**Bootcamp Data Science**

**Cohorte Octobre 2023**

**Projet PulmoSCAN**

Détection du COVID-19 et autres maladies pulmonaires à partir de radiographies du thorax, grâce à l’apprentissage profond (*deep learning*)

**Rapport final**

**Version du 30/12/2023**

Equipe projet : Steve Costalat, Thibaut Gazagnes, Nicolas Gorgol

Mentor projet : Gaël Penessot

Table des matières

[1. Introduction : contexte, enjeux et objectifs du projet 2](#_Toc154825490)

[a. Contexte et problématique 2](#_Toc154825491)

[b. Objectifs du projet 4](#_Toc154825492)

[2. Exploration et préparation des données 5](#_Toc154825493)

[a. Description du jeu de données 5](#_Toc154825494)

[b. Premières analyses 5](#_Toc154825495)

[c. Equilibre des classes 8](#_Toc154825496)

[d. Pré-traitement des données pour la phase de modélisation 12](#_Toc154825497)

[3. Modélisation 14](#_Toc154825498)

[a. Classification du problème et choix de la métrique 14](#_Toc154825499)

[b. Démarche d’expérimentation et résultats 15](#_Toc154825500)

[c. Modèle LeNet-5 17](#_Toc154825501)

[d. Optimisation des hyperparamètres du modèle LeNet 18](#_Toc154825502)

[e. Approche d’apprentissage par transfert à partir de modèles pré-entraînés 19](#_Toc154825503)

[f. Interprétation des résultats 1](#_Toc154825504)

[4. Approfondissements 7](#_Toc154825505)

[a. Expérimentation 1 : Approche réduite sur la base des distributions d’intensité 7](#_Toc154825506)

[b. Expérimentation 2 : Augmentation de données (pré-traitement) 10](#_Toc154825507)

[c. Expérimentation 3 : Entraînement sur un jeu de données 3 fois plus important 10](#_Toc154825508)

[d. Expérimentation 4 : Pré-traitement supplémentaire des images en entrée 10](#_Toc154825509)

[e. Expérimentation 5 : Classification 3 classes 11](#_Toc154825510)

[f. Expérimentation 5 : apprentissage sur les données hors zone pulmonaire 11](#_Toc154825511)

[5. Conclusion et perspectives 14](#_Toc154825512)

[a. Pertinence de la solution 14](#_Toc154825513)

[b. Limites identifiées 14](#_Toc154825514)

[c. Pistes d’amélioration et solutions alternatives à envisager 15](#_Toc154825515)

[d. Potentiel de généralisation pour un usage réel 15](#_Toc154825516)

[e. L’avis d’un radiologue professionnel 15](#_Toc154825517)

[f. Bénéfice du projet pour l’équipe 15](#_Toc154825518)

[g. Difficultés rencontrées 16](#_Toc154825519)

[6. Annexes 17](#_Toc154825520)

[a. Bibliographie 17](#_Toc154825521)

# Introduction : contexte, enjeux et objectifs du projet

## Contexte et problématique

L’imagerie médicale est un outil de diagnostic majeur dans la plupart des champs de la médecine : radiographie, échographie, IRM, scanner… Ces outils produisent une quantité importante de données sous forme d’images, que les professionnels de santé interprètent à l'œil ou aidés de logiciels. Le développement de la Computer Vision a permis de faire grandement avancer l’analyse des images médicales afin d’améliorer les diagnostics et de faire gagner du temps aux praticiens.

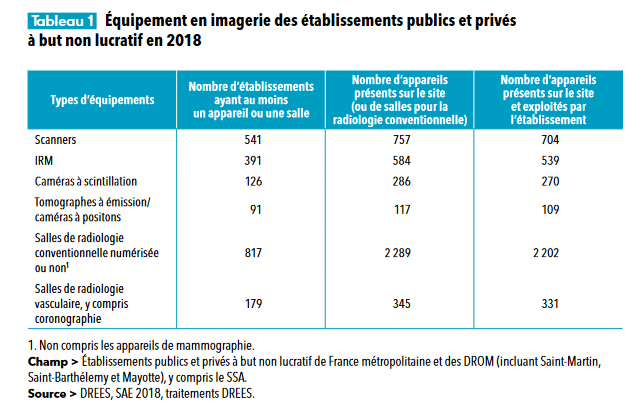
La pandémie de COVID-19 a surpris le monde entier par sa vitesse de propagation et sa viralité. Les professionnels de santé ont dû faire face à un flux important de malades, et au besoin de diagnostiquer et d’orienter rapidement les cas positifs vers les bons traitements. En France en particulier, la tension sur les effectifs a été particulièrement forte, dans un contexte préexistant de pénurie de médecins généralistes et dans certaines spécialités.

