

DIPLÔME DE DATASCIENTIST

---

# RADIOGRAPHIES

# COVID-19

---

Soutenu le 08/12/2025, par :

**Asma KERKACHE**

**Maamar BENHENNI**

**Richard TONNANG**

Devant le jury composé de :

# REMERCIEMENTS

A l'issu de cette formation au sein de l'organisme DataScientest, nous tenons à remercier l'ensemble des acteurs qui ont contribué au déroulé de ces trois mois.

Nous remercions tout d'abord les fondateurs de la plateforme DataScientest, Yoel Tordjman, Benjamin Brami et Charles Sutton.

Nous remercions également les équipes pédagogiques pour la qualité de l'enseignement dispensé tout au long de la formation.

Les équipes management, notamment Cameron, pour la gestion et la transmission des informations relatives aux différentes étapes de la formation. Ainsi qu'à l'ensemble des équipes techniques assurant le bon fonctionnement de la plateforme.

Nous adressons ensuite nos remerciements à notre chef de cohorte, Vincent Lalanne, pour sa disponibilité et ses conseils tout au long du parcours.

Nous tenons également à remercier notre mentor de projet, Antoine Fradin, dont l'encadrement et les retours pertinents ont grandement facilité l'avancement et l'aboutissement du projet.

Enfin, la dynamique de notre groupe a permis à chacun d'apporter sa part de réflexion et de stratégie. L'entente a été naturelle et l'avancement du projet, fluide.

# AVANT PROPOS

Le travail présenté dans ce rapport a été réalisé dans le cadre d'une formation de data science au sein de l'organisme DataScientest en collaboration avec l'École des MINES Paris. Elle s'est déroulée entre septembre 2025 et décembre 2025 sur la plateforme DataScientest Learn sous la direction du chef de cohorte Vincent Lalanne et du mentor de projet Antoine Fradin.

Le sujet du projet concerne l'élaboration d'un modèle permettant de diagnostiquer le Covid-19 à partir de radiographies thoraciques.

Le rapport est composé de quatre grandes parties, la première présente l'exploration des données, les trois suivantes traitent des différentes études effectuées. La numérotation des figures, des tableaux et des références bibliographiques est continue sur l'ensemble du document. Enfin les annexes sont regroupées en fin de rapport.

# TABLE DES MATIÈRES

INTRODUCTION .....	5
EXPLORATION DES DONNÉES .....	10
1. Répartition et structure du jeu de données .....	10
2. Taille et format des fichiers :.....	12
Conclusion : .....	13
PARTIE 1 : APPROCHE MACHINE LEARNING .....	14
1. Pré-processing.....	14
1.1. Répartition des données .....	14
1.2. Taille des images .....	14
1.3. Traitement des images .....	16
2. Sélection du modèle .....	17
2.1. Analyse préliminaire des modèles .....	18
2.2. Filtrage des modèles.....	18
3. Modélisation.....	20
3.1. Entrainement.....	20
3.2. Evaluation et interprétations.....	22
PARTIE 2 : DEEP LEARNING – DENSENET121.....	24
1. Pré-processing.....	24
2. Modélisation.....	25
2.1. Entraînement.....	25
2.2. Evaluation et interprétations.....	26
PARTIE 3 : DEEP LEARNING – INCEPTIONV3 .....	30
1. Pré-processing.....	30
2. Modélisation.....	32
2.1. Entraînement.....	32
2.2. Evaluation et interprétations.....	34
CONCLUSION .....	38
ANNEXES .....	40
BIBLIOGRAPHIE .....	44

# INTRODUCTION

Au cours de la pandémie mondiale, la recherche médicale a dû s'adapter rapidement aux exigences d'un virus imprévisible. Face à cette crise, des approches innovantes ont émergé, parmi lesquelles l'intelligence artificielle (IA). En particulier, les réseaux neuronaux convolutifs (*CNN*), capables d'analyser des images radiographiques avec une précision impressionnante, proposant une nouvelle voie pour un diagnostic rapide et fiable du COVID-19.

## *Présentation et intérêt du sujet*

Le projet de classification multi-classes des radiographies pulmonaires pour la détection du COVID-19 s'inscrit dans un contexte où l'intelligence artificielle (IA) propose des solutions innovantes face aux défis posés par les crises sanitaires mondiales. Durant la pandémie de COVID-19, la détection rapide de la maladie est devenue cruciale, et l'utilisation des radiographies pulmonaires s'est révélée être un outil précieux dans ce processus. Cependant, l'analyse des radiographies nécessite une expertise spécifique et peut être chronophage pour les radiologues, en particulier en période de forte demande.

Ce projet a pour objectif de développer un modèle automatisé capable de classer les radiographies pulmonaires en quatre catégories principales : COVID-19, pneumonie virale, opacité pulmonaire, et radiographies normales. Le système proposé combine plusieurs approches d'apprentissage automatique et d'apprentissage profond afin d'assurer une classification précise et rapide, contribuant ainsi à un diagnostic plus efficace et à une prise en charge rapide des patients.

Pour ce faire, le projet repose sur l'utilisation de réseaux neuronaux convolutifs (*CNN*), qui sont particulièrement adaptés à l'analyse d'images complexes. Cependant, il intègre également des techniques de machine learning traditionnelles telles que les machines à vecteurs de support (*SVM*), la régression logistique, les k-plus-proches voisins (*KNN*), les réseaux de neurones simples et *XGBoost*. L'intégration de ces méthodes permet de comparer les performances et d'identifier la technique la plus adaptée pour chaque cas spécifique.

L'intérêt de cette approche hybride réside dans sa capacité à offrir une solution flexible et robuste pour la détection du COVID-19 et d'autres pathologies pulmonaires. En combinant des modèles complexes comme les *CNN* avec des algorithmes plus simples mais efficaces comme les *SVM* ou *XGBoost*, le projet vise à créer un système capable de fonctionner rapidement, même dans des environnements à ressources limitées.

L'une des grandes forces de ce projet est sa scalabilité : une fois le modèle développé, il pourra être appliqué à d'autres contextes médicaux, telle que la détection de diverses pathologies pulmonaires. De plus, l'automatisation de l'analyse des radiographies pourrait alléger la charge

de travail des radiologues, leur permettant de se concentrer sur les cas complexes, tout en assurant une détection rapide et fiable des cas suspects.

Au-delà de son application immédiate pour la COVID-19, ce projet a un potentiel considérable pour améliorer la qualité des soins dans des environnements où l'accès à des radiologues spécialisés est limité, tels que dans les zones rurales ou dans des pays à faibles ressources. L'utilisation de l'IA pour l'analyse des radiographies pourrait ainsi rendre le diagnostic plus accessible, précis et rapide à l'échelle mondiale.

### *Cadre théorique*

L'analyse d'images médicales est un domaine en pleine expansion, soutenu par des progrès significatifs dans les technologies de l'intelligence artificielle (IA) et du machine learning. Ces techniques permettent de traiter des images complexes, comme les radiographies pulmonaires, pour détecter des anomalies et aider au diagnostic de diverses pathologies. L'une des approches les plus populaires dans ce domaine est l'apprentissage profond, qui a montré d'excellentes performances dans des tâches telles que la détection de tumeurs, la segmentation d'organes, ou encore la classification de maladies pulmonaires.

### *Classification d'images médicales : Concepts et Méthodes*

La classification d'images médicales consiste à attribuer une étiquette (ou une catégorie) à une image en fonction des caractéristiques visuelles qu'elle présente. Les méthodes de classification peuvent être supervisées ou non supervisées, mais pour les images médicales, l'approche supervisée est la plus courante, car elle repose sur des ensembles de données annotés par des experts (radiologues, médecins).

Parmi les méthodes les plus utilisées, on retrouve :

- Les réseaux neuronaux convolutifs (*CNN*), qui sont des modèles d'apprentissage profond spécialement conçus pour traiter des données visuelles. Ces réseaux ont l'avantage d'apprendre directement les caractéristiques importantes des images sans nécessiter une extraction manuelle de ces caractéristiques.
- Les machines à vecteurs de support (*SVM*), qui sont largement utilisées pour des tâches de classification binaire ou multi-classes. Elles sont efficaces, même avec des jeux de données de petite taille peuvent bien fonctionner avec des données linéairement séparables.
- Les k-plus-proches voisins (*KNN*), une méthode simple mais puissante qui classe une image en fonction des classes des exemples les plus proches dans l'espace des caractéristiques.
- *XGBoost*, au lieu de se baser sur un seul arbre de décision, une technique d'ensemble combine les prédictions de nombreux arbres différents. Cette approche vise à renforcer la robustesse et à améliorer la performance globale de la classification en exploitant la

sagesse collective de ces arbres, ce qui conduit généralement à des résultats plus fiables et précis.

- Les réseaux de neurones simples, qui sont une version plus basique des *CNN* et peuvent être utilisés pour des tâches de classification moins complexes.

Ces techniques ont été adaptées avec succès dans de nombreux domaines médicaux, y compris la radiologie pulmonaire, où l'analyse des radiographies thoraciques permet de détecter des maladies telles que le cancer du poumon, la pneumonie et la tuberculose.

### *Applications dans la détection du COVID-19*

Depuis l'émergence de la pandémie de COVID-19, de nombreuses études ont exploré l'utilisation des images radiographiques thoraciques pour identifier les signes du virus. L'analyse d'images, en particulier les radiographies et les tomodensitogrammes (CT scans), est devenue une méthode complémentaire aux tests PCR pour détecter la COVID-19. Plusieurs études ont démontré que les images radiologiques des poumons des patients COVID-19 montrent des caractéristiques distinctes telles que des opacités en verre dépoli, des consolidations ou des infiltrats pulmonaires bilatéraux, qui peuvent être utilisées pour classer les patients.

Quelques travaux notables incluent :

- Chouhan *et al.*<sup>[1]</sup>, qui ont utilisé un modèle *CNN* pour classer les radiographies thoraciques en trois catégories : COVID-19, pneumonie et images normales. Leur modèle a montré une grande précision, démontrant l'efficacité des *CNN* pour la détection du COVID-19.
- Wang *et al.*<sup>[2]</sup> ont appliqué une architecture de *CNN* à un ensemble de données d'images CT pour détecter le COVID-19, atteignant des performances remarquables en termes de précision et de rappel. Leur travail a été parmi les premiers à prouver que l'IA pouvait être utilisée de manière fiable dans la détection du COVID-19.
- Rahimzadeh *et al.*<sup>[3]</sup> ont développé un système de diagnostic basé sur un modèle hybride combinant *CNN* et *SVM* pour identifier les signes de COVID-19 à partir de radiographies. Ce type de modèle hybride a montré des résultats très prometteurs dans l'amélioration des taux de détection tout en réduisant le taux de faux positifs.

### *Limites des études existantes et opportunités pour ce projet*

Bien que les études sur l'utilisation de l'IA pour la détection du COVID-19 soient prometteuses, plusieurs défis restent à relever :

- **Variabilité des données** : Les images médicales sont souvent de qualité variable, en fonction des équipements utilisés, des protocoles de prise en charge, et des

caractéristiques des patients. Cela peut rendre difficile l'entraînement de modèles robustes capables de généraliser à de nouvelles données.

- **Biais de données** : De nombreux modèles sont formés sur des ensembles de données limités qui ne couvrent pas toutes les variations possibles de la maladie. Cela peut réduire la capacité du modèle à identifier correctement des cas atypiques ou des variations de la pathologie.
- **Complexité des modèles** : Bien que des modèles complexes comme les CNN montrent de très bons résultats, leur déploiement dans des environnements cliniques peut être limité par des contraintes de temps et de ressources computationnelles, notamment dans les hôpitaux avec des infrastructures limitées.

### *Problématique*

La pandémie de COVID-19, qui a débuté fin 2019, a exacerbé la pression sur les systèmes de santé mondiaux, révélant l'urgence de développer des méthodes de détection rapides et efficaces. Le diagnostic précoce est crucial pour limiter la propagation du virus, optimiser les traitements et éviter les surcharges hospitalières. Toutefois, malgré les avancées dans les tests diagnostiques, plusieurs défis demeurent dans la mise en œuvre de méthodes de détection efficaces, notamment dans les pays à faibles ressources et dans les environnements à forte demande.

### *Les défis du diagnostic du COVID-19*

Le test PCR, bien que considéré comme la norme pour la détection du COVID-19, reste long et coûteux, avec des résultats parfois obtenus plusieurs heures, voire plusieurs jours, après le prélèvement. De plus, ce test n'est pas disponible dans toutes les régions, notamment dans les zones rurales et les pays en développement, où l'accès aux infrastructures médicales est limité. Par ailleurs, les tests rapides antigéniques, bien que plus accessibles, ont montré une précision inégale, particulièrement dans les stades précoce de l'infection.

Dans ce contexte, l'utilisation des radiographies pulmonaires est apparue comme une méthode complémentaire intéressante pour la détection rapide du COVID-19. Les radiographies permettent une première évaluation visuelle de l'état des poumons et permettent d'identifier des signes caractéristiques de la maladie, comme les opacités en verre dépoli, les infiltrats pulmonaires bilatéraux, et d'autres lésions spécifiques. Cependant, l'interprétation de ces images nécessite une expertise spécialisée et un temps d'examen considérable, ce qui peut être un obstacle majeur en période de crise sanitaire mondiale.

## *Plan du projet*

Ce rapport est structuré en quatre grandes parties, organisées de manière progressive afin d'aboutir à une analyse complète du problème étudié.

La première partie est consacrée à l'exploration des données, étape commune et indispensable à toutes les stratégies développées par la suite. Elle vise à comprendre la structure du jeu de données, identifier ses particularités, détecter d'éventuels déséquilibres entre classes et mettre en place les prétraitements nécessaires avant toute modélisation.

La deuxième partie présente une première stratégie reposant sur une approche de machine learning classique, appliquée sur des caractéristiques extraites manuellement. Cette section détaille la construction des *features*, la sélection du modèle, ainsi que l'évaluation de ses performances.

Les troisième et quatrième parties s'orientent vers des approches de deep learning, mobilisant deux modèles de réseaux neuronaux distincts. Ces parties décrivent l'architecture des modèles, les étapes d'entraînement, les optimisations mises en place et les performances obtenues.

L'ensemble de ces approches permet ainsi une comparaison complète, bien que non exhaustive, des différentes méthodes de classification dans ce contexte.

Enfin une conclusion générale viendra clore ce rapport en résumant les différentes études abordées et les résultats obtenus.

