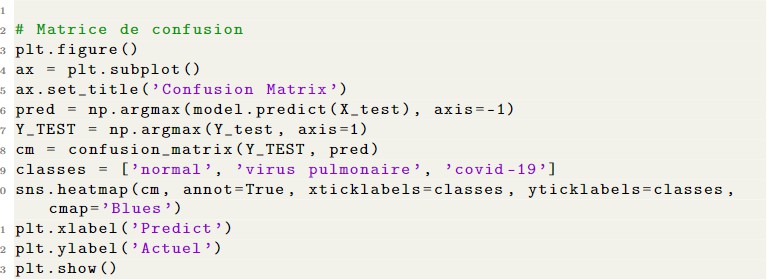


Figure 4.11: performances ConvNet rapides

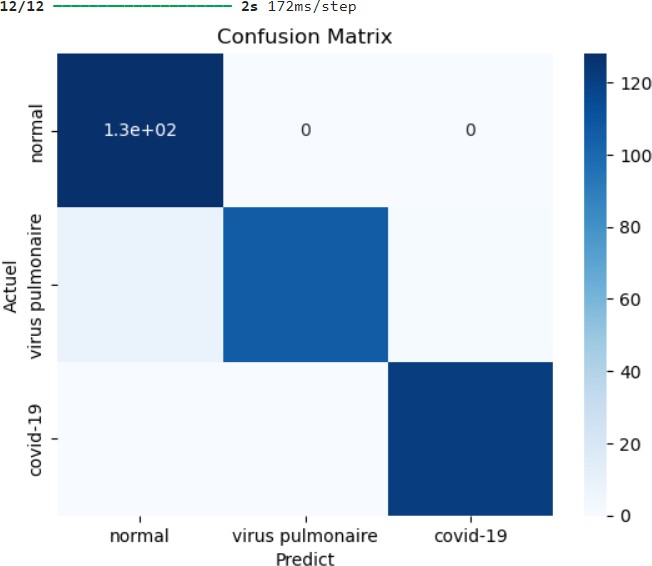
Donc en distingue que ces trois fichiers suivent une séquence logique dans le développement et l'application d'un modèle de réseau de neurones convolutifs (CNN) pour la classification d'images pulmonaires. Le premier fichier prépare les données, le deuxième développe et entraîne le modèle, et le troisième utilise le modèle pour effectuer des prédictions sur de nouvelles images. Ensemble, ils forment un pipeline complet de traitement, d'entraînement et de prédiction pour la détection des maladies pulmonaires.

**Code**



Listing 4.14 code Matrice de confusion.

Ce code produit une matrice de confusion pour visualiser la performance de notre modèle de classification. La matrice de confusion compare les prédictions du modèle aux valeurs réelles des données de test pour trois classes : 'normal', 'virus pulmonaire', et 'covid-19'. Les cellules annotées de la matrice montrent le nombre de prédictions correctes et incorrectes pour chaque classe. Un graphique heatmap est généré à l'aide de seaborn, avec des couleurs en nuances de bleu pour représenter les fréquences, et des étiquettes pour les axes pour indiquer les classes prédites et réelles.



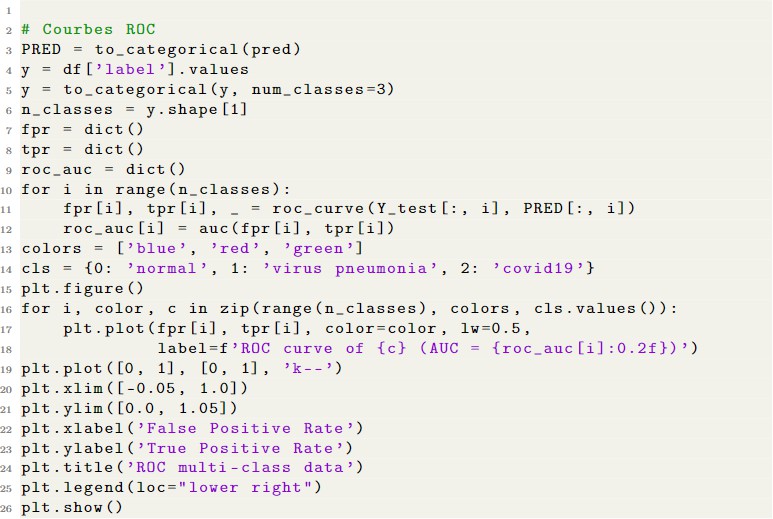
Analyse globale :

Le modèle montre une excellente performance pour la classe 'normal' avec des prédictions très précises.