L’analyse des données d’imagerie médicale, afin d’établir un diagnostic rapide et fiable, a été explorée par plusieurs équipes de chercheurs autour du monde (comme le *Vision and Image Processing Research Group* de l’Université de Waterloo au Canada).

La radiographie, le scanner et l’IRM sont trois techniques d’imagerie médicale dites “non-invasives” et indolores, car elles ne nécessitent pas d’opération. La radiographie et le scanner utilisent des rayons-X, tandis que l’IRM utilise des ondes électromagnétiques.

Bien que le scanner ait généralement une meilleure sensibilité de détection, la radiographie est plus couramment utilisée dans la pratique clinique en raison des avantages qu'elle présente, notamment son faible coût, sa faible dose de rayonnement, sa facilité d'utilisation et sa grande accessibilité dans les hôpitaux publics autour du monde.[[1]](#footnote-1)

D’après une étude de la DREES[[2]](#footnote-2) (Direction de la Recherche, des Études, de l'Évaluation et des Statistiques du Ministère de la Santé et de la Prévention), la France comptait en 2018 **2289 salles de radiologie conventionnelle** en établissements publics ou privé non lucratif**, contre 757 scanners** et **584 IRM.**

****

La radiographie consiste à utiliser des rayons X qui traversent le corps humain et sont capturés sur un film électronique. Les rayons X sont plus ou moins absorbés en fonction de la densité des tissus, et les images permettent au radiologue de visualiser par contraste l’intérieur du corps humain.

Les images numérisées issues d’appareils de radiographies peuvent être exploitées grâce aux techniques de Computer Vision. Développer un modèle de classification fiable permettrait d’utiliser cette technique pour faire gagner du temps dans le diagnostic du Covid.

Aucun de nous trois n’est un professionnel de santé, ni n’avons de formation à l’interprétation de radiologies. Nous allons donc nous en remettre à un ou plusieurs modèles d’apprentissage profond (« deep learning ») pour essayer de distinguer des traits caractéristiques dans les données, et d’établir une classification fiable.

Nous avons choisi ce projet pour le sens qu’il porte : améliorer les techniques de diagnostic afin d'accélérer la prise en charge des patients. De plus, plusieurs membres de l’équipe ont une expérience passée dans le domaine de la santé.

## Objectifs du projet

Le projet vise à développer un modèle d’apprentissage automatique visant à classer **de manière fiable** **une radiographie pulmonaire s**elon l’un des quatre diagnostics suivants :

1. Patient **sain** (“Normal”)
2. Patient atteint du **Covid-19**
3. Patient atteint de **pneumopathie virale**
4. Patient diagnostiqué avec une **opacité pulmonaire** (qui regroupe plusieurs types d'infections pulmonaires non-COVID).

L’équipe projet ne dispose pas d’expérience préalable en imagerie médicale. Toutefois, Steve Costalat a travaillé 20 ans en environnements cliniques (bloc opératoires, service d'urgence et service de réanimation) et dispose d’une expertise des problématiques de protocoles de diagnostic en environnement critique.

L'objectif pour l'équipe est avant tout d'explorer les différentes approches et limites qu'offre le 'Deep Learning' pour l'analyse de radiographie pulmonaires dans le diagnostic de pneumopathie.

# Exploration et préparation des données

|  |
| --- |
| **En synthèse,**   * Le jeu de données est constitué de **21 165 radiographies** de la zone pulmonaire disponibles sous formes d’images au format PNG et de taille harmonisée 299\*299. Toutes les images sont **labellisées** (pré-classées par diagnostic). * **La surface utile des images est uniquement de 27 % en moyenne**. Nous travaillerons donc sur les images masquées, puisque les masques sont fournis avec le jeu de données. (Cette approche est recommandée dans plusieurs articles scientifiques consultés, par exemple [celui-ci](https://www.sciencedirect.com/science/article/pii/S0031320322003077), pour s’assurer que le modèle extrait bien des caractéristiques distinctives à l’intérieur de la zone pulmonaire). * Le jeu de données est déséquilibré : la classe « Normal » est sur-représentée. |

## Description du jeu de données

Le jeu de données utilisé est une banque d’images de radiographies pulmonaires, consolidée par une équipe de chercheurs de l'université du Qatar, à Doha, et de l'université de Dhaka (Bangladesh). Cette banque d’image est en accès libre sur Kaggle[[3]](#footnote-3).

Elle a été constituée à partir de bases existantes (comme l’*Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE [1], Novel Corona Virus 2019 Dataset developed by Joseph Paul Cohen and Paul Morrison, and Lan Dao in GitHub)* et de plusieurs dizaines de publications scientifiques.