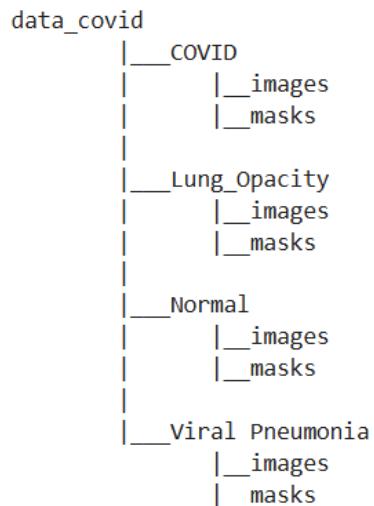
# EXPLORATION DES DONNÉES

## 1. Répartition et structure du jeu de données

Dans le cadre de ce projet nous avons exploité plusieurs ensembles de données publics de radiographies thoraciques disponibles et rassemblés sur le site de Kaggle.<sup>[4]</sup> Le jeu de données regroupe des images et leur masques correspondants classées en quatre catégories :

- Covid-19
- Pneumonie virale
- Opacité pulmonaire
- Normal

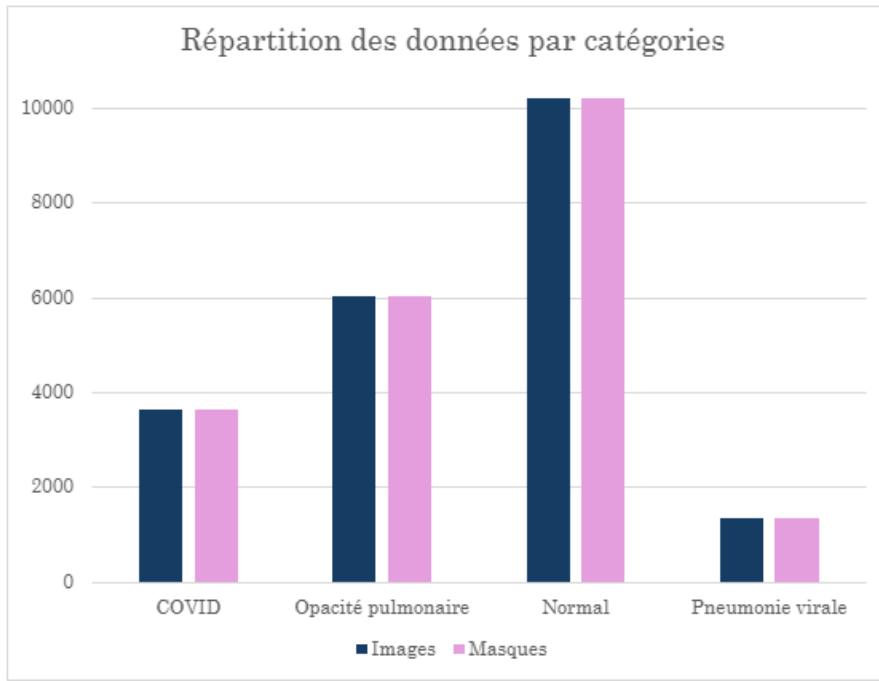
Les images sont archivées dans des dossiers correspondants à leur catégorie, comme illustré dans la *Figure 1*, qui présente l'organisation du jeu de données.



***Figure 1 : Structure du jeu de données***

L'intégrité des données a été vérifiée, aucun élément corrompu ni nom de fichier invalide n'a été détecté. Cette étape permet de s'assurer que toutes les images et leurs masques associés sont exploitables pour la suite de l'étude.

Nous nous sommes ensuite intéressés à la répartition du nombre d'images et de masques au sein de chaque catégorie. Cette étape permet d'identifier un éventuel déséquilibre des classes, qui pourrait influencer les performances du modèle. Les résultats de cette analyse sont présentés dans la *Figure 2* :



**Figure 2 :** Histogramme sur la répartition des données par catégorie

Nous constatons que les données sont réparties de manière inégale entre les différentes catégories. En effet, la catégorie Normal compte le plus d'images et de masques, et représente environ 48% des données. Ceci peut s'expliquer par la forte disponibilité de radiographies thoraciques saines dans les dépôts publics.

Elle est suivie par les catégories Opacité pulmonaire, COVID et Pneumonie virale, qui représentent respectivement 28%, 17% et 6% des données.

Néanmoins, le nombre d'images est identique au nombre de masques dans chaque catégorie, ce qui indique qu'aucun masque n'est manquant.

Il est toutefois nécessaire de traiter cette disparité car un modèle entraîné sur un jeu de données aussi déséquilibré risque de favoriser la classe majoritaire (ici Normal) au détriment des classes minoritaires notamment la classe Covid-19 qui nous intéresse.

Pour éviter ce souci, différentes approches sont possibles :

- Sous-échantillonnage de la classe majoritaire
- Sur-échantillonnage des classes minoritaires
- Pondération des classes dans la fonction de perte
- Suppression de certaines données

Dans ce projet, nous avons appliqué deux traitements différents, que nous détaillerons plus en avant dans ce manuscrit, afin de répondre à cette problématique.

## 2. Taille et format des fichiers :

Un autre aspect important du jeu de données concerne la taille et le format des images et des masques. L'exploration préliminaire a permis d'identifier l'extension des fichiers ainsi que les dimensions associées à chaque catégorie.

Les informations sont regroupées dans le *Tableau 1* ci-dessous :

Tableau 1: Formats et tailles des images et des masques

Catégories	Format		Taille	
	Images	Masques	Images	Masques
COVID	PNG	PNG	299 x 299	256 x 256
Normal	PNG	PNG	299 x 299	256 x 256
Opacité pulmonaire	PNG	PNG	299 x 299	256 x 256
Pneumonie Virale	PNG	PNG	299 x 299	256 x 256

Nous observons que les formats des images et des masques sont identiques (PNG), ce qui facilite leur manipulation et les étapes ultérieures de traitement.

Cependant, les dimensions diffèrent : les images présentent une résolution de 299x299 pixels, tandis que les masques sont fournis en 256x256 pixels.

Cette disparité nécessite une uniformisation des tailles afin de pouvoir superposer correctement chaque masque à son image correspondante lors de l'étape de pré-traitement.

Afin d'illustrer visuellement les différentes catégories du jeu de données, la *Figure 3* présente un échantillon d'images représentatives pour chacune d'elles.

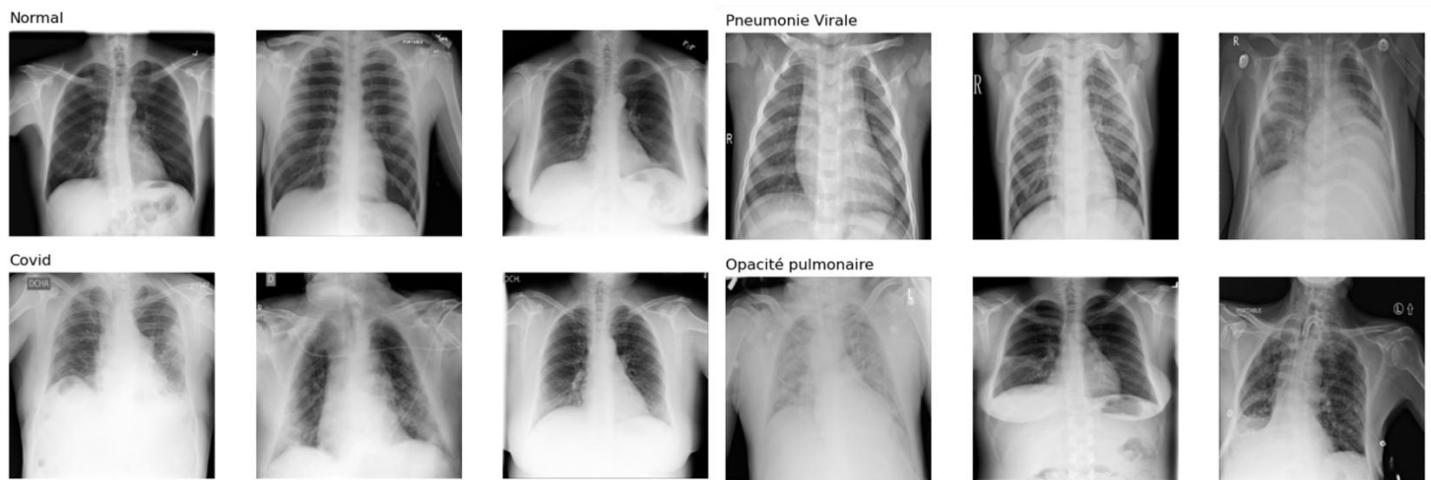


Figure 3 : Echantillon d'images aléatoire par catégorie

## **Conclusion :**

L'exploration du jeu de données a permis d'avoir une vision claire de sa structure, de sa qualité et des contraintes associées à son exploitation. Les images sont organisées de manière cohérente dans des dossiers correspondant aux différentes catégories cliniques. L'intégrité des fichiers a été vérifiée, aucun fichier corrompu n'a été détecté.

L'analyse a également mis en évidence la présence de métadonnées associées aux fichiers, mais celles-ci ne contiennent aucune information clinique sur les patients (âge, sexe, antécédents médicaux, etc). Elles se limitent principalement à des informations techniques, telles que le format, taille ou encore le chemin d'accès des images.

Sur le plan quantitatif, nous avons une grande disparité entre les classes, notamment la classe Normal, majoritaire.

Sur le plan technique, les images et masques sont fournis au format PNG, cependant, leurs dimensions ne sont pas homogènes : les images sont en 299x299 pixels et les masques en 256x256 pixels. Une étape d'uniformisation est nécessaire.

Ces constatations nous ont permis d'élaborer trois approches, chacune repose sur un pré-processing ainsi qu'une stratégie de modélisation propre. Les sections suivantes détaillent ces trois approches, la méthodologie ainsi que les performances obtenues.

# PARTIE 1 : APPROCHE MACHINE LEARNING

La première approche repose sur des méthodes de machine learning. Dans un premier temps, les images ont été converties en matrices numériques à l'aide de la bibliothèque *Numpy*.<sup>[5]</sup> Après différents traitements, qui seront détaillés dans cette section, l'ensemble de données a été organisé dans un tableau *DataFrame* via la bibliothèque *Pandas*<sup>[6]</sup>, afin de faciliter les manipulations et les analyses ultérieures.

## 1. Pré-processing

### 1.1. Répartition des données

Pour cette approche, nous avons retenu trois catégories principales, correspondant aux classes finales du modèle :

- 0 → Normal
- 1 → Covid-19
- 2 → Autres pathologies (Opacité pulmonaire et pneumonie virale)

Ce regroupement a été effectué afin de réduire le déséquilibre entre les différentes catégories initiales du jeu de données. En effet, celles-ci présentaient des tailles très hétérogènes, ce qui aurait pénalisé l'apprentissage du modèle en introduisant un biais en faveur des classes majoritaires.

Afin d'assurer un apprentissage équilibré, nous avons uniformisé le nombre d'images par catégories en sélectionnant 3610 radiographies pour Normal et Covid-19, 2268 pour Opacité pulmonaire et 1342 pour Pneumonie virale. La classe « Autres pathologies » regroupe un mélange de radiographies issues des catégories pneumonie virale et opacité pulmonaire.

Cette étape de rééquilibrage est essentielle en machine learning supervisé, car elle permet :

- D'éviter qu'un modèle ne privilégie systématiquement la classe la plus représentée,
- D'améliorer la généralisation,
- De rendre les métriques (précision, rappel, F1-score) plus comparables entre les catégories.

### 1.2. Taille des images

Afin d'identifier la taille d'image la plus adaptée à l'approche machine learning, plusieurs dimensions ont été testées aussi bien pour les images que pour leurs masques associés.

Dans un premier temps, chaque radiographie a été redimensionnée à différentes résolutions. Le masque correspondant a ensuite été ajusté à la même taille afin de garantir une superposition correcte lors de l'application du masque sur l'image.

Après la mise à l'échelle, les images ont été aplatis afin d'être représentées sous forme de vecteurs(1D) numériques compatibles avec la plupart des algorithmes de machine learning traditionnels.

Cette étape est indispensable, car contrairement aux méthodes de deep learning, ces algorithmes ne peuvent pas exploiter directement une structure matricielle ou une représentation 2D.

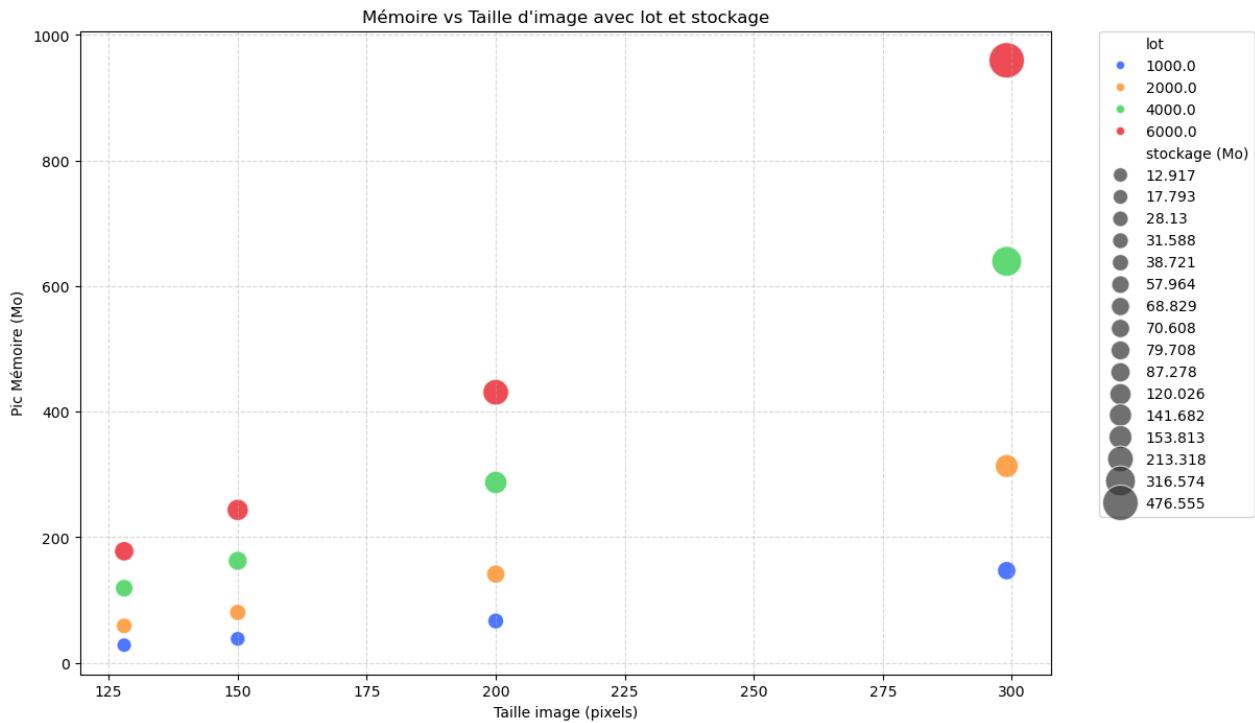
Nous avons ensuite évalué l'impact de ces différentes tailles sur :

- L'utilisation mémoire (RAM),
- Le temps d'exécution,
- Le stockage disque,
- Ainsi que la dimension finale du vecteur d'entrée (nombre de pixels).