La performance pour la classe 'covid-19' est également élevée mais pas parfaite.

Le modèle a le plus de difficulté avec la classe 'virus pulmonaire', ce qui pourrait indiquer un besoin d'amélioration pour mieux distinguer cette classe.

Pour améliorer les résultats, il pourrait être utile d'examiner les caractéristiques des images dans la classe 'virus pulmonaire' et d'ajuster ou d'affiner les techniques de prétraitement ou les hyperparamètres du modèle.



Listing 4.15 code Matrice de ROC pour visualiser les performances du modèle de réseau de neurones.

Ce code produit des courbes ROC pour trois classes, mettant en évidence les performances de votre modèle pour chaque classe spécifique. Les courbes ROC sont tracées pour les classes 'normal', 'virus pneumonia', et 'covid19', avec des couleurs respectives bleu, rouge, et vert. Les valeurs AUC (Area Under Curve) sont calculées et affichées sur le graphique pour chaque classe, indiquant une performance élevée de votre modèle, notamment pour la classe 'Covid-19' avec une AUC de 0.98, montrant une excellente capacité de distinction.

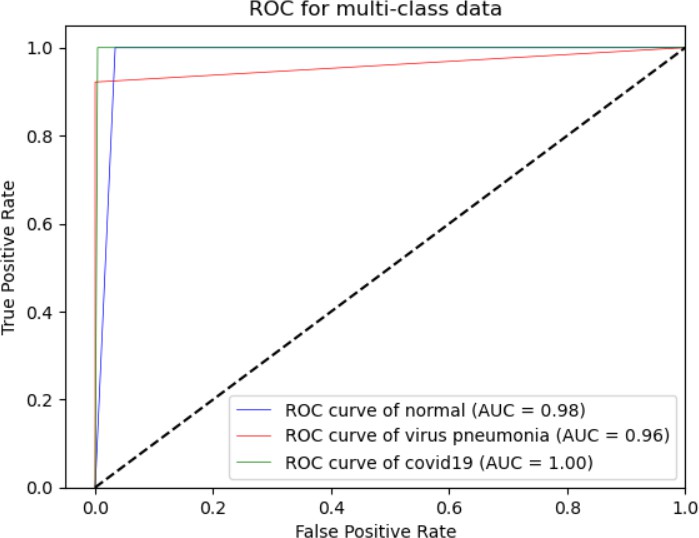


Figure 4.12 : la courbes de ROC CNN utiliser.

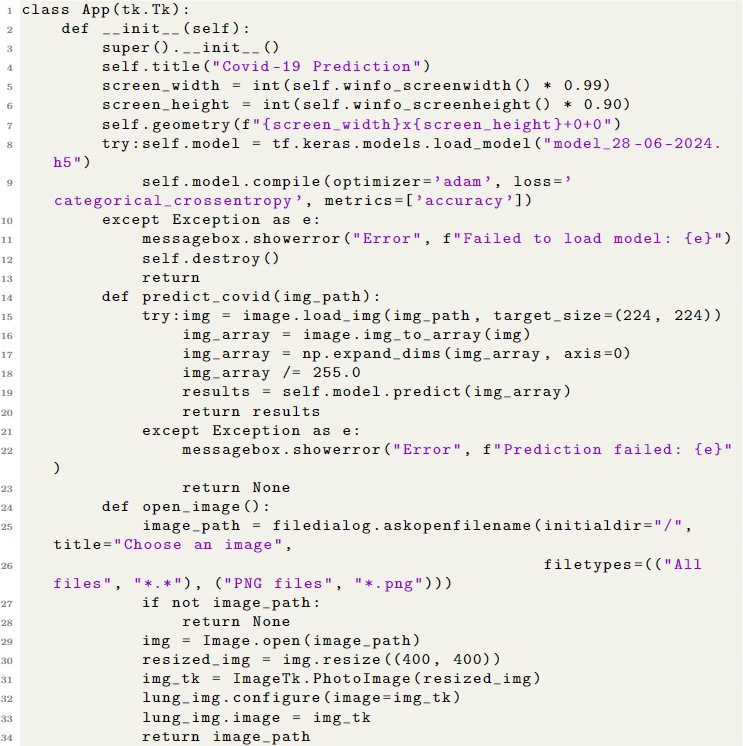
Dans votre graphique, l'interprétation des courbes ROC est la suivante : pour la classe 'Normal' (courbe bleue, AUC = 0.95), votre modèle distingue bien les images normales des anormales. Pour la classe 'Virus Pneumonia' (courbe rouge, AUC = 0.95), la performance est également très élevée. Enfin, pour la classe 'Covid-19' (courbe verte, AUC = 0.98), la performance est excellente, montrant une distinction efficace des images Covid-19.