Chaque image est fournie avec un masque de segmentation, qui permet d’isoler la zone pulmonaire et de masquer les parties de l’image radio non pertinentes pour le diagnostic.

Cette base de données est à usage académique et non commercial.

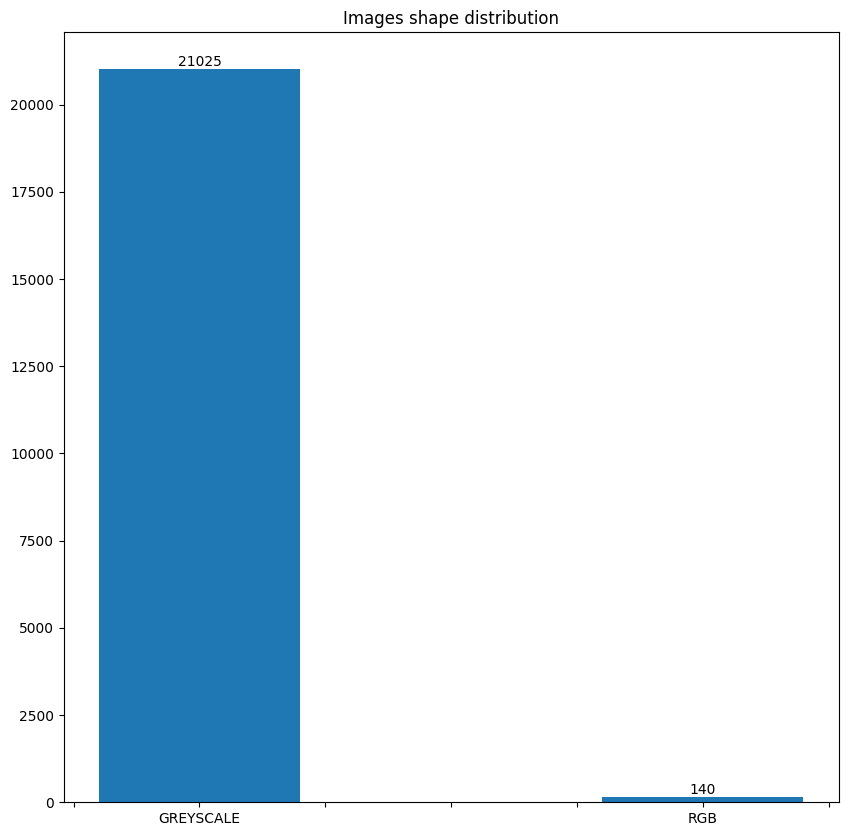
Une première analyse des images et de leurs attributs montre que le jeu de données semble de qualité homogène (tailles, résolutions, orientations, …).

## Premières analyses

Le set contient **21 165 images** de radiographies pulmonaires avec leurs masques associés. Les images de radios pulmonaires sont au format PNG, de taille 299\*299 pixels en uint8. Elles sont en mode « niveaux de gris » pour plus de 99% d’entre elles, et « RGB » pour 140 images. Après vérification ces images en RGB ont 3 trois canaux identiques, nous les convertirons donc en niveaux de gris dans le pré-traitement

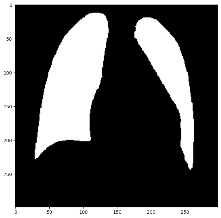


Affichage de 3 images au hasard dans le jeu de données



Répartition des images par mode (RBG/Niveaux de gris) dans le jeu de données (images radio)

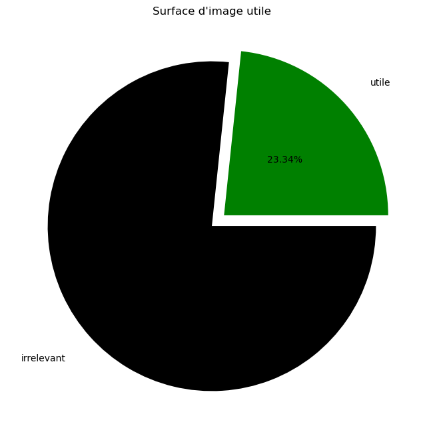
Les masques fournis sont au format PNG également mais de dimensions 256\*256. Ils permettent d’isoler la surface pulmonaire sur les images. Ils vont nous permettre de limiter les données transmises au modèle après le pré-traitement. Comme leur taille est légèrement différente des images, nous les redimensionnerons et les convertirons en Niveaux de gris dans le pré-traitement