Le *Tableau 2* présente les résultats obtenus pour chaque taille testée, mettant en évidence le compromis nécessaire entre précision potentielle, charge mémoire et contraintes computationnelles.

*Tableau 2 : Impact de la taille et du nombre d'images sur la mémoire et le temps de calcul*

Nombre d'images	Taille images	Pic mémoire (Mo)	Temps d'exécution (s)	Colonnes pixels	Stockage (Mo)
1000	128	28.102	1.405	13544	12.917
	150	38.122	1.326	18657	17.793
	200	66.488	1.369	33122	31.588
	299	146.719	1.508	74038	70.608
2000	128	58.834	2.515	14748	28.130
	150	80.161	2.712	20301	38.721
	200	141.192	2.759	36086	68.829
	299	313.462	3.000	80642	153.813
4000	128	118.782	5.100	15195	57.964
	150	162.418	5.103	20895	79.708
	200	287.195	5.441	37141	141.682
	299	639.463	5.922	82988	316.574
6000	128	177.676	7.617	15253	87.278
	150	243.462	7.756	20976	120.026
	200	430.878	8.087	37280	213.318
	299	959.698	8.780	83284	476.555



**Figure 4 :** Mémoire en fonction de la taille des images et des lots

Comme attendu, les résultats (illustrés en *Figure 4*) montrent que l'augmentation de la tailles des images et du nombre d'images entraîne une hausse significative de la mémoire nécessaire, du temps d'exécution et de l'espace de stockage.

Les formats les plus grands (notamment 299x299) deviennent rapidement coûteux, rendant nécessaire un compromis entre qualité visuelle et contraintes computationnelles.

### 1.3. Traitement des images

Après avoir stocké les vecteurs d'images dans un DataFrame X et les labels correspondants dans un DataFrame y, plusieurs étapes de prétraitement ont été réalisées afin de préparer les données pour les différents modèles de machine learning.

Ces étapes ont pour objectif d'améliorer la qualité des images, d'extraire des caractéristiques pertinentes et d'assurer une séparation fiable entre entraînement et test.

Les principales étapes de prétraitement sont les suivantes :

#### 1. Définition des paramètres *HOG* et *CLAHE* :

Les paramètres des descripteurs *HOG* (nombre de cellules, pixels par cellule, orientations...) et du filtre *CLAHE* (clipLimit, tileSize) ont été adaptés pour chaque taille d'image testée, afin de garantir une extraction cohérente des informations.

#### 2. Calcul dynamique de la longueur du vecteur *HOG* :

Selon la taille de l'image, la dimension finale du vecteur *HOG* change. Cette étape permet d'ajuster automatiquement la longueur du vecteur afin qu'il puisse ensuite être utilisé dans un modèle de machine learning classique.

#### 3. Application du filtre *CLAHE* :

*CLAHE* est appliqué à chaque image pour améliorer le contraste local, ce qui permet une meilleure mise en évidence des détails anatomiques et potentiellement des anomalies pulmonaire.

#### 4. Extraction des caractéristiques *HOG* :

Les descripteurs *HOG* sont calculés pour chaque image. Ils permettent de capturer les gradients et les contours, utiles pour différencier les structures pulmonaires selon la pathologie.

#### 5. Standardisation des vecteurs de caractéristiques :

Les vecteurs *HOG* obtenus sont normalisés à l'aide de *StandardScaler*, afin d'avoir des variables centrées et réduites, ce qui améliore la convergence et la stabilité des modèles d'apprentissage.

#### 6. Séparation train/test avec stratification :

- La fonction *train\_test\_split(..., stratify=y\_array)* est utilisée pour garantir une répartition identique des classes dans les ensembles d'entraînement et de test.
- Cette stratification permet de prévenir les déséquilibres dans les sous-échantillons et assure une évaluation plus fiable des performances.

#### 7. Sauvegarde des splits :

Les différents jeux de données (train/test) obtenus pour chaque taille d'images sont enregistrés dans des fichiers *.pkl*.

L'ensemble de ces traitements permet ainsi d'obtenir des données préparées, normalisées et structurées, prêtes à être utilisées pour l'entraînement et l'évaluation des différents modèles de machine learning.

## 2. Sélection du modèle

Afin d'identifier le modèle de machine learning le plus adapté à notre problématique, une phase de modélisation préliminaire a été menée. Celle-ci a consisté à tester plusieurs algorithmes sur différentes tailles d'image et sur plusieurs volumes d'échantillon.

L'objectif était d'évaluer l'impact combiné de la résolution des images, de la quantité de données d'entraînement et du choix du modèle sur les performances finales.

Le protocole suivi est décrit ci-dessous :

#### 1. Définition des tailles et échantillons

- Tailles des images : 64, 128, 176, 224, 299 pixels
- Nombre d'images pour l'entraînement : 1664, 2664, 4664, 6664, 8664

#### 2. Sélection des modèles

- Classique : *SVC*, *KNN*, *MLP*, *LogisticRegression*
- Arbre : *DecisionTree*, *RandomForest*, *ExtraTrees*, *GradientBoosting*, *XGBoost*

#### 3. Boucle d'expérimentation

- Pour chaque type de modèle (Classique ou Arbre)
- Pour chaque taille d'échantillon
- Pour chaque taille d'image

#### 4. Préparation des données

- Chargement des splits train/test préalablement générés
- Evaluation sur toutes les tailles d'échantillons pour observer l'impact de la quantité de données sur les scores

## 5. Entraînement et évaluation

- Entraînement de chaque modèle sur le sous-échantillon correspondant
- Calcul du score sur l'ensemble d'entraînement (score\_train)
- Calcul du score sur l'ensemble de test (score\_test)
- Mesure du temps d'entraînement pour chaque modèle

## 6. Stockage des résultats

- Tous les résultats sont ajoutés à une liste *resultat*
- Sauvegarde intermédiaire en CSV après chaque taille d'échantillon pour éviter toute perte en cas d'interruption.

### 2.1. Analyse préliminaire des modèles

Une première évaluation générale a été réalisée sur l'ensemble des modèles présentés précédemment. Le *Tableau 3* ci-dessous résume les performances moyennes pour chaque classifieur.

*Tableau 3 : Evaluation moyenne préliminaire*

ClassifiEURS	score_train	score_test	Temps(s)
<b>DecisionTree</b>	1.000000	0.514604	22.1452
<b>ExtraTrees</b>	1.000000	0.695844	1.9712
<b>GradientBoosting</b>	0.915268	0.710836	1640.7248
<b>KNN</b>	0.747336	0.623692	0.0144
<b>LR</b>	0.993328	0.667204	3.8416
<b>MLP</b>	1.000000	0.743372	15.0468
<b>RandomForest</b>	1.000000	0.691880	3.6944
<b>SVC</b>	0.918276	0.735008	334.7444
<b>XGB</b>	1.000000	0.737008	90.8952

Les résultats montrent des comportements très variés selon les modèles. Certains obtiennent un score d'entraînement parfait (score\_train = 1.0), révélant un fort risque de surapprentissage, tandis que leurs scores sur l'ensemble de test restent nettement inférieurs.

### 2.2. Filtrage des modèles

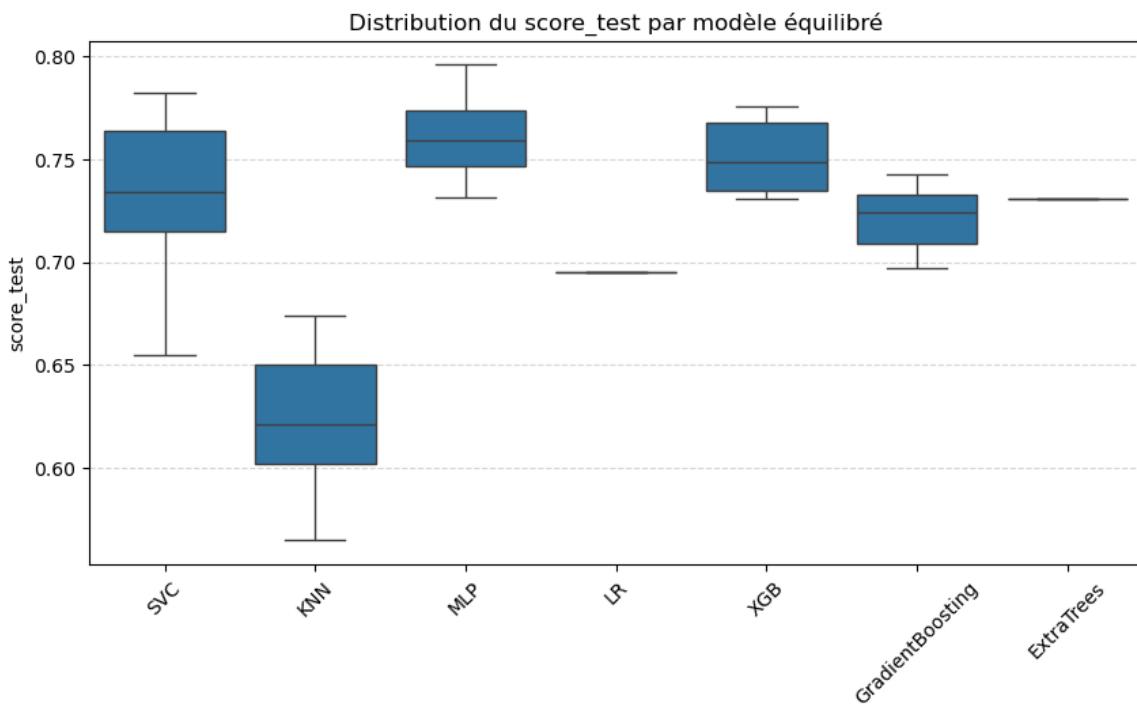
Cette évaluation reste néanmoins générale, elle ne prend pas en compte la taille des images ou le volume des données.

Afin d'identifier les modèles réellement prometteurs, un premier filtre est appliqué, basé sur les critères suivants :

- Score test  $\geq 0.7$ , performances acceptables
- Ecart ( $\text{score\_train} - \text{score\_test}$ )  $\leq 0.27$ , peu ou pas de surapprentissage.

Ces conditions permettent de conserver uniquement les modèles à la fois performants et stables. A l'issu de cette étape, le nombre de modèles retenus passe de 9 à 7.

La *Figure 5* présente la distribution des score\_test obtenus pour les modèles jugés équilibrés. Ce graphique permet d'identifier certains modèles prometteurs pour notre projet.



**Figure 5 :** Distribution des score test par modèles équilibrés

Un second filtre est appliqué afin d'affiner la sélection des modèles. La stratégie adoptée est la suivante :

- Calcul de la moyenne du score\_train et du score\_test pour chaque combinaison (taille/classifieur).
- Pour chaque modèle, calcul de l'écart avec le score\_train moyen et le score\_test moyen
- Les modèles conservés sont ceux présentant un écart  $\leq 0.02$ .

Après cette seconde étape de filtrage, 5 modèles sont retenus. Leur comportement en fonction du volume de données est présenté dans l'*Annexe 1*, qui regroupe l'ensemble des courbes montrant l'évolution des scores en fonction du nombre d'images.

L'étude des graphes met en évidence plusieurs tendances importantes :

- XGB présente les performances les plus constantes, avec des scores homogènes et un écart minimal entre train et test, et ce quelle que soit la taille de l'échantillon.
- MLP affiche également une bonne stabilité, avec une faible variation entre les tailles d'image.

Ces deux modèles ressortent ainsi comme les candidats les plus robustes.

En revanche, les autres modèles présentes des limitations :

- ExtraTrees et GradientBoosting : performances insuffisantes pour certaines tailles d'échantillons, dues à un manque de données représentatives.
- SVC : scores très instables d'une taille d'image à l'autre, rendant son comportement difficilement prévisible.

Au regard de l'ensemble de ces observations, nous retenons finalement le modèle XGB<sup>[7]</sup> avec une taille d'image de 128x128 pixels. Il se distingue par sa stabilité face aux variations de taille et de volume d'images et par l'écart minimal entre le score d'entraînement et le score de test.

Il constitue donc le modèle le plus fiable pour la suite du projet.