##### Développement d’application

On va créer une interface graphique avec Tkinter pour charger et prédire des images radiographiques pulmonaires en utilisant un modèle CNN. Cette interface va nous permet de sélectionner une image, de faire une prédiction sur le statut COVID-19, et d'afficher le résultat avec une confiance associée.

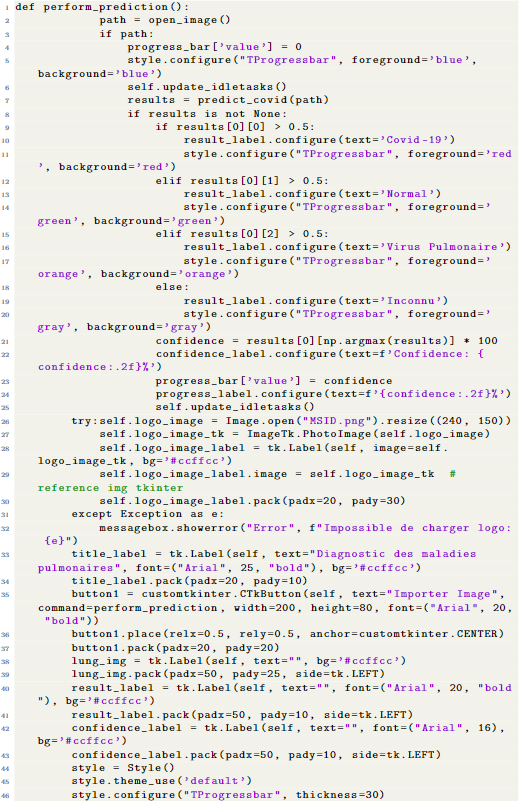
###### Implémentation de l'interface utilisateur (Tkinter, customtkinter)

Code crée pour notre application GUI avec Tkinter pour prédire le statut des maladies pulmonaire tel que Covid-19 à partir d'images radiographiques pulmonaires en utilisant un modèle de réseau de neurones convolutifs (CNN) préalablement entraîné.



Listing 4.16 code de création d’une interface graphique.

Ce code crée une interface graphique pour prédire le statut COVID-19 à partir d'images radiographiques pulmonaires en utilisant Tkinter. La classe App configure la fenêtre principale de l'application et charge un modèle de réseau de neurones convolutifs pré-entraîné avec TensorFlow. Une fonction permet de sélectionner une image via une boîte de dialogue, puis de la prétraiter avant d'effectuer une prédiction. Les résultats de la prédiction, incluant la probabilité de chaque classe (COVID-19, normal, autre virus pulmonaire), sont affichés à l'utilisateur. L'application permet également d'afficher une image du poumon et le logo de l'application.



Listing 4.17 exécute une prédiction COVID-19 à partir d'images radiographiques en utilisant Tkinter

Le code exécute une prédiction COVID-19 à partir d'images radiographiques en utilisant Tkinter. La fonction perform\_prediction() ouvre une image, effectue la prédiction avec le modèle et affiche le résultat et la confiance. Le logo de l'application est chargé et affiché. Des widgets d'interface, tels que des labels et des boutons, sont ajoutés pour l'interaction utilisateur. Enfin, l'application est lancée en instanciant la classe App et en démarrant la boucle principale Tkinter pour exécuter l'interface.

###### Enchaînement de l'Exécution

Démarrage : L'application démarre et affiche la fenêtre principale avec le titre et le logo.