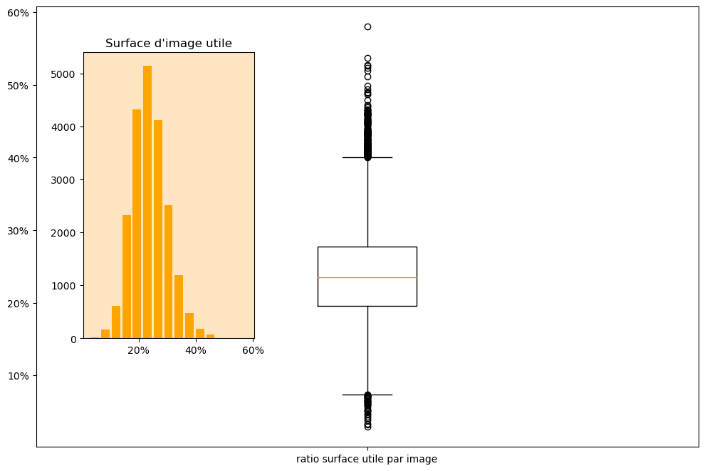
Exemple de masque

### Surface d’image utile

Grâce aux masques, nous avons pu calculer la **surface moyenne d’image utile** pour le modèle (“utile” étant la partie de l’image qui reste visible après application du masque). Celle-ci s’élève à 23%, ce qui nous incite à utiliser les masques afin de réduire le volume d’information inutile en entrée du modèle.



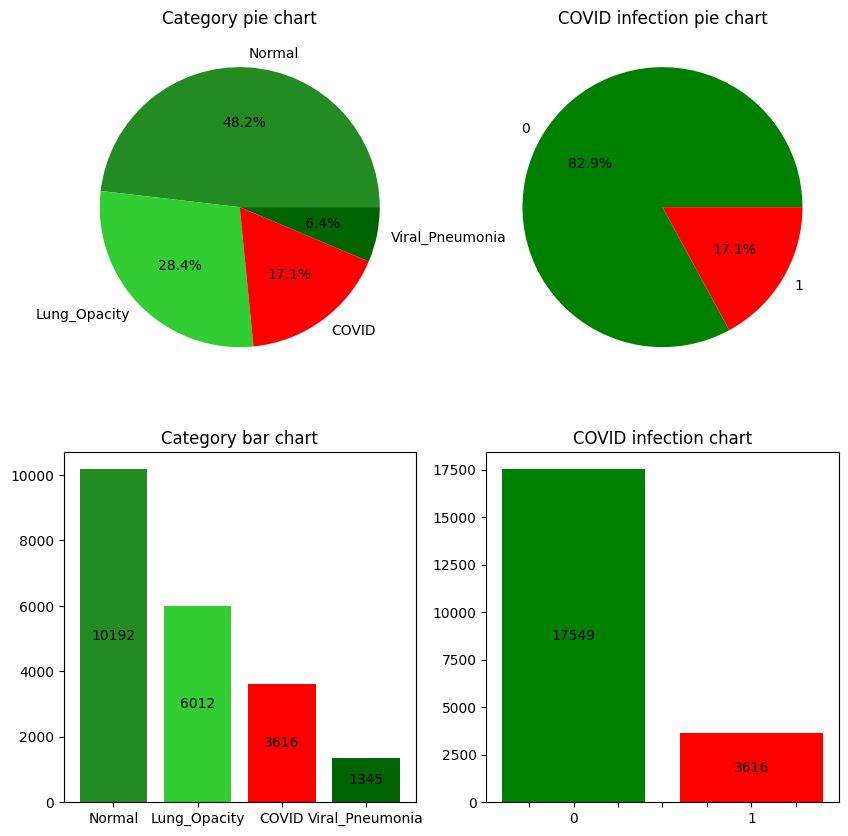
Nous avons également affiché la distribution de la surface d’image utile sur l’ensemble du jeu de données :



## Equilibre des classes

Le jeu de données n’est pas équilibré : la catégorie “Normal” (porteur sain) représente 48% des images, l’opacité pulmonaire 28%, le Covid 17%, et la pneumonie virale 6%.