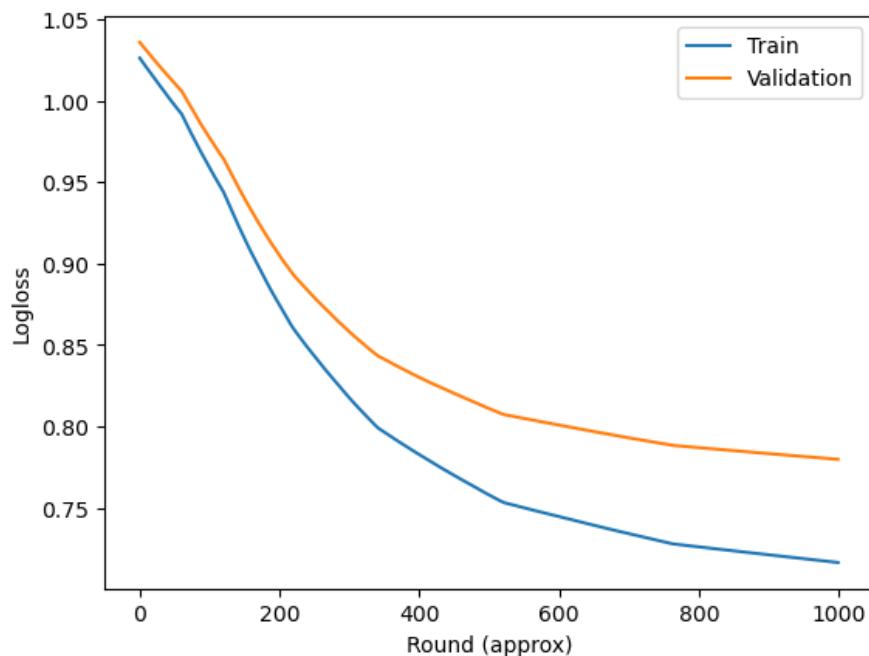
### 3. Modélisation

#### 3.1. Entrainement

Le modèle XGB sélectionné précédemment est alors optimisé et entraîné en ajustant différents paramètres, notamment :

- La valeur du *max\_depth*
- L'amplification des poids de la classe Covid-19
- La recherche du seuil F1 optimal pour la classe Covid-19

Avant d'affiner ces différents paramètres, un premier entraînement sur 1000 itérations a été réalisé afin d'observer le comportement initial du modèle. La *Figure 6* illustre ainsi l'évolution de la logloss au fil des itérations pour les ensembles d'entraînement et de validation :



*Figure 6 : La courbe de loss du modèle XGB*

La courbe d'apprentissage montre que le modèle converge de manière stable vers un minimum satisfaisant. La *Figure 7* ci-dessous présente ensuite la matrice de confusion issu de ce premier entraînement. On observe que la classe Covid (1) atteint un rappel d'environ 70%, un résultat

correct mais insuffisant pour une application médicale. Les optimisations évoquées précédemment ont donc pour objectif d'améliorer cette valeur, en renforçant la capacité du modèle à détecter correctement les cas Covid.

	Prédit 0	Prédit 1	Prédit 2
Réel 0	0.752	0.138	0.110
Réel 1	0.181	0.699	0.119
Réel 2	0.141	0.232	0.627

Figure 7 : Matrice de confusion lors du premier entraînement du modèle XGB

Plusieurs essais ont alors été réalisés afin de trouver la meilleure combinaison entre le *max\_depth* et le facteur d'amplitude du poids associé à la classe Covid. L'Annexe 5, regroupe l'ensemble des matrices de confusions hybride obtenues pour les différentes configurations testées.

Après analyse et filtrage des résultats, les paramètres offrant le meilleur compromis, en particulier un F1-score optimal pour la classe Covid, ont été retenus :

- *max\_depth* = 6
- Facteur d'amplification du poids de la classe Covid = 1.5

L'entraînement final a ensuite été réalisé sur 300 arbres avec les paramètres ci-dessus. Les *Figure 8* et *Figure 9* ci-dessous illustre l'évolution des performances avant et après optimisation.

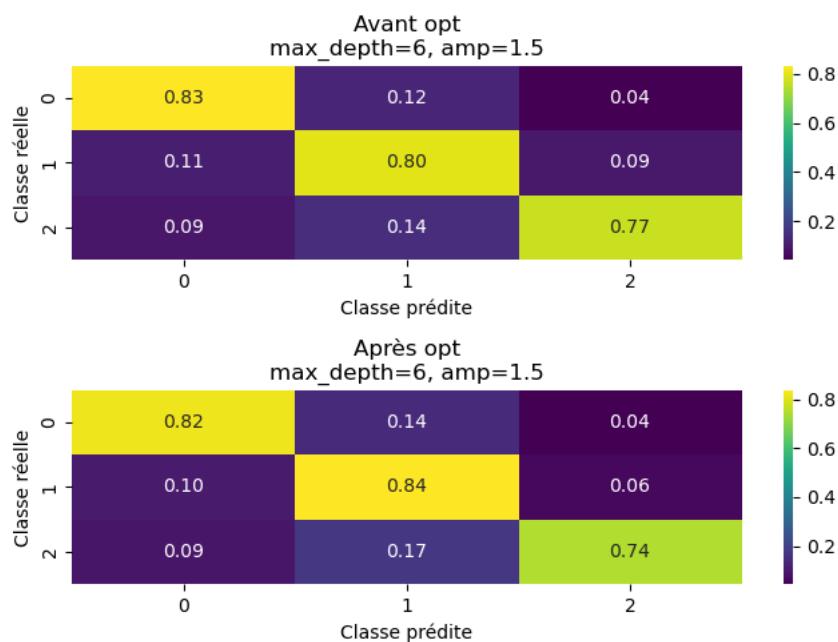


Figure 8 : Matrice de confusion avant et après optimisation

Avant	precision	recall	f1-score	support
0	0.807	0.832	0.819	577
1	0.748	0.798	0.772	578
2	0.849	0.766	0.805	578
accuracy			0.799	1733
macro avg	0.801	0.799	0.799	1733
weighted avg	0.801	0.799	0.799	1733
Après	precision	recall	f1-score	support
0	0.811	0.818	0.814	577
1	0.731	0.837	0.781	578
2	0.877	0.742	0.804	578
accuracy			0.799	1733
macro avg	0.806	0.799	0.800	1733
weighted avg	0.806	0.799	0.800	1733

*Figure 9 : Evolution des rapports de classifications*

Le rappel pour la classe Covid (1) a été amélioré, signe que le modèle confond désormais moins cette classe avec les deux autres classes.

La précision, quant à elle, a été légèrement diminuée, ce qui traduit une hausse modérée du taux de faux positifs.

Globalement, le modèle identifie donc mieux les cas Covid, au prix d'une légère augmentation des faux positifs, cela reste toutefois acceptable dans un contexte médical, où la priorité est de minimiser au maximum les faux négatifs, c'est-à-dire les cas de Covid non détectés.

### 3.2. Evaluation et interprétations

Le modèle optimisé a ensuite été évalué sur le jeu de donnée de test afin d'analyser ses performances sur des images inédites.

La *Figure 10* présente le rapport de classification obtenu sur cet ensemble de test.

Le modèle XGB affiche des performances globalement satisfaisantes avec une accuracy de 0.775, ce qui indique qu'environ 77.5% des images du test sont correctement classées.

	precision	recall	f1-score	support
0	0.802	0.798	0.800	722
1	0.693	0.813	0.748	722
2	0.859	0.715	0.780	722
accuracy			0.775	2166
macro avg	0.785	0.775	0.776	2166
weighted avg	0.785	0.775	0.776	2166

*Figure 10 : Rapport de classification du modèle XGB optimisé sur l'ensemble de test*

La classe Normal (0), présente une bonne précision (80%) et un rappel de 79.8%. Cela signifie que la majorité des images réellement saines sont correctement détectées et que peu d'images d'autres classes sont confondus avec des images saines.

La classe Covid (1), son rappel atteint 81.3%, ce qui est encourageant car le modèle détecte bien les cas Covid. La précision, plus faible (69%), suggère une proportion plus élevée de faux positifs, mais ce compromis reste acceptable dans un contexte médical.

La classe Autres pathologies (2), les performances sont équilibrées, avec une précision élevée (86%) et un rappel correct (72%). Le modèle identifie bien cette classe mais a tendance à confondre certaines images.

Classe prédictive	0	1	2
Classe réelle			
0	0.797784	0.152355	0.049861
1	0.119114	0.813019	0.067867
2	0.077562	0.207756	0.714681

**Figure 11 :** Matrice de confusion du modèle XGB optimisé sur l'ensemble de test

La matrice de confusion en *Figure 11*, confirme les tendances observées précédemment. La majorité des prédictions se situent correctement sur la diagonale, ce qui indique que le modèle distingue globalement bien les trois classes.

La classe Covid-19 (1) présente un taux de rappel élevé (81%), cependant on observe encore des confusions plus importante avec la classe Normal (0 – 12%) qu'avec la classe Autres (2 – 6%).

La classe Normal (0) est également bien reconnue (80%) avec plus de confusion avec la classe Covid-19 (1 – 15%) qu'avec la classe Autres (2 – 5%).

Enfin la classe Autres pathologies (2), est la moins bien détectée avec 71% de prédictions correctes, elle est majoritairement confondue avec la classe Covid (1 – 21%).

En conclusion, le modèle XGB offre des performances correctes avec une bonne capacité à détecter les cas de Covid-19. Il reste néanmoins moins performant qu'un modèle de deep learning, plus profond et avec plus de capacité à capturer la complexité des images radiographiques.

# PARTIE 2 : DEEP LEARNING – DENSENET121

La seconde approche s'inscrit dans le domaine du deep learning et fait appel au transfert d'apprentissage à travers l'utilisation du modèle pré-entraîné *DenseNet121*.<sup>[8]</sup>

## 1. Pré-processing

Les étapes de pré-traitement nécessaires avant l'entraînement du modèle DenseNet121 sont décrites ci-dessous. Elles permettent de structurer les données, de les adapter aux contraintes du réseau neuronal et de garantir une bonne qualité d'apprentissage.

1. **Chargement et structuration des données** : Les images sont récupérées et réparties en quatre classes.
  - 0 → Covid-19
  - 1 → Opacité pulmonaire
  - 2 → Normal
  - 3 → Pneumonie virale
2. **Redimensionnement** : Toutes les images sont ajustées à une taille de 224×224 pixels, la dimension requise par le modèle *DenseNet121* pour son fonctionnement.
3. **Variables explicatives et variable cible** : Les données d'entrée sont constituées de *tensors* d'images (224, 224, 3) en tant que *features* (variables explicatives), et les labels correspondants comme *target* (variable à prédire).
4. **Separation X\_train, X\_test, y\_train, y\_test** : Les données sont divisées en deux ensembles :
  - Train : Ensemble d'entraînement (80%)
  - Test : Ensemble d'évaluation (20%)
5. **Normalisation** : Pour améliorer la vitesse de convergence lors de l'apprentissage, les données sont normalisées afin qu'elles se situent entre 0 et 1.
6. **Pondération des classes** : La fonction *compute\_class\_weight* permet d'attribuer des poids inversement proportionnel à la fréquence de chaque classe afin d'améliorer l'équilibre de l'apprentissage.
7. **Conversion en format .h5** : Les images et labels sont stockés dans un fichier *HDF5* (.h5) afin de faciliter un chargement par batch, évitant ainsi une surcharge de la mémoire RAM pendant l'entraînement dans Google Colab.
8. **Séparation des données d'entraînement en train validation** : La séparation des données de l'ensemble d'entraînement entre les jeux train (85%) et validation (15%) se fait à l'aide d'un découpage aléatoire.

Le jeu de validation est utilisé pour suivre l'évolution au fil des époques d'entraînement. Il permet également d'activer des *callbacks* tels que :

- *EarlyStopping* : Arrêt automatique en cas de stagnation des performances,
- *ModelCheckpoint* : Sauvegarde du meilleur modèle.
- *ReduceLROnPlateau* : Réduit automatiquement le taux d'apprentissage lorsqu'il détecte une stagnation des performances pendant l'entraînement.

Ainsi l'ensemble de ces étapes permet d'obtenir un jeu de données propre, normalisé et suffisamment enrichi pour être exploité efficacement par le modèle *DenseNet121* dans la phase de modélisation.

## 2. Modélisation

### 2.1. Entraînement

Le modèle *DenseNet121* pré-entraîné sur ImageNet sert d'extracteur de caractéristiques. Des couches entièrement connectées sont ensuite ajoutées en sortie pour permettre la classification en quatre catégories. L'architecture du modèle est en *Annexe 2*.

Par la suite, 50 couches ont été dégelées afin de réaliser un fine-tuning sur notre jeu de données. Les paramètres retenus pour la compilation du modèle sont les suivants :

- Optimiseur : Adam, avec un learning rate fixé à  $1.10^{-5}$
- Fonction de perte : « *Sparse categorical crossentropy* »
- Métrique : Accuracy

Les *callbacks* décrits précédemment ont également été intégrés. Le modèle est ensuite entraîné sur 10 époques. La *Figure 12* ci-dessous présente l'évolution de la loss ainsi que de l'accuracy sur les ensembles d'entraînement et de validation au cours de l'apprentissage.

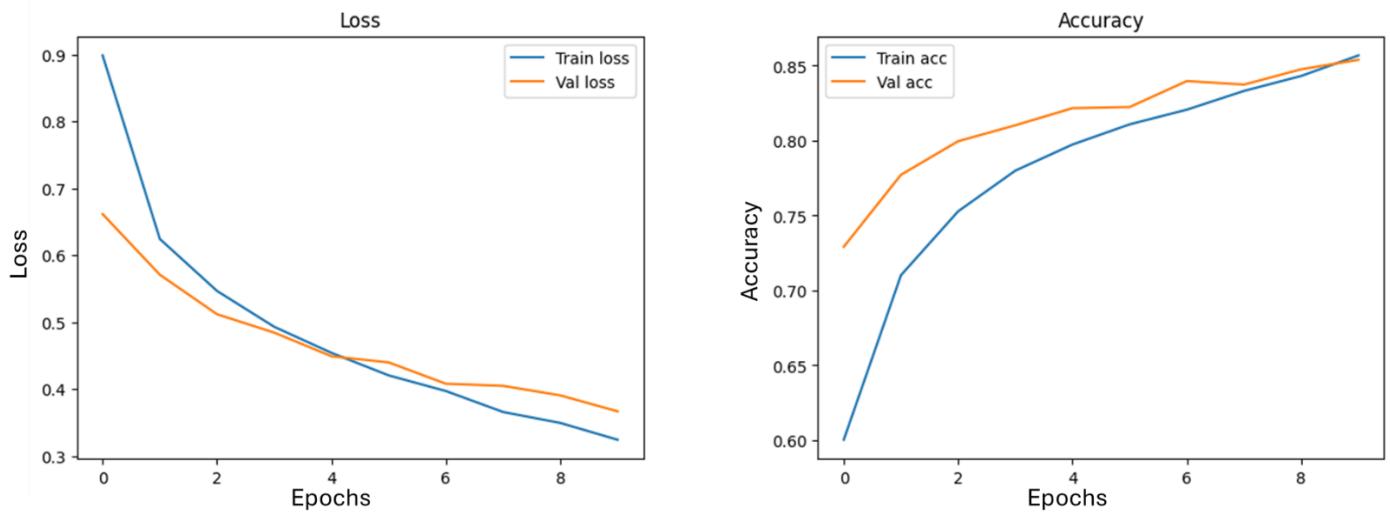


Figure 12 : Evolution de la loss et de l'accuracy du modèle *DenseNet121*

Dans la phase initiale (0 à 2 epochs), l'accuracy d'entraînement augmente rapidement, ce qui traduit une montée en compétence rapide du modèle. La validation suit la même tendance, indiquant qu'à ce stade le modèle généralise correctement.