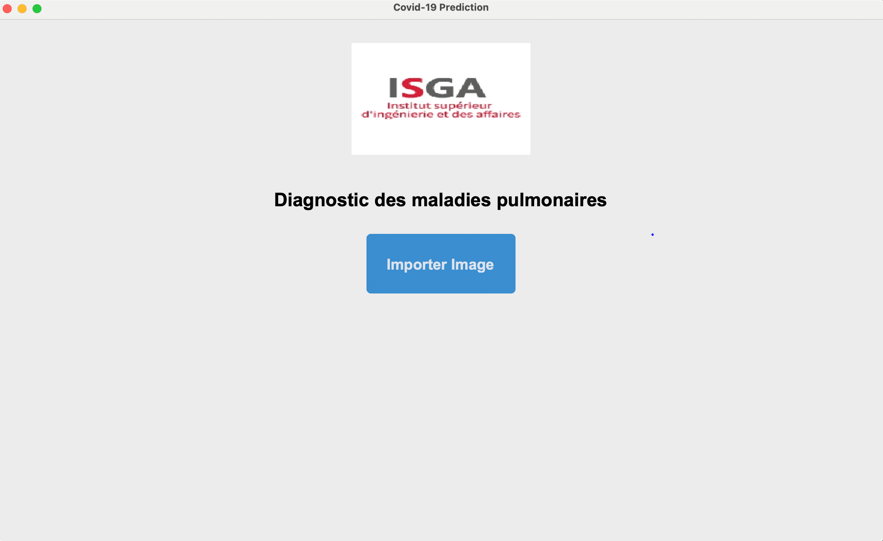


Figure 4.13 : Premier interface de l’application.

Sélection d'une Image : L'utilisateur clique sur le bouton "Importer Image".

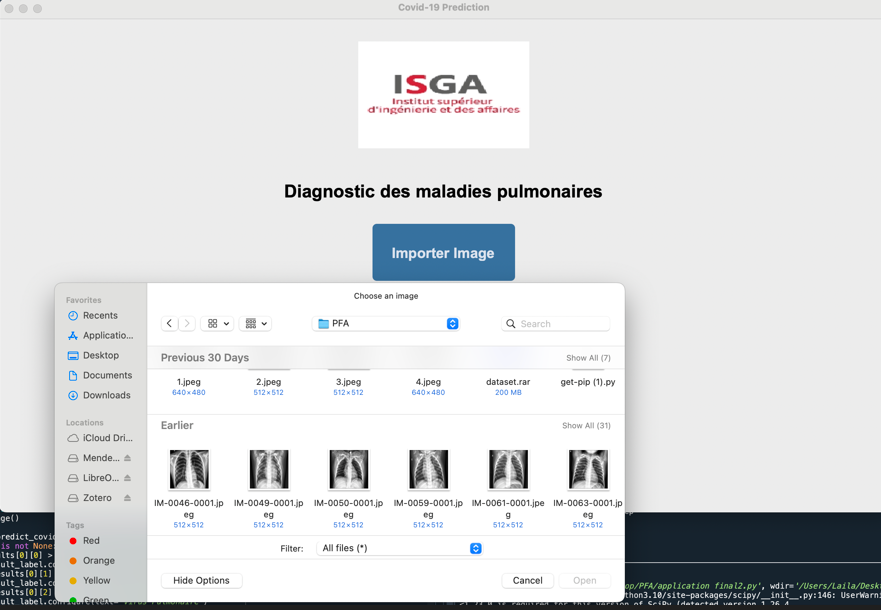


Figure 4.14 : Sélection d'une Image.

Ouverture de l'Image : Une boîte de dialogue s'ouvre pour permettre à l'utilisateur de sélectionner une image. L'image est redimensionnée et affichée.

Après avoir collectée et prétraitée l’image, ce qui inclut son redimensionnement pour correspondre aux dimensions d'entrée du modèle et sa normalisation pour ajuster les valeurs des pixels. Le modèle CNN préalablement entraîné est chargé à partir de son fichier de sauvegarde, avec son architecture et ses poids optimisés. L'image prétraitée est alors passée au modèle, qui utilise ses couches convolutées pour extraire les caractéristiques visuelles de l'image et produire un résultat, sous forme d'une classe prédite. Enfin, cette prédiction est interprétée pour déterminer l'enjeu ou la classe associée à l'image, permettant ainsi de prendre la décision et affiche le résultat.



Figure 4.15 : Sélection d'une Image Covid-19.

Prédiction : L'image sélectionnée est prétraitée et soumise au modèle pour prédiction.

Affichage du Résultat : Le résultat de la prédiction et la confiance associée sont affichés à l'utilisateur. Exemple : **Covide-19 et sa confidence est 98.55%.**

Le pourcentage de 98.55% représente la probabilité ou la confiance du modèle CNN que l'image importée montre des signes de Covid-19.