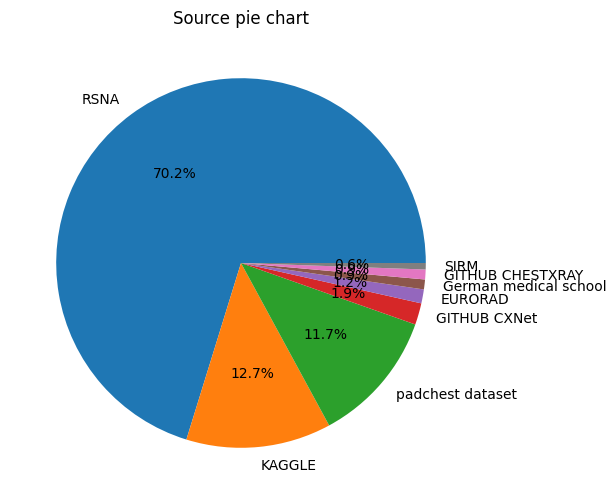
La classe COVID qui est l’objet de la détection représente donc environ ⅙ des images du set (3 616 sur 21 165). Il faudra donc rééquilibrer les données utilisées pour l’entraînement des modèles afin de ne pas biaiser sa précision.



Les images sont accompagnées d’un fichier “metadata.csv” pour chaque set (Normal, Covid, etc) qui résume le nom du fichier, ses dimensions, et l’url du jeu de données source. L’analyse des propriétés des images sur Python nous a permis d’identifier des erreurs dans le fichier metadata (par exemple, format 256\*256 alors que les images sont en 299\*299).

### Sources des images

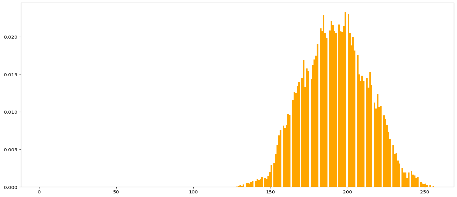
La répartition des sources montre que les images sont issues de 8 sources, et que plus de 70% des images proviennent d’un jeu de données préexistant utilisé pour la détection de pneumonies sur Kaggle.



Répartition des images du jeu de données par source

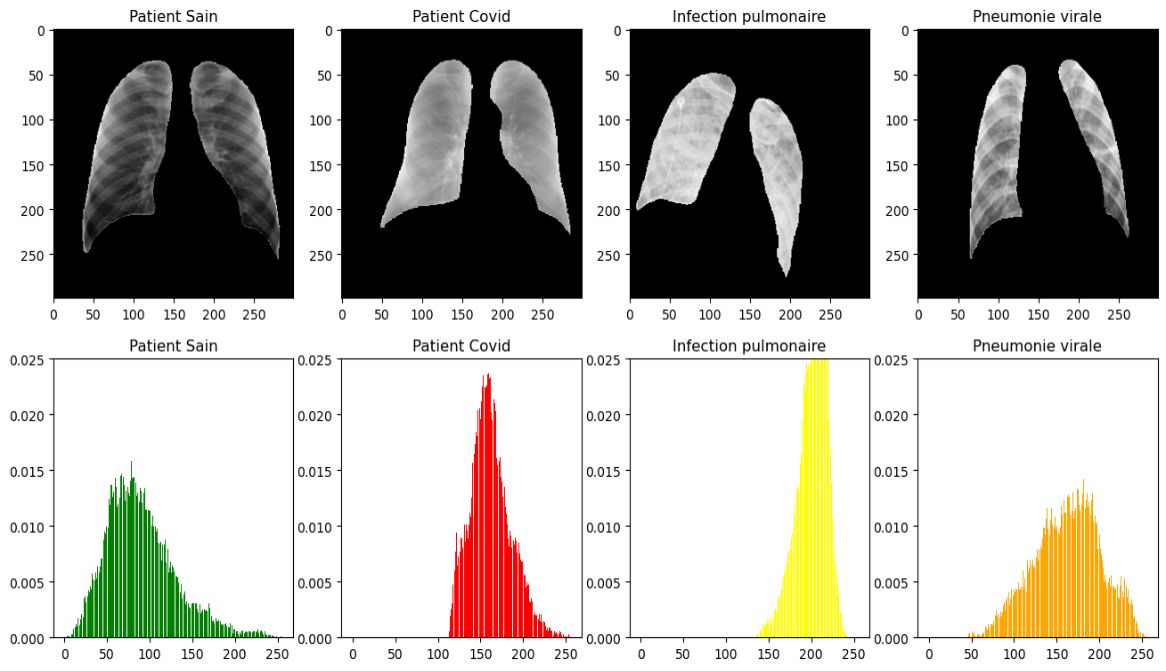
### Histogrammes de distribution des pixels

Nous avons également calculé et affiché l’histogramme de la distribution des pixels de chaque image en niveaux de gris :