La phase intermédiaire (3 à 6 epochs), les deux courbes progressent de manière parallèle. L'écart entre les deux reste minime, montrant que le modèle ne surapprend pas.

Enfin, dans la phase finale (7 à 9 epochs), les deux courbes se stabilisent vers 0.85, l'écart est faible ( $< 0.02$ ) ce qui confirme une bonne généralisation du modèle.

## 2.2. Evaluation et interprétations

Une fois la phase d'entraînement achevée, le modèle est appliqué à l'ensemble de test afin d'évaluer sa capacité à généraliser sur des données inédites. Pour cela plusieurs métriques en classification sont utilisées, notamment la précision, le rappel et score F1, calculés pour chacune des classes.

*Tableau 4 : Rapport de classification de l'évaluation*

	Précision	Recall	F1-score	Support
<b>Covid-19</b>	0.76	0.76	0.76	723
<b>Opacité pulmonaire</b>	0.85	0.82	0.84	1203
<b>Normal</b>	0.89	0.90	0.90	2038
<b>Pneumonie virale</b>	0.89	0.96	0.92	269
<b>Accuracy</b>			0.86	4233
<b>Macro avg</b>	0.85	0.86	0.85	4233
<b>Weighted avg</b>	0.86	0.86	0.86	4233

Le *Tableau 4* ci-dessus présente le rapport de classification obtenu, regroupant l'ensemble de ces indicateurs ainsi que les métriques globales telle que l'accuracy.

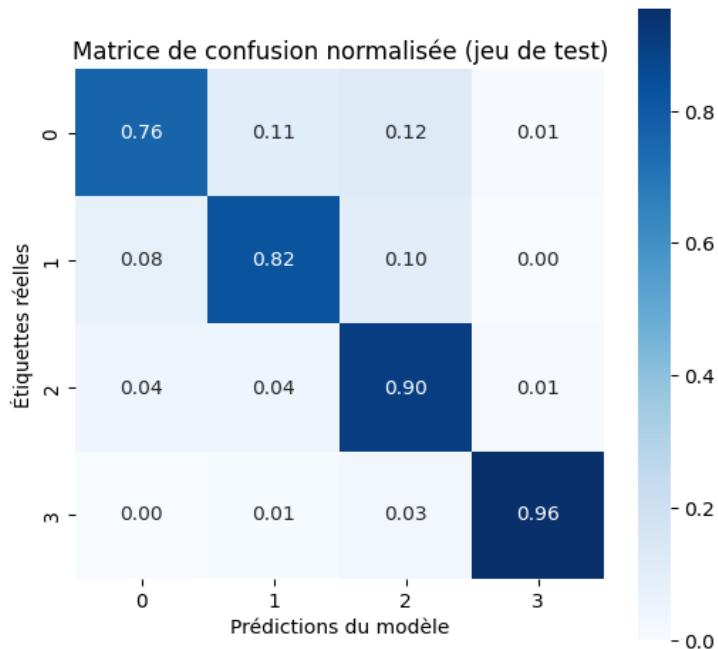
La précision pour chaque classe est élevée ( $> 76\%$ ), ce qui indique que le modèle ne fait relativement pas d'erreur lorsqu'il prédit une classe positive.

Le modèle a un excellent rappel pour la classe 3 (pneumonie virale) avec un rappel de 96%, ce qui signifie qu'il identifie la plupart des instances de cette classe. Le rappel est plutôt bon pour les autres classes également.

Le F1-score est très bon pour toutes les classes (autour de 86%), ce qui montre un bon compromis entre rappel et précision.

L'accuracy globale est de 86%, ce qui est un très bon résultat pour un modèle de classification d'images. Cela signifie que le modèle fait une prédiction correcte pour environ 86% des images.

Pour compléter cette analyse, la *Figure 13* présente la matrice de confusion, un outil indispensable permettant de visualiser en détail les erreurs de classification commises par le modèle.

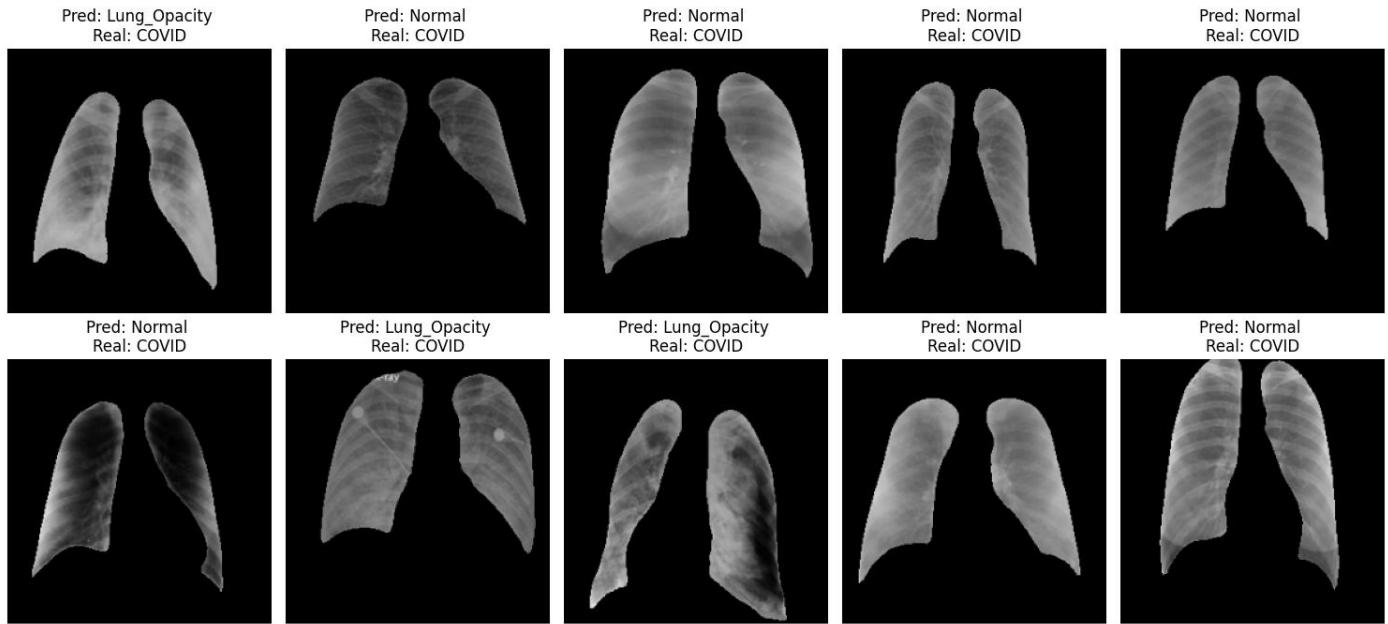


**Figure 13 :** Matrice de confusion d'évaluation

Les valeurs sur la diagonale correspondent aux prédictions correctes, tandis que les autres cases indiquent les confusions entre classes.

- La classe 0 – Covid-19 : Le modèle identifie correctement 76% des radiographies Covid-19. Les principales confusions concernent les classes 1-opacité pulmonaire (11%) et 2-normal (12%), ce qui est cohérent, certaines atteintes Covid peuvent présenter des caractéristiques proches.
- La classe 1 – Opacité pulmonaire : Le modèle obtient 82% de bonnes prédictions. Les erreurs se répartissent principalement vers la classe 2-normal (10%) et 0-Covid-19 (8%).
- La classe 2 – Normal : C'est l'une des classes les mieux reconnues avec 90% de bonnes prédictions. Les confusions sont faibles, 4% avec 0-Covid et 1-opacité pulmonaire et 1% avec 3-pneumonie virale. Le modèle a une bonne capacité à distinguer les images normales des images pathologiques.
- La classe 3 – Pneumonie virale : Le modèle obtient un excellent taux de reconnaissance de 96%, le plus élevé parmi les quatre classes. Les erreurs sont très faibles 3% vers 2-normal et 1% vers 1-opacité pulmonaire.

La *Figure 14* illustre plusieurs exemples de prédictions erronées, mettant en évidence les principales confusions réalisées par le modèle. On observe notamment que la classe Covid-19 est fréquemment confondue avec les classes normal et opacité pulmonaire. Ces confusion s'expliquent par la similarité visuelle de certains motifs radiologiques entre ces catégories, en particulier lorsque les signes d'infection sont discrets ou atypiques. Cela montre que, malgré de bonnes performances globales, le modèle peut rencontrer des difficultés à distinguer des cas limites ou peu marqués.



**Figure 14 :** Principales confusions du modèle DenseNet121

Au-delà des erreurs de classification, il est également important d'évaluer la capacité du modèle à distinguer correctement chaque classe selon différents seuils de décision. Pour cela, les courbes ROC (Receiver Operating Characteristic) et leurs surfaces sous la courbe (AUC) constituent des outils particulièrement pertinents. Elles permettent d'analyser la sensibilité et la spécificité du modèle pour chaque classe, indépendamment d'un seuil fixé.

L'étude des courbes ROC/AUC permet ainsi de compléter l'analyse fournie par le rapport de classification et la matrice de confusion, en offrant une vision plus globale de la robustesse du modèle face aux variations de seuil. Des valeurs d'AUC élevées indiquent que le modèle distingue efficacement les classes, même dans les situations les plus ambiguës.

Les courbes ROC présentées dans la *Figure 15* montrent que le modèle distingue très bien les différentes classes, avec des scores AUC élevés pour chacune d'elles. La classe pneumonie virale atteint un AUC parfait de 1.0, indiquant une séparation optimale. Les classes normal (AUC=0.96) et opacité pulmonaire (AUC=0.96) présentent également d'excellentes performances, témoignant d'une forte capacité du modèle à différencier ces catégories. Enfin, la classe Covid-19 obtient un AUC de 0.95, ce qui demeure très satisfaisant malgré les confusions observées précédemment avec les classes normal et opacité pulmonaire.

Globalement, ces courbes confirment la robustesse du modèle et sa capacité à maintenir un haut niveau de discrimination entre les classes, même dans un contexte multi-classes. Cependant, malgré de bons résultats globaux du modèle, la classe Covid-19 reste celle qui est la moins bien détectée, avec des scores inférieurs aux autres catégories.

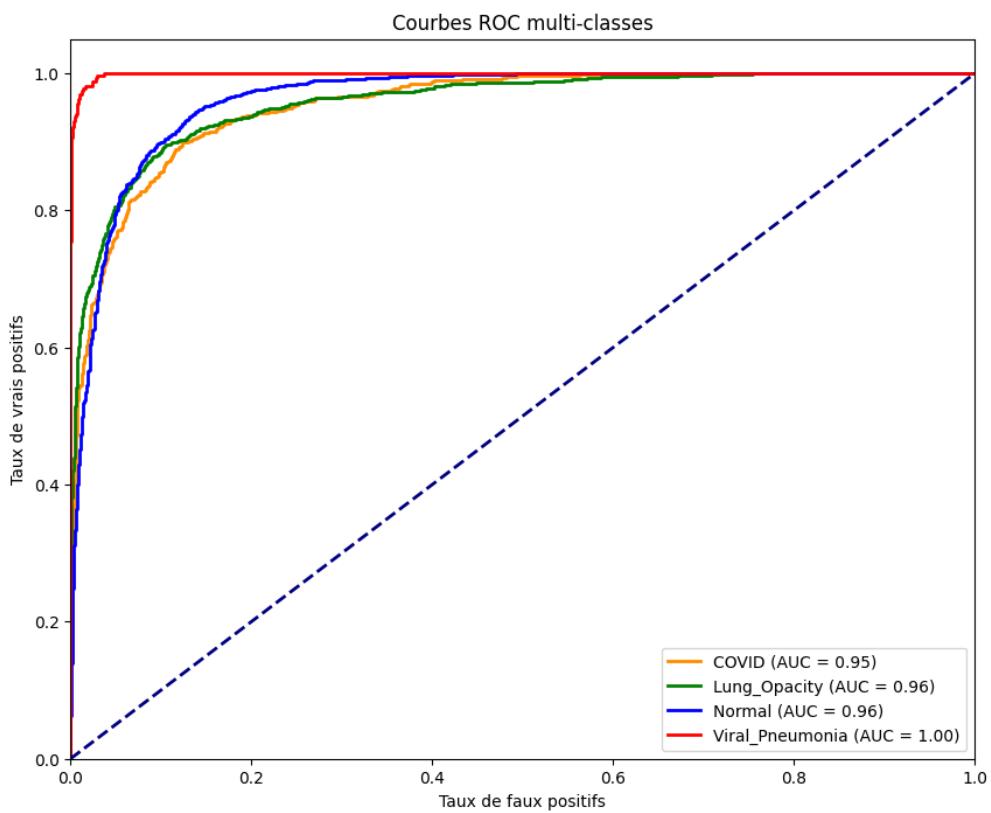


Figure 15 : Courbes ROC/AUC du modèle DenseNet121

Dans l'ensemble le modèle *DenseNet121* se révèle performant et fiable, offrant une bonne capacité de généralisation et une discrimination robuste entre les différentes classes, malgré quelques confusions résiduelles, en particulier autour de la classe Covid-19.

# PARTIE 3 : DEEP LEARNING – INCEPTIONV3

La troisième approche s'inscrit également dans le domaine du deep learning et fait appel au transfert d'apprentissage à travers l'utilisation du modèle pré-entraîné *InceptionV3*.<sup>[9]</sup> En s'appuyant sur la stratégie proposé par Alqatani *et al.*<sup>[10]</sup> Leur travail est fondé sur l'utilisation d'*InceptionV4* pour la détection du Covid-19 à partir de radiographies thoraciques, a montré d'excellents résultats. En nous inspirant de leur méthodologie, nous utilisons ici le modèle pré-entraîné *InceptionV3* et adaptons leur approche à notre propre jeu de données.

## 1. Pré-processing

Dans cette partie, nous présentons la stratégie adoptée pour corriger le déséquilibre entre les classes au sein du jeu de données. Comme il s'agit d'images médicales, il serait délicat d'appliquer des transformations trop invasives (telles que translation, zoom, rotation...) susceptibles de dénaturer les structures anatomiques.