Une confiance de 98.55% signifie que, sur la base des caractéristiques extraites de l'image, le modèle estime qu'il y a environ 98 chances sur 100 que cette image corresponde à un cas de Covid-19. Cela indique une prédiction modérément confiante.

Boucle Principale : L'application continue à fonctionner, permettant à l'utilisateur d'interagir davantage ou de fermer la fenêtre.

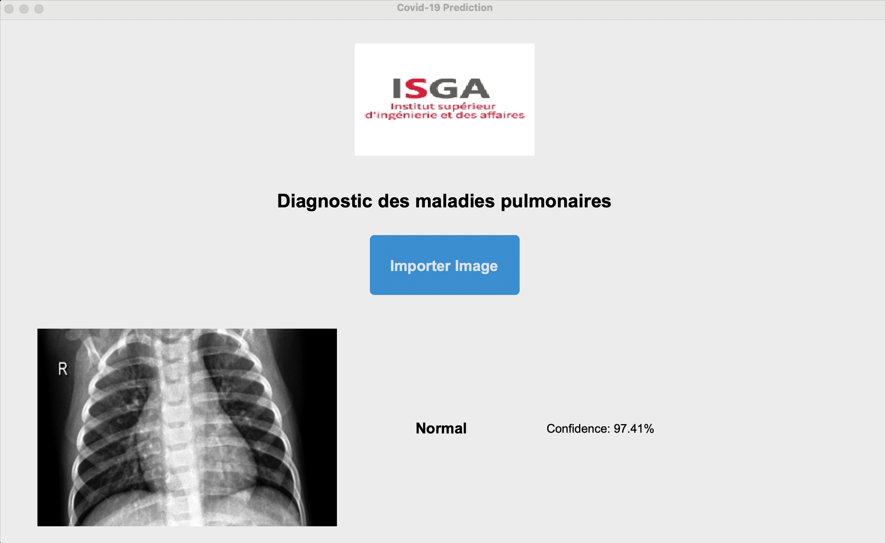


Figure 4.16 : Sélection d'une Image Normal.

Affichage du Résultat : Le résultat de la prédiction et la confiance associée sont affichés à l'utilisateur. Exemple : **Normal et sa confidence 97.41%.**

**97.41%** est un pourcentage élevé de confiance, ce qui signifie que le modèle est très certain que l'image importée montre des poumons sans anomalie détectable, donc normaux.

Boucle Principale : L'application continue à fonctionner, permettant à l'utilisateur d'interagir davantage ou de fermer la fenêtre.

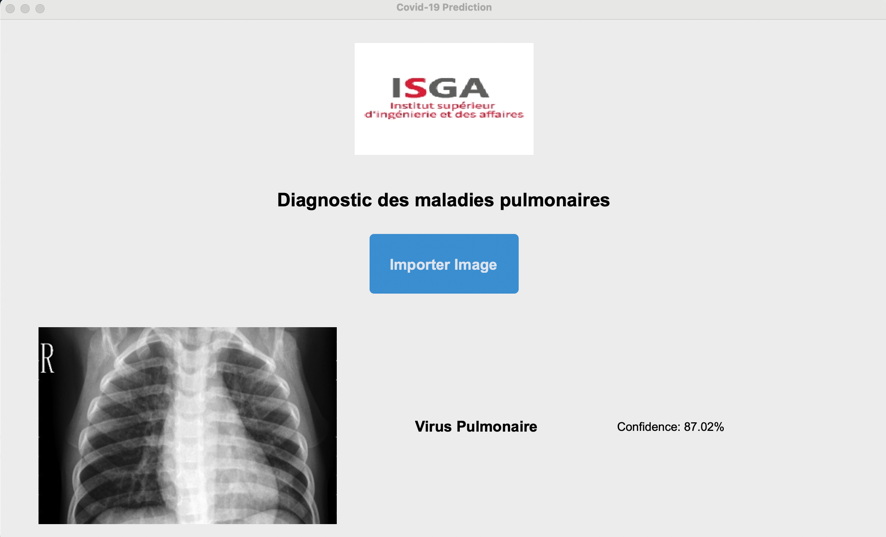


Figure 4.17 : Sélection d'une Image Virus Pulmonaire.

Affichage du Résultat : Le résultat de la prédiction et la confiance associée sont affichés à l'utilisateur. Exemple : **Virus Pulmonaire et sa confidence 87.02%**

Boucle Principale : L'application continue à fonctionner, permettant à l'utilisateur d'interagir davantage ou de fermer la fenêtre.