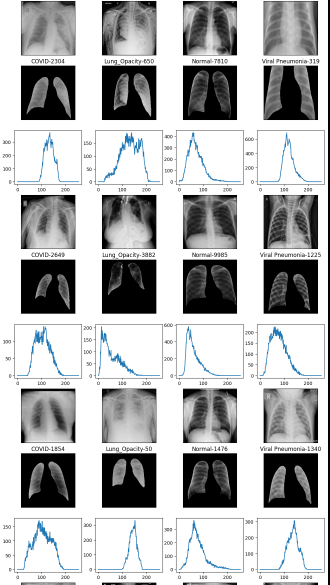
Exemple d’histogramme de répartition des pixels par intensité (entre 0 et 255) pour une image aléatoire

Nous avons également affiché aléatoirement une ou plusieurs images par set, avec sa distribution associée. Cela montre des différences entre les distributions qui pourraient être liées à des différences d’intensité dans les radios en fonction de la maladie. Nous ne pouvons pas vérifier cette hypothèse à ce stade et devrons laisser les modèles d’apprentissage profond déterminer les attributs qui permettent de catégoriser les radios.



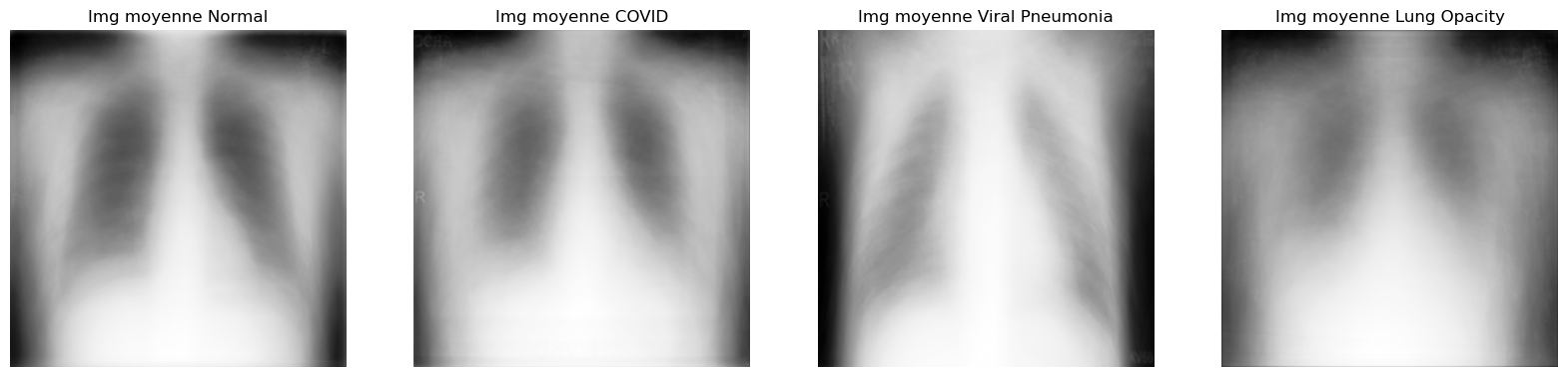
● Une réduction de dimension de 256x256 en un simple vecteur 1x256 semble possible en remplaçant l'image par la distribution d'intensité.

Autre exemple avec une image de chaque catégorie, la même image à laquelle on a appliqué le masque, avec leurs histogrammes :



### Images moyennes et statistiques d’intensité

Nous avons également pu calculer et afficher les “images moyennes” sur un sous-set de 100 images par catégorie (pour accélérer le traitement). Cela nous a permis de repérer une opacité plus marquée pour les diagnostics positifs (Covid, pneumonie virale, opacité pulmonaire) que pour les patients sains. Les différences entre les différentes maladies pulmonaires sont moins évidentes à repérer à l'œil nu.



Le traitement des images masquées grâce à leur conversion en array de pixels (grâce aux modules OpenCV, PIL et numpy) nous a permis de calculer plusieurs statistiques sur les images du jeu de données. On constate que la densité moyenne des images COVID est plus élevée que les autres catégories, ce qui laisse penser qu’elles sont en moyenne plus “claires” que celles des autres catégories (au sens où elles contiennent en moyenne plus de pixels plus clairs que les autres catégories).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Catégories | Moyenne  Densités moyennes | Écart-type  Densités moyennes | Moyenne  Écart-type des Densités | Écart-type  Écart-type des Densité |
| Covid | 139.4 | 25.0 | 54.6 | 14.8 |
| Opacités pulmonaires | 125.9 | 23.6 | 57.3 | 9.8 |
| Normal | 129.3 | 22.4 | 61.6 | 9.5 |
| Pneumonie virale | 125.3 | 19.0 | 58.7 | 10.8 |

## Pré-traitement des données pour la phase de modélisation

L’analyse exploratoire a permis d’identifier :

* Des différences de format et taille dans les images
* Un déséquilibre des classes et en particulier une sous-représentation de la classe COVID qui est l’objectif de la détection
* Une opportunité de réduire les données non pertinentes en entrée du modèle grâce aux masques.