Pour cette raison, nous avons choisi une approche centrée sur l'équilibrage des classes sans recourir à un sur-échantillonnage. Ainsi le nombre d'images par catégorie a été harmonisé en prenant la classe Covid-19 comme référence. Les catégories sont alors réorganisées comme suit :

- 0 → Sain : 3500 images
- 1 → Covid-19 : 3500 images
- 2 → Autres : Regroupant Opacité pulmonaire et Pneumonie virale, avec 3500 images réparties équitablement entre ces deux catégories

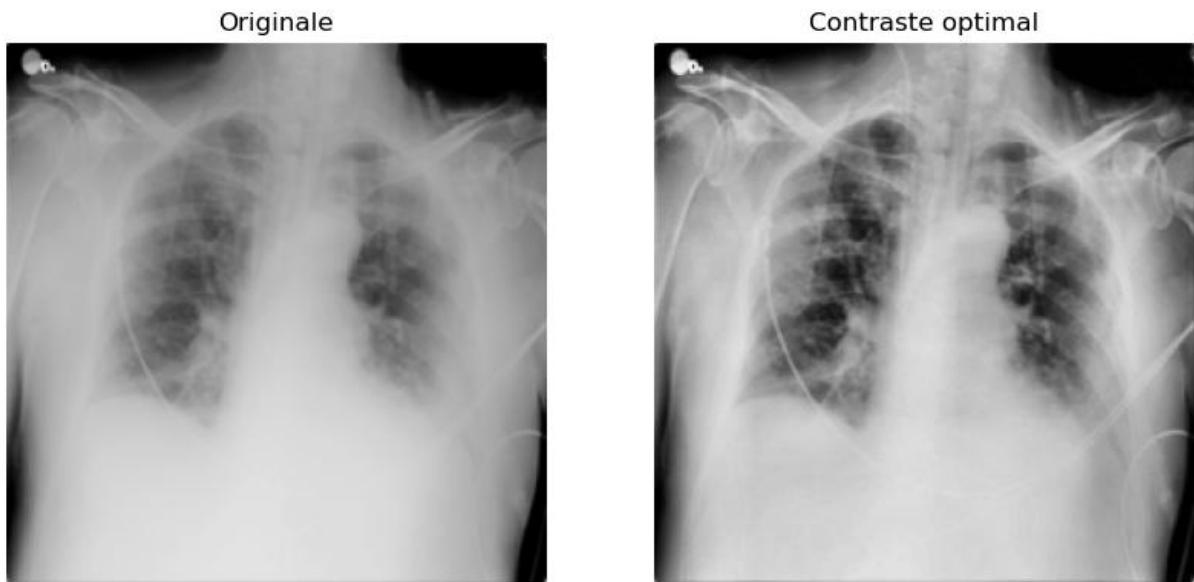
Le pré-traitement et la séparation des données ont ensuite été réalisés selon les étapes suivantes :

1. Mise de côté d'un lot de 10 images par catégorie (Covid, Sain, Autres) entièrement inédites pour le modèle et réservées à la soutenance.
2. Sélection aléatoire de 3500 images pour chacune des catégories Covid, Sain et Autres.
3. Redimensionnement et application d'un contraste et du masque correspondant.
4. Division des données en ensemble d'entraînement, de validation et de test selon les proportions 70/15/15.

### *Le choix du contraste*

Le contraste joue un rôle essentiel dans l'amélioration de la qualité visuelle des images et aide le modèle à mieux distinguer les différentes structures présentes sur les radiographies. Pour cela, nous avons utilisé la méthode *CLAHE*, qui permet d'ajuster le contraste. Les différents essais réalisés sont présentés en *Annexe 3*.

Ainsi, la *Figure 16* illustre clairement la différence entre une radiographie brute et la même image après application de la méthode *CLAHE*, en mettant en évidence l'amélioration du contraste obtenue.



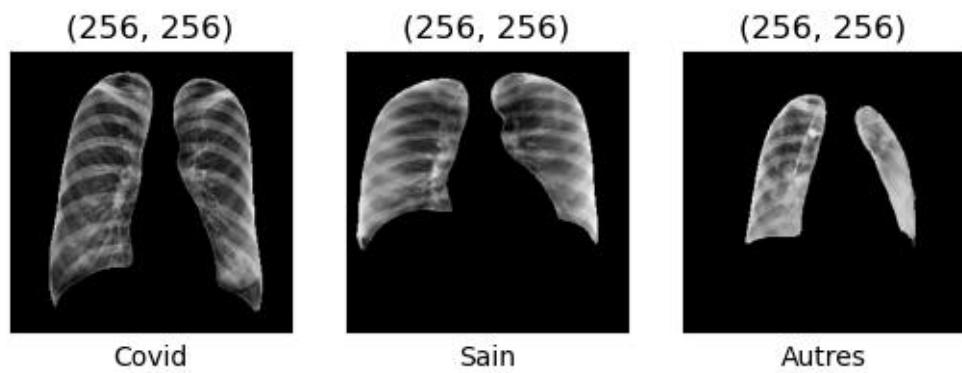
**Figure 16 :** Avant / Après application du filtre CLAHE

Les paramètres retenus pour l'application du *CLAHE* sont un *clipLimit* de 2.5 et un *tileGridSize* de (8,8), ceux-ci offrant le meilleur équilibre entre amélioration du contraste et préservation des structures anatomiques.

#### *Redimensionnement et application du masque*

Comme mentionné dans la partie dédiée à l'exploration des données, les radiographies et les masques n'ont pas les mêmes dimensions. Il est donc nécessaire d'uniformiser les tailles avant de pouvoir appliquer correctement les masques.

Ainsi, toutes les images ont été redimensionnées à la taille du masque, soit (256x256), avant l'application de ce dernier. La *Figure 17* illustre un exemple de ce traitement pour chacune des trois catégories, Covid, Sain et Autres.



**Figure 17 :** Pré-traitement des images radiographiques

## Séparation et sauvegarde des données

Afin de permettre un entraînement optimal du modèle les données ont été réparties en trois ensembles distincts :

- 70% des données pour l'ensemble d'entraînement
- 15% des données pour l'ensemble de validation
- 15% des données pour l'ensemble de test

Pour ce faire, une première séparation a été effectuée entre un ensemble d'entraînement et de test selon un ratio 85/15. Ensuite, l'ensemble d'entraînement a fait l'objet d'un second split pour être divisé en un sous-ensemble d'entraînement et un sous-ensemble de validation avec un ratio 82,4/17,6, permettant d'obtenir les proportions finales souhaitées 70/15/15.

Tableau 5 : Répartition des données

Split	Sain	Covid-19	Autres	Total
Entraînem ent	2451	2451	2451	7353
Validation	524	524	524	1572
Test	525	525	525	1575

Le *Tableau 5* ci-dessus résume la réparation finale des données après cette procédure. Enfin, les ensembles ainsi constitués ont été sauvegardés au format *compressed.npz* pour leur utilisation ultérieure.

## 2. Modélisation

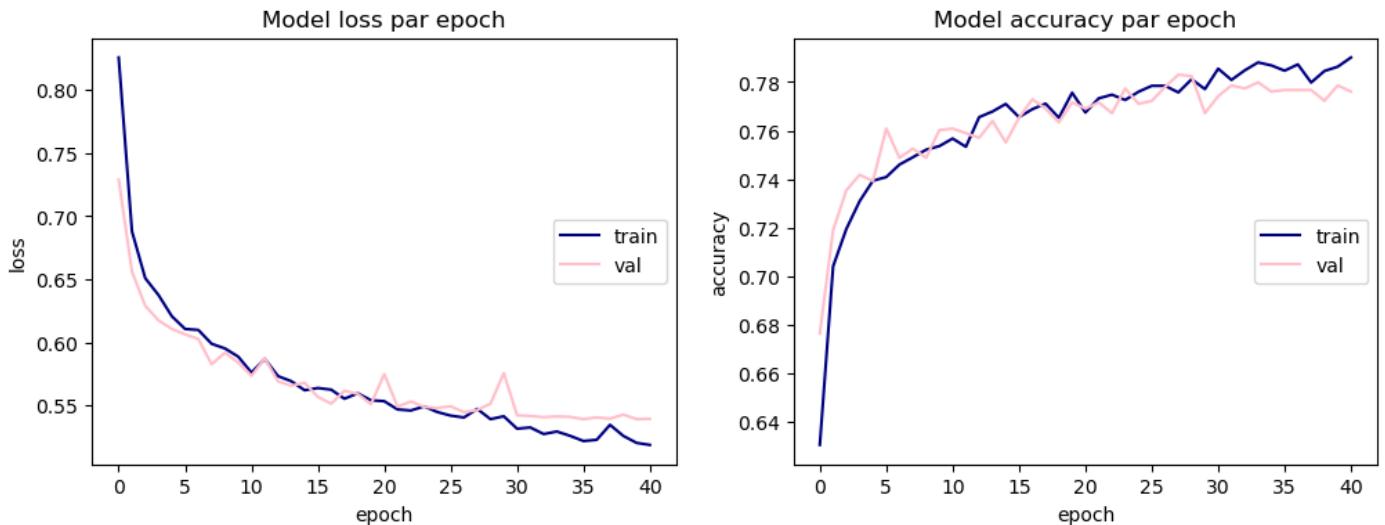
### 2.1. Entrainement

Avant d'être utilisé par le modèle sélectionné, les données ont été soumises à un pré-traitement spécifique, incluant notamment une étape de standardisation. Pour cela, la fonction *preprocess\_input* est appliquée, permettant de ramener les valeurs des images dans l'intervalle [-1,1].

Comme évoqué précédemment, la stratégie d'entraînement adoptée repose sur les choix décrits dans l'article d'Alqatani *et al.*<sup>[10]</sup>. Après un *resizing* des images au format attendu par le modèle (299x299), la tête de réseau a été modifiée afin de correspondre à nos trois classes. L'architecture du modèle est présentée en *Annexe 4*, tandis que les principaux paramètres retenus sont :

- Optimiseur : SGD (*learning\_rate* =  $1.10^{-4}$ , *momentum*=0.9)
- Fonction de perte : « *Sparse categorical crossentropy* »
- Métrique : Accuracy
- Callbacks : *EarlyStopping*, *ReduceLROnPlateau*, *ModelCheckpoint*.

L'entraînement a été réalisé sur 100 époques (*EarlyStopping* : 41 époques), conformément aux recommandations de l'article, et les courbes d'accuracy et de loss obtenues sont présentées dans la *Figure 18* ci-dessous :



**Figure 18 :** Courbes de loss et d'accuracy du modèle InceptionV3 sans couches dégelées

Les courbes ci-dessus permettent de suivre le comportement du modèle *InceptionV3* au cours de l'entraînement. On constate que la loss d'entraînement diminue de manière régulière au fil des époques. La loss de validation adopte une tendance similaire avec quelques fluctuations. L'absence d'écart important entre les deux courbes indique que le modèle ne présente pas de sur-apprentissage notable et qu'il parvient à conserver une bonne capacité de généralisation.

Les courbes d'accuracy confirment cette observation. En effet, l'accuracy d'entraînement augmente rapidement au début avant de se stabiliser autour de 78%. L'accuracy de validation suit la même dynamique et reste très proche de cette dernière. Cette cohérence entre les deux courbes traduit une bonne capacité de généralisation du modèle.

Globalement, ces résultats montrent que l'entraînement du modèle *InceptionV3* s'est déroulé de manière stable et efficace.

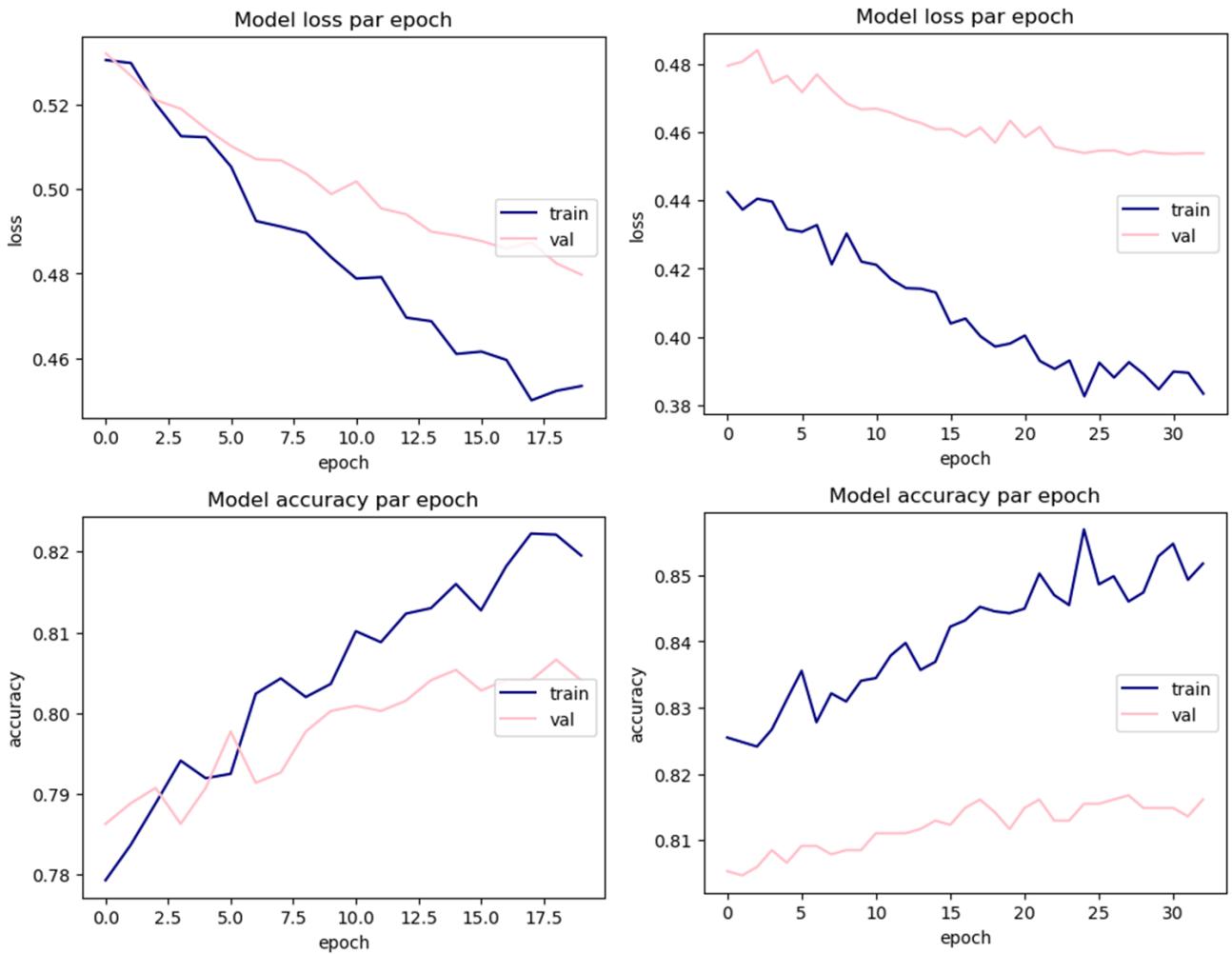
Afin d'améliorer davantage les performances du modèle, nous avons choisi de dégeler les 50 dernières couches du réseau. Cependant, cette stratégie a rapidement entraîné une forte divergence entre les courbes d'entraînement et de validation, se traduisant par :

- Un écart de loss d'environ 0.27
- Une différence d'accuracy proche de 12%

Ce comportement est caractéristique d'un surapprentissage important, indiquant que le modèle mémorise trop les données d'entraînement au détriment de sa capacité de généralisation.