Figure 4.18 : Sélection d'une Image Inconnu.

Affichage du Résultat : Le résultat de la prédiction et la confiance associée sont affichés à l'utilisateur. Exemple : **Virus Pulmonaire 97.30%**

#### Conclusion

Dans notre projet, nous avons développé une nouvelle méthode de traitement d'images médicales pour détecter les maladies pulmonaires à partir d'images radiographiques du thorax. Nous avons appliqué une égalisation d'histogramme pour améliorer la distribution d'intensité des images, puis utilisé une architecture CNN légère pour entraîner notre modèle à détecter le COVID-19 et d'autres maladies. L'évaluation du modèle a été réalisée avec la matrice de confusion et la courbe ROC, obtenant un taux de précision de 96,5%. Il serait préférable de créer notre propre base de données d'images radiographiques et d'utiliser des techniques supplémentaires comme la segmentation pour améliorer les performances du modèle.

## Conclusion Générale du Projet

Ce projet a exploré l'application des techniques avancées de traitement d'image et d'intelligence artificielle (IA) pour améliorer la détection des maladies pulmonaires. En utilisant des méthodes d'apprentissage automatique, nous avons cherché à développer des outils diagnostiques plus précis et rapides.

Nous avons d'abord établi une compréhension approfondie des maladies pulmonaires, soulignant l'importance de la détection précoce pour améliorer les résultats des traitements. Ensuite, nous avons examiné les techniques actuelles de traitement d'image médicale, telles que la radiologie et l'imagerie par résonance magnétique (IRM), et leur intégration avec l'IA pour automatiser et améliorer l'interprétation des résultats.

L'utilisation de l'apprentissage automatique, notamment les réseaux neuronaux convolutifs (CNN), s'est avérée particulièrement efficace pour le traitement des images médicales. Nous avons collecté et préparé un ensemble de données d'images pulmonaires, incluant des radiographies thoraciques annotées. La préparation des données a impliqué des techniques de prétraitement comme l'égalisation d'histogramme pour améliorer le contraste et la clarté des images.

Nous avons ensuite développé et entraîné un modèle CNN pour la classification des images pulmonaires. L'évaluation des performances du modèle sur un ensemble de test indépendant a montré une grande capacité à distinguer les images saines des images malades, démontrant ainsi l'efficacité de notre approche.

Le modèle a été intégré dans une application conviviale, permettant aux professionnels de santé de télécharger des radiographies et d'obtenir une analyse rapide. Cette application complète les diagnostics traditionnels en fournissant un deuxième avis rapide et basé sur des données.

Les résultats de ce projet montrent que notre modèle peut détecter les maladies pulmonaires avec une précision comparable à celle des experts humains, tout en réduisant le temps nécessaire pour analyser une image. Cette avancée est particulièrement bénéfique dans les situations d'urgence, où une détection rapide est essentielle.

#### Bibliographie

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Cet article explique les bases de l'apprentissage profond et ses applications, notamment dans la reconnaissance d'images et le traitement du langage naturel.

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097- 1105.

Cet article présente les réseaux de neurones convolutifs (CNN) et leur application à la classification d'images sur le vaste ensemble de données ImageNet.

1. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van der Laak, J. A. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

Un examen approfondi des applications de l'apprentissage profond dans l'analyse des images médicales, couvrant divers types de réseaux neuronaux et leurs utilisations spécifiques.

1. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

Cet article introduit les réseaux de convolution denses (DenseNet), une avancée significative dans la conception de réseaux de neurones profonds.

1. O’Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Une introduction aux concepts fondamentaux des CNN, expliquant comment ils sont utilisés pour le traitement des images.

Livres

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Un manuel complet couvrant tous les aspects de l'apprentissage profond, des bases théoriques aux applications pratiques.

1. Raschka, S., & Mirjalili, V. (2019). Python Machine Learning. Packt Publishing Ltd.

Ce livre fournit une introduction pratique à l'apprentissage automatique en Python, avec des exemples de code et des applications concrètes.

1. Chollet, F. (2017). Deep Learning with Python. Manning Publications.

Ce livre, écrit par le créateur de Keras, offre une approche pratique de l'apprentissage profond avec des exemples en Python.

Thèses et Rapports

1. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Cette thèse discute des améliorations de l'architecture Inception pour les tâches de vision par ordinateur, avec des implications pour la classification d'images et la détection d'objets.