Nous avons donc retenu les étapes suivantes pour créer le jeu de données pré-processé à injecter dans les modèles :

1. **Conversion**: convertir toutes les images en Niveaux de gris
2. **Redimensionnement** : Redimensionner les images en 256\*256 pour correspondre à la taille des images radio. Nous avons hésité entre garder les images en 299\*299 ou les réduire, nous pourrons revenir sur ce choix en fonction des résultats des premiers modèles.
3. **Réduction des informations inutiles :** Appliquer les masques aux images
4. **Equilibrage :** Équilibrer le jeu de données grâce à la méthode du sous-échantillonnage sans remise, en s’alignant sur la fréquence de la classe minoritaire (Viral Pneumonia ici).
5. **Augmentation des données** : Il n'est pas certain que d'augmenter nos données soit nécessaire à ce stade. L'expérimentation des premiers modèles nous indiquera s'il est utile de le faire pour la suite.

Lors de l'analyse des données suite à l'observation de nombreuses distributions d'intensité des images masquées, l'hypothèse a été émise dans le groupe que les formes des distributions semblaient similaires selon le diagnostic final. Si tel est en effet le cas cela représenterait un moyen de réduire considérablement la taille du modèle en utilisant uniquement cette distribution en entrée. Cette option pourra être investiguée par la suite.

Enfin, il nous faudra séparer le jeu d’images en un jeu d’entraînement et un jeu de test, en veillant à l’équilibre des classes dans la répartition aléatoire.

|  |  |  |
| --- | --- | --- |
| **Catégorie** | **Nombre d’images dans le jeu de données initial** | **Nombre d’images après ré-échantillonnage** |
| **Normal** | 10192 | 1345 |
| **Autres ('Lung Opacity')** | 6012 | 1345 |
| **Covid** | 3616 | 1345 |
| **Pneumonie Virale** | 1345 | 1345 |

# Modélisation

## Classification du problème et choix de la métrique

Il s’agit ici d’un problème de classification multi-classe, puisque nous cherchons à déterminer un diagnostic à partir d’images radio, parmi les 4 possibilités suivantes :

Une image contenant capture d’écran

Description générée automatiquement

La tâche à réaliser s’apparente à de la reconnaissance d’image (« computer vision »), avec un enjeu d’extraction des bons attributs permettant de caractériser chaque maladie à partir d’une seule image radio. C’est une tâche complexe même pour un ou une radiologie expérimentée : l’interprétation seule de radios du thorax présente une sensibilité (rappel) de 69%[[4]](#footnote-4), donc un taux fort de faux négatifs. En comparaison, le test PCR affiche une sensibilité de 91%.

C’est pourquoi la plupart des organisations professionnelles de radiologie ne recommandent pas l’usage de la radio ou du scanner pour détecter le COVID.

Pour ce problème, nous avons choisi de comparer les modèles sur la base de la métrique de justesse sur le jeu de validation (« validation accuracy »).

En effet, nous ne cherchons pas seulement à identifier le COVID, mais également à détecter les 3 autres diagnostics possibles (Normal, Pneumopathie virale, Opacité pulmonaire).

Nous avons équilibré les classes dans le jeu de données d’entraînement, il y a donc moins de risques que la mesure de justesse cache une sur-représentation de la classe « Normale », comme il peut se produire dans un problème de détection d’anomalie.

Une fois les modèles les plus pertinents retenus, nous analyserons la performance sur un jeu de test ainsi que la matrice de confusion et les métriques complètes (f1-score, précision, rappel) pour s’assurer qu’il n’y a pas d’écart majeur de performance entre les classes.

## Démarche d’expérimentation et résultats

Voici en synthèse les expérimentations successives que nous avons menées :

1. **Implémentation d’un modèle de base : LeNet-5 :**
2. **Optimisation des hyperparamètres du modèle LeNet**
3. **Implémentation d’une approche d’apprentissage par transfert à partir de modèles pré-entraînés**