Face à ce constat, nous avons réduit le nombre de couches dégelées à 20. Un premier essai sur 20 époques a montré une tendance encourageante, ce qui nous a conduit à poursuivre l'entraînement en ajoutant 80 époques supplémentaires, pour un total de 100 époques (*EarlyStopping* : 55 époques).

Les courbes correspondantes sont présentées dans la *Figure 19* ci-dessous :



**Figure 19 :** Courbes de loss et d'accuracy du modèle InceptionV3 avec 20 couches dégelées

Après ce réglage, l'écart entre les courbes de loss n'est plus que de 0.07, et celui de l'accuracy d'environ 3%, ce qui correspond à des valeurs désormais raisonnables et compatibles avec un bon niveau de généralisation.

## 2.2. Evaluation et interprétations

Le modèle, une fois finalisé, a été évalué sur l'ensemble de test afin d'analyser son comportement sur des données totalement nouvelles. Le *Tableau 6* présente le rapport de classification, qui résume les performances du modèle sur l'ensemble de test.

**Tableau 6 :** Rapport de classification du modèle InceptionV3

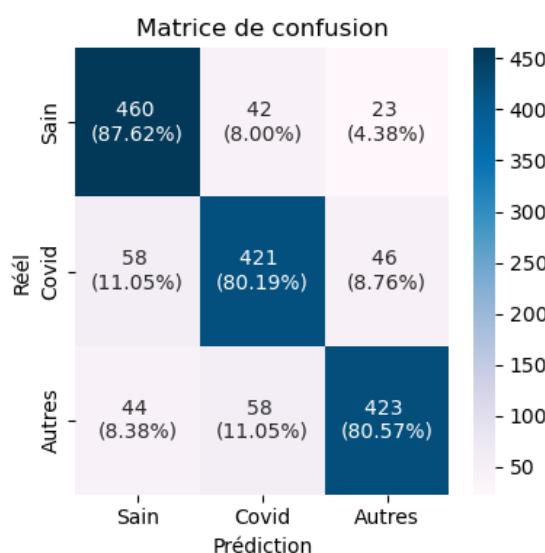
	Précision	Recall	F1-score	Support
<b>Sain</b>	0.82	0.88	0.85	525
<b>Covid-19</b>	0.81	0.80	0.80	525
<b>Autres</b>	0.86	0.81	0.83	525
<b>Accuracy</b>			0.83	1575
<b>Macro avg</b>	0.83	0.83	0.83	1575
<b>Weighted avg</b>	0.83	0.83	0.83	1575

Le rapport de classification met en évidence des performances globalement équilibrées et cohérentes entre les trois classes, avec une accuracy globale de 83%, ce qui est satisfaisant compte tenu de la complexité des images radiographiques.

Pour la classe Sain, les résultats sont relativement bons. Le modèle obtient une précision de 82% et un rappel de 88%, indiquant qu'il parvient à identifier correctement une large majorité des radiographies réellement saines, tout en commettant relativement peu de faux positifs. Le F1-score de 85%, reflète un bon équilibre entre précision et rappel, confirmant que les structures pulmonaires saines sont bien reconnues.

Concernant le Covid-19, les performances sont légèrement inférieures mais restent satisfaisantes, avec une précision de l'ordre de 80%. Cela suggère que le modèle détecte correctement les signes caractéristiques de l'infection. Le F1-score de 80% traduit une cohérence dans le comportement du modèle sur cette classe complexe.

La classe Autres, regroupant l'opacité pulmonaire et la pneumonie virale, obtient les meilleures performances en précision (86%), ce qui signifie que lorsque le modèle prédit cette catégorie, il se trompe peu. Le rappel, à 81%, reste satisfaisant et indique une capacité correcte à récupérer la majorité des cas réels appartenant à cette classe. Le F1-score de 83% confirme cette bonne segmentation.



**Figure 20 :** Matrice de confusion du modèle InceptionV3

Afin d'obtenir une vision plus fine des erreurs commises par le modèle, il est également pertinent d'examiner la matrice de confusion en *Figure 20*, qui met en évidence la nature des confusions entre les différentes classes de l'ensemble de test.

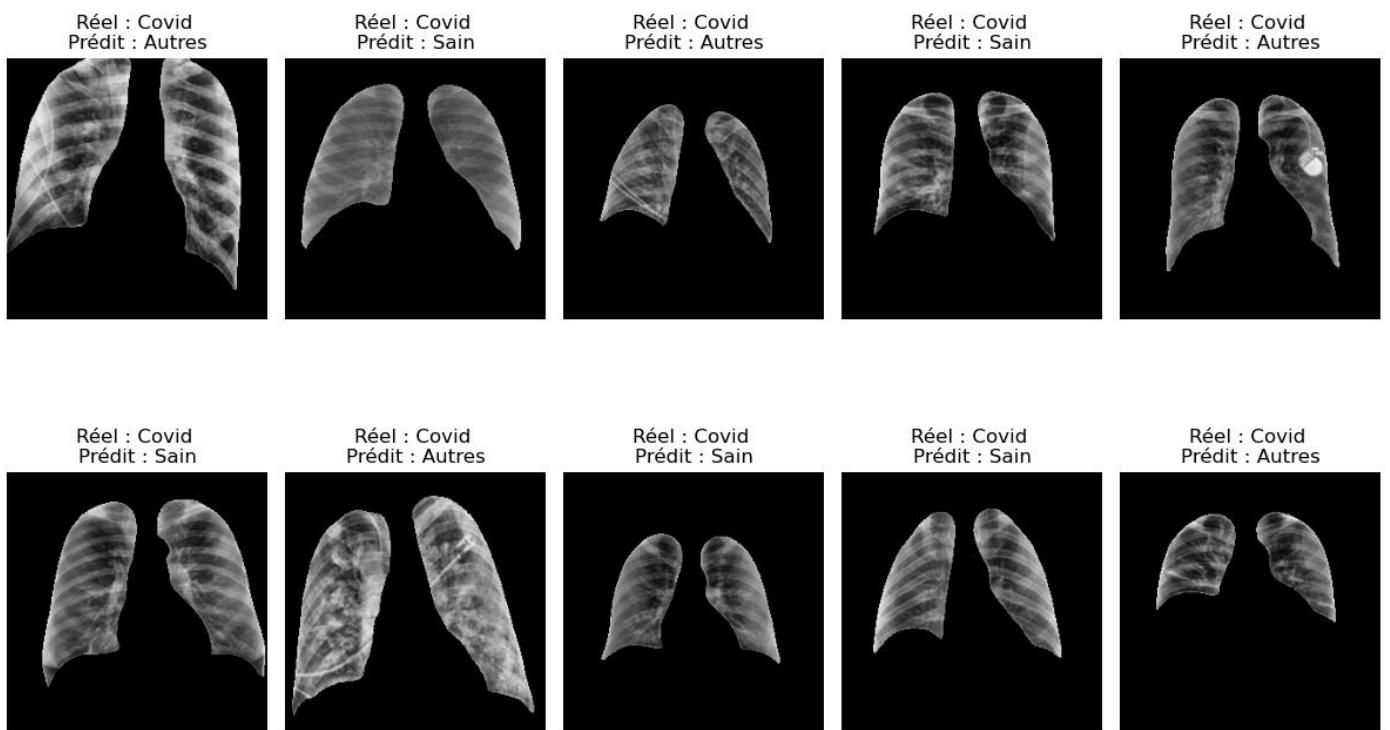
Elle montre que le modèle distingue globalement bien les trois classes, avec des taux de bonnes classification compris entre 80% et 88%.

La classe Sain est la mieux reconnue (87,62%), avec quelques confusion avec Covid (8%) et Autres (4%).

La classe Covid présente un taux correct de 80%, mais le modèle la confond légèrement plus avec la classe Sain (11%) qu'avec la classe Autres (8%).

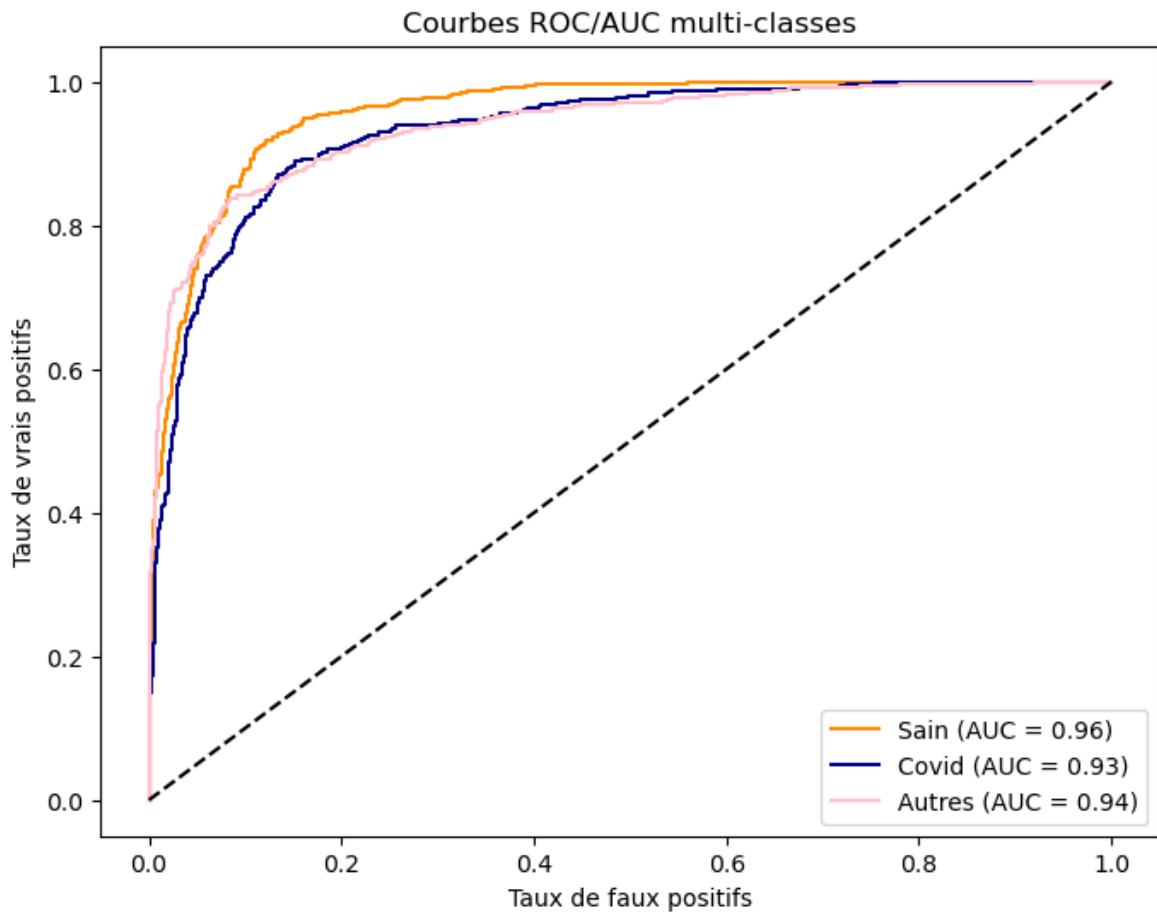
Enfin, la classe Autres obtient également environ 80% de bonnes prédictions avec des erreurs légèrement plus dirigées vers la classe Covid (11%) que vers la classe Sain (8%).

La *Figure 21* illustre plusieurs exemples mal classés par le modèle. On observe que ces erreurs restent relativement équilibrées entre les deux catégories (Sain et Autres), suggérant que le modèle ne favorise pas une classe particulière.



***Figure 21 : Principales confusions du modèle InceptionV3***

Afin de compléter cette analyse et d'évaluer plus globalement la capacité du modèle à distinguer les classes, il est utile d'examiner les courbes ROC/AUC, qui permettent de montrer la qualité de séparation entre les différentes catégories.



**Figure 22 :** Courbes ROC/AUC du modèle *InceptionV3*

La *Figure 22* présente les courbes ROC/AUC obtenues pour les trois classes du modèle *InceptionV3*. Les valeurs d’AUC, comprises entre 0.93 et 0.96 indiquent une très bonne capacité du modèle à distinguer chaque catégorie.

La classe Sain (AUC=0.96) est la mieux séparée, les classes Covid (AUC=0.93) et Autres (AUC=0.94) présentent des performances légèrement inférieures, cohérente avec les confusions observées précédemment, mais restent à un niveau satisfaisant.

La forme globalement élevée et régulière des courbes confirme que le modèle maintient une bonne proportion de vrais positifs tout en limitant les faux positifs.

Ces résultats montrent que le modèle *InceptionV3* offre une bonne capacité de discrimination entre les classes, avec des performances stables et cohérentes à travers l’ensemble des métriques utilisées.

# CONCLUSION

Durant notre étude, nous avons exploré différentes approches afin de répondre à la problématique de classification de radiographies thoraciques pour la détection des cas de Covid-19.