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770- 778).

Un rapport important sur les réseaux de neurones résiduels (ResNet), qui ont amélioré les performances des modèles de vision par ordinateur en résolvant le problème du gradient de vanishing. Sources en ligne

1. TensorFlow Documentation. Retrieved from https://[www.tensorflow.org/](http://www.tensorflow.org/)

La documentation officielle de TensorFlow, une bibliothèque open-source pour le machine learning et l'apprentissage profond.

1. Keras Documentation. Retrieved from https://keras.io/

La documentation officielle de Keras, une API de réseau de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow.

1. COVID-QU Dataset. Retrieved from https://[www.kaggle.com/](http://www.kaggle.com/)

La base de données d'images radiographiques pulmonaires utilisée dans ce projet, disponible sur la plateforme Kaggle.

1. Python Official Documentation. Retrieved from https://docs.python.org/3/ La documentation officielle du langage de programmation Python.

#### Webographie

1. Radiopaedia.org. Chest X-ray. Consulté le 11 avril 2024, à partir de https://radiopaedia.org/articles/chest-x-ray
2. World Health Organization (WHO). Coronavirus disease (COVID-19) pandemic. Consulté le 11 avril 2024, à partir de https:/[/www.who.int/](http://www.who.int/emergencies/diseases/novel-coronavirus-2019)e[mergencies/diseases/novel-coronavirus-2019](http://www.who.int/emergencies/diseases/novel-coronavirus-2019)
3. Towards Data Science. A Comprehensive Introduction to Different Types of Convolutions in Deep Learning. Consulté le 11 avril 2024, à partir de https://towardsdatascience.com/a- comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-7127fdcc412c
4. Kaggle. Chest X-Ray Images (Pneumonia). Consulté le 12 avril 2024, à partir de https:/[/www.kaggle.com/paultimothymooney/chest](http://www.kaggle.com/paultimothymooney/chest-xray-pneumonia)-[xray-pneumonia](http://www.kaggle.com/paultimothymooney/chest-xray-pneumonia)
5. Google AI Blog. AI for Radiology: CheXpert - A Large Dataset for Chest Radiograph Interpretation. Consulté le 11 avril 2024, à partir de https://ai.googleblog.com/2019/01/ai-for- radiology-chexpert-large-dataset.html
6. The Lancet Digital Health. AI for COVID-19: How Is AI Helping in the Fight Against the Coronavirus?. Consulté le 12 avril 2024, à partir de https:/[/www.thelanc](http://www.thelancet.com/journals/landig/article/PIIS2589-7500(20)30192-2/fulltext)e[t.com/journals/landig/article/PIIS2589-7500(20)30192-2/fulltext](http://www.thelancet.com/journals/landig/article/PIIS2589-7500(20)30192-2/fulltext)
7. National Institutes of Health (NIH). Deep Learning for Chest Radiograph Diagnosis: CheXNet. Consulté le [date], à partir de https:/[/www.nih.gov/new](http://www.nih.gov/news-events/news-releases/deep-learning-)s[-events/news-releases/deep-learning-](http://www.nih.gov/news-events/news-releases/deep-learning-) chest-radiograph-diagnosis-chexnet
8. RSNA.org. RSNA Pneumonia Detection Challenge. Consulté le 13 avril 2024, à partir de https:/[/www.rsna.org/edu](http://www.rsna.org/education/ai-resources-and-training/ai-image-challenge/rsna-pneumonia-)c[ation/ai-resources-and-training/ai-image-challenge/rsna-pneumonia-](http://www.rsna.org/education/ai-resources-and-training/ai-image-challenge/rsna-pneumonia-) detection-challenge
9. MIT Technology Review. AI in Medicine: Predicting Lung Diseases from X-rays. Consulté le

12 avril 2024, à partir de https://[www.technologyreview.com/2019/05/23/134468/ai-in-](http://www.technologyreview.com/2019/05/23/134468/ai-in-) medicine-predicting-lung-diseases-from-x-rays/

1. Mayo Clinic. COVID-19: AI-Driven X-ray Analysis. Consulté le 12 avril 2024, à partir de https:/[/www.mayoclinic.org/covid](http://www.mayoclinic.org/covid-19/ai-driven-x-ray-analysis)-[19/ai-driven-x-ray-analysis](http://www.mayoclinic.org/covid-19/ai-driven-x-ray-analysis)