La première approche s'est focalisée sur l'utilisation d'un modèle de machine learning, après plusieurs essais, le modèle retenu est *XGBoost*. Il offrait les meilleures performances. Après optimisation et évaluation du modèle sur l'ensemble de test, les résultats obtenus sont les suivants :

- Accuracy globale : 77.5%
- Précision de la classe Covid-19 : 69.3%
- Rappel de la classe Covid-19 : 81.3%
- F1-score de la classe Covid-19 : 74.8%

Ces résultats sont satisfaisants pour un modèle de machine learning. Néanmoins compte tenu de la nature des images, il apparaît plus pertinent d'explorer une approche basée le deep learning.

La seconde approche s'est ainsi appuyée sur un modèle de deep learning, le modèle *DenseNet121*. Après entraînement et fine-tuning du modèle, l'évaluation sur l'ensemble de test conduit aux performances suivantes :

- Accuracy globale : 86%
- Précision de la classe Covid-19 : 76%
- Rappel de la classe Covid-19 : 76%
- F1-score de la classe Covid-19 : 76%

Malgré une très bonne accuracy globale, la classe Covid-19 demeure la moins bien détectée, affichant des scores inférieurs à 80%. Ceci souligne la difficulté du modèle à distinguer précisément cette catégorie particulière.

La troisième approche repose également sur un modèle de deep learning, basé sur *InceptionV3*. Après un entraînement puis une phase de fine-tuning, les résultats obtenus sont les suivants :

- Accuracy globale : 83%
- Précision de la classe Covid-19 : 81%
- Rappel de la classe Covid-19 : 80%
- F1-score de la classe Covid-19 : 80%

Ces résultats montrent un équilibre satisfaisant entre la performance globale et celle obtenue sur la classe d'intérêt, Covid-19. Le modèle présente donc des performances encourageantes.

Cependant, en comparant l'ensemble des approches testées, la classe Covid-19 a été légèrement plus confondue avec la classe Sain qu'avec les autres pathologies. C'est ce qui renforce la complexité dans la détection de la classe Covid-19. En effet, si elle était confondue avec une

autre pathologie pulmonaire, le patient serait tout de même pris en charge. Or, le fait que les modèles ont tendance à classer les patients atteints de Covid-19 en patients sains conduit, dans certains cas, à la non-prise en charge de ces derniers. C'est là tout l'enjeu dans la conception de ces modèles.

Dans un cadre médical, à notre sens, les trois modèles ne peuvent être utilisés tels quels, ils méritent encore des ajustements supplémentaires afin de réduire les erreurs commises notamment la classification des patients Covid-19 en patients sains.

Le modèle de machine learning peut encore être amélioré en ajustant plus finement les hyperparamètres et en augmentant le jeu de données. Mais il reste tout de même plus pertinent de se diriger vers des modèles de deep learning lorsqu'il s'agit de traiter des images.

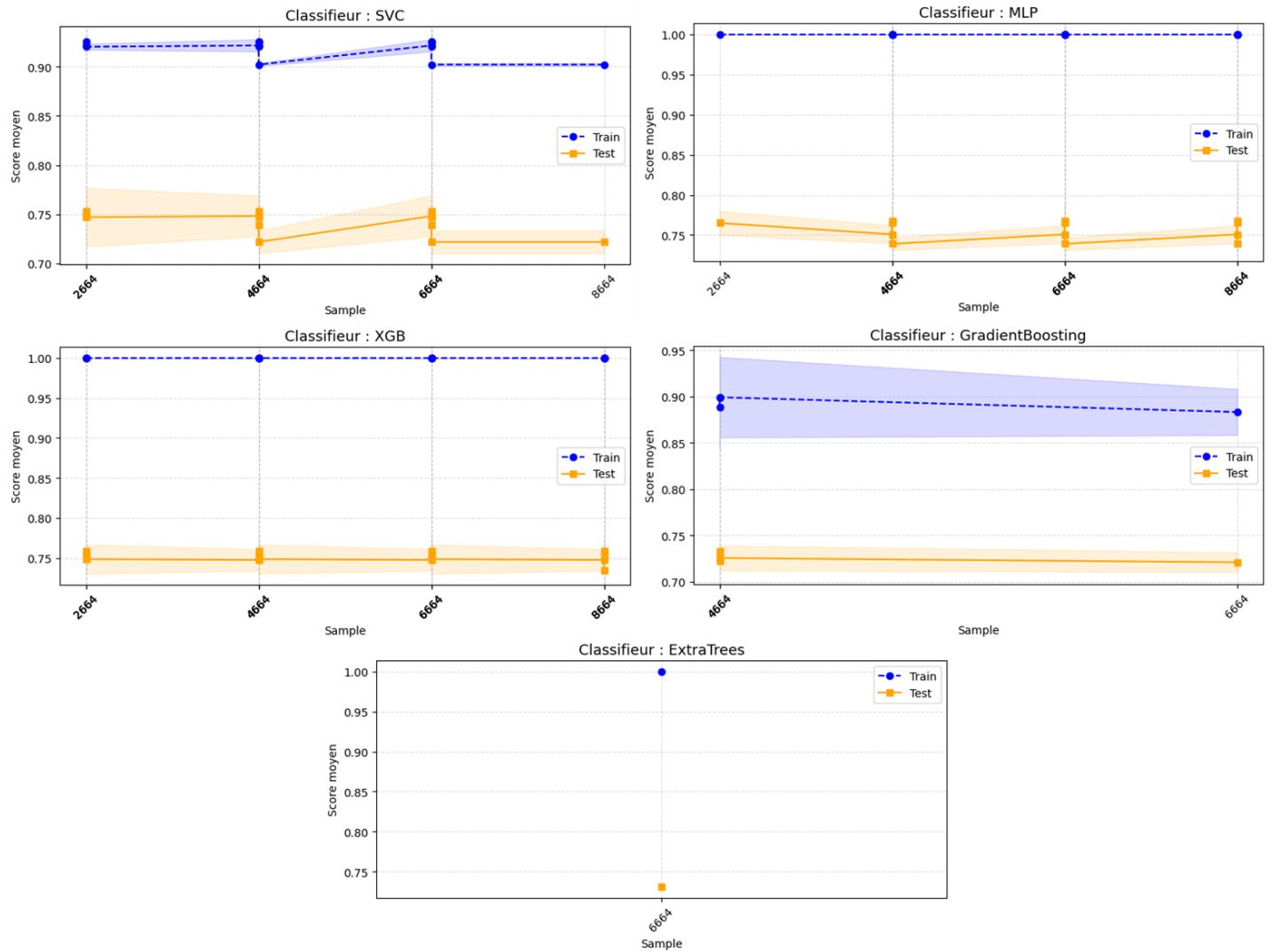
Concernant les deux modèles de deep learning qui ont été utilisés, *DenseNet121* et *InceptionV3*, leur différence repose sur leur mode de fonctionnement. En effet, *DenseNet121* se distingue par sa structure compacte où chaque couche réutilise les informations de toutes les précédentes, tandis qu'*InceptionV3* analyse l'image à plusieurs échelles en parallèle grâce à ses modules multibranches, ce qui le rend plus lourd mais capable de capturer des motifs plus complexes.

Cela explique que *DenseNet121*, grâce à sa forte capacité de réutilisation des caractéristiques, obtienne de meilleures performances globales mais se montre plus sensible sur la classe Covid-19. A l'inverse, *InceptionV3*, légèrement moins performant en moyenne, présente des résultats plus équilibrés entre les classes, ce qui traduit une meilleure robustesse face aux variations interclasses.

Ces résultats encouragent à poursuivre l'exploration de modèles plus spécialisés ou d'approches hybride (combinaison de plusieurs modèles, attention mechanism, *CNN* + *XGB*...) afin d'améliorer encore la sensibilité pour les cas Covid-19, critère essentiel dans un contexte médical réel.

# ANNEXES

Annexe 1 : Les scores moyens en fonction du volume de données par classifieur



Annexe 2 : Architecture du modèle DenseNet121

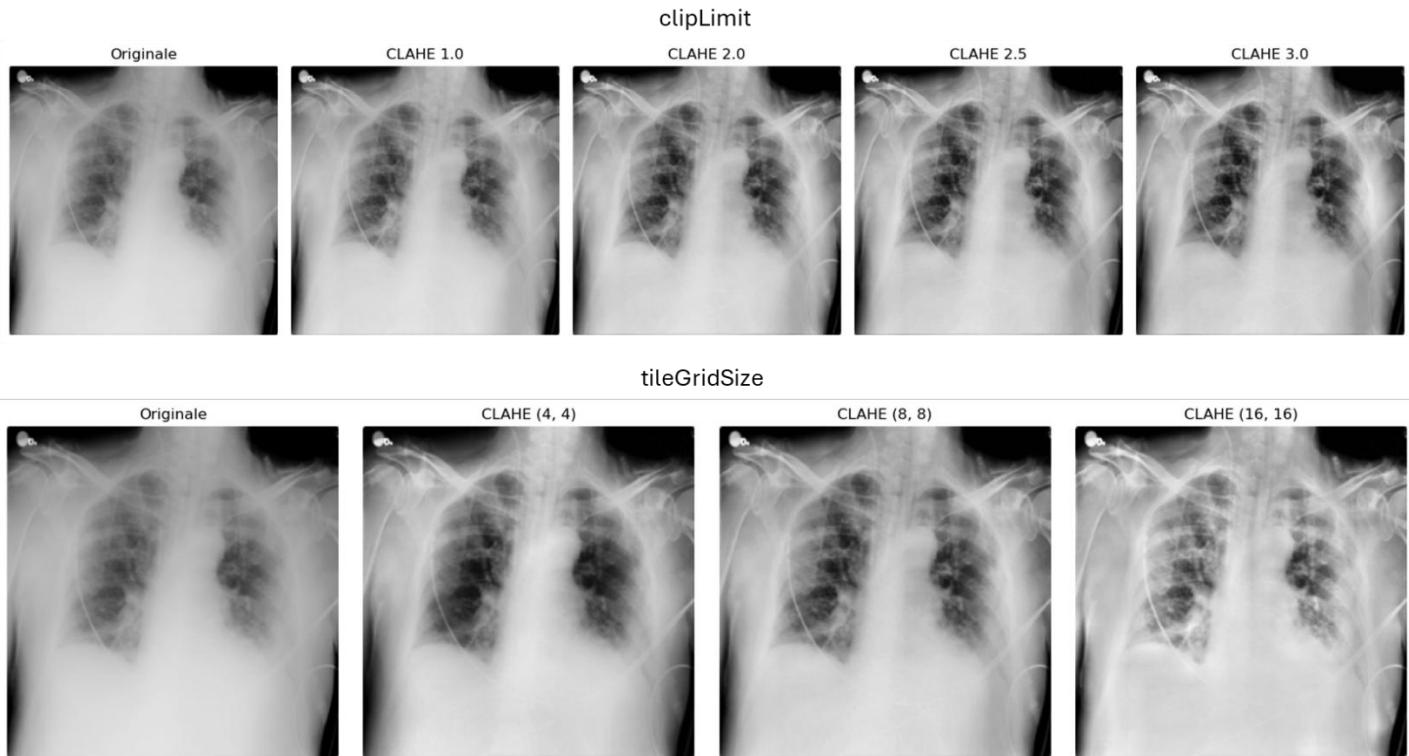
Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
global_average_poo... (GlobalAveragePool...)	(None, 1024)	0	relu[0][0]
dense (Dense)	(None, 1024)	1,049,600	global_average_p...
dropout (Dropout)	(None, 1024)	0	dense[0][0]
dense_1 (Dense)	(None, 512)	524,800	dropout[0][0]
dropout_1 (Dropout)	(None, 512)	0	dense_1[0][0]
dense_2 (Dense)	(None, 4)	2,052	dropout_1[0][0]

Total params: 8,613,956 (32.86 MB)

Trainable params: 1,576,452 (6.01 MB)

Non-trainable params: 7,037,504 (26.85 MB)

Annexe 3 : Variation des paramètres CLAHE



Annexe 4 : Architecture du modèle InceptionV3

Layer (type)	Output Shape	Param #
input_layer_5 (InputLayer)	(None, 256, 256, 3)	0
resizing_2 (Resizing)	(None, 299, 299, 3)	0
inception_v3 (Functional)	(None, 8, 8, 2048)	21,802,784
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 3)	6,147

Total params: 21,808,931 (83.19 MB)

Trainable params: 6,147 (24.01 KB)

Non-trainable params: 21,802,784 (83.17 MB)

Annexe 5 : Les matrices de confusion optimisées pour les différentes combinaisons pour le modèle XGB



## BIBLIOGRAPHIE

- [1] A. M. Ismael et A. Şengür, « Deep learning approaches for COVID-19 detection based on chest X-ray images », *Expert Systems with Applications*, vol. 164, p. 114054, févr. 2021, doi: 10.1016/j.eswa.2020.114054.
- [2] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, et A. A. Efros, « CNN-generated images are surprisingly easy to spot... for now », 2019, *arXiv*. doi: 10.48550/ARXIV.1912.11035.
- [3] M. Rahimzadeh et A. Attar, « A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2 », *Informatics in Medicine Unlocked*, vol. 19, p. 100360, 2020, doi: 10.1016/j.imu.2020.100360.
- [4] T. Rahman, M. Chowdhury, et A. Khandakar, « Covid-19 radiography database ». 2021. [En ligne]. Disponible sur: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data>
- [5] C. R. Harris *et al.*, « Array programming with NumPy », *Nature*, vol. 585, n° 7825, p. 357-362, sept. 2020, doi: 10.1038/s41586-020-2649-2.
- [6] W. McKinney, « Data Structures for Statistical Computing in Python », présenté à Python in Science Conference, Austin, Texas, 2010, p. 56-61. doi: 10.25080/Majora-92bf1922-00a.
- [7] T. Chen et C. Guestrin, « XGBoost: A Scalable Tree Boosting System », in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, août 2016, p. 785-794. doi: 10.1145/2939672.2939785.
- [8] G. Huang, Z. Liu, L. van der Maaten, et K. Q. Weinberger, « Densely Connected Convolutional Networks », 2016, *arXiv*. doi: 10.48550/ARXIV.1608.06993.
- [9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, et Z. Wojna, « Rethinking the Inception Architecture for Computer Vision », 2015, *arXiv*. doi: 10.48550/ARXIV.1512.00567.
- [10] A. Alqahtani *et al.*, « A Transfer Learning Based Approach for COVID-19 Detection Using Inception-v4 Model », *Intelligent Automation & Soft Computing*, vol. 35, n° 2, p. 1721-1736, 2023, doi: 10.32604/iasc.2023.025597.