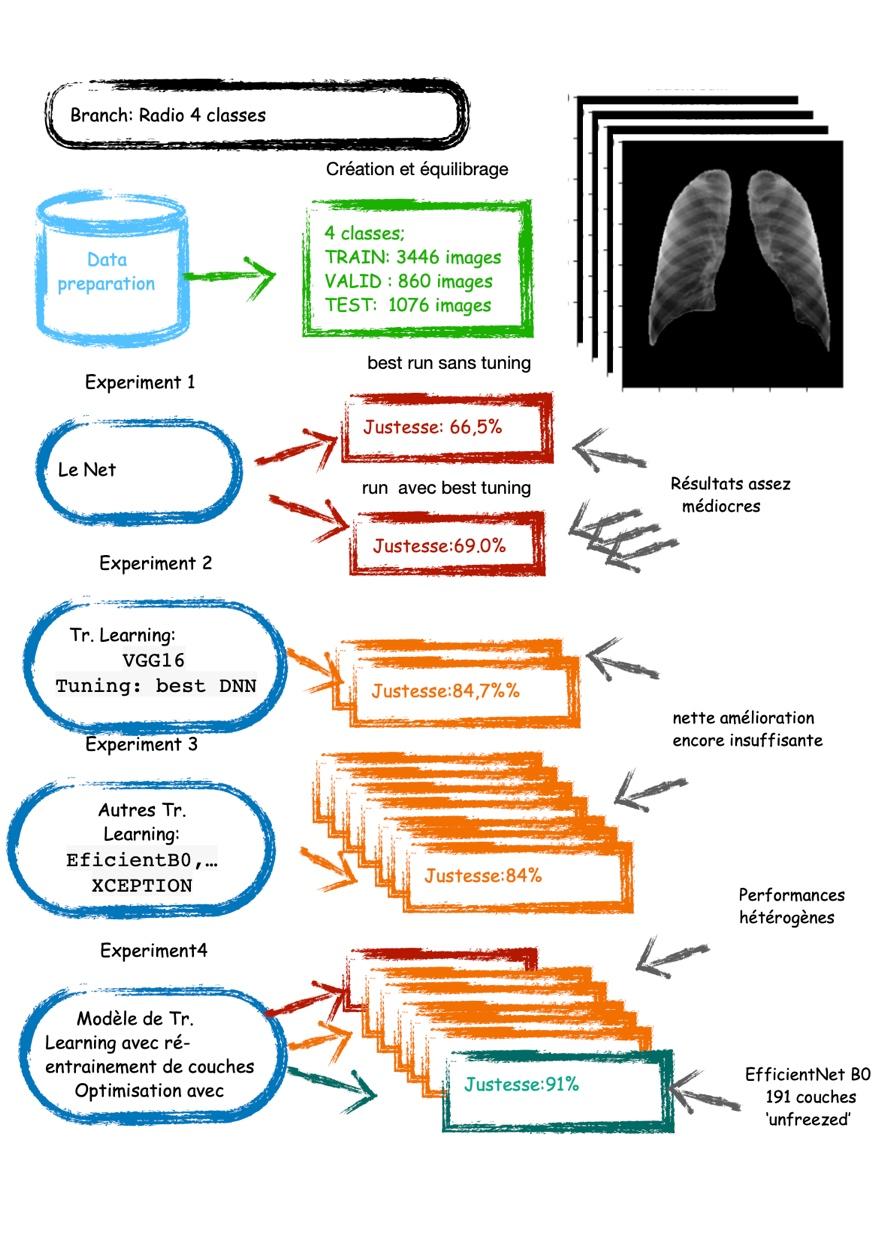


Schéma de synthèse de la méthode d’expérimentation

Toutes ces expérimentations, à l’exception de la 1.a visant à faire varier les jeux de données, ont été réalisées sur la base d’un **même jeu de données d’entraînement pré-traité comportant 4304 images (= 1076 par classe), et d’une structure de notebook commune.**

Lors de l’import dans Keras, 20% des images sont mises de côté aléatoirement pour la « validation » et le calcul de la « validation accuracy ». Le jeu de test utilisé pour l’évaluation finale est également le même pour tous les modèles et correspond à 20% du jeu de données initial équilibré, soit **1076** images (269 par classe).

**Synthèse des résultats obtenus (classés par modèle) :**

Les meilleurs résultats obtenus sont surlignés en couleur.

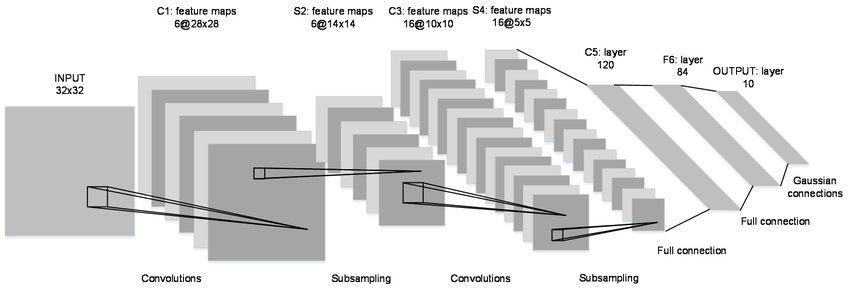
|  |  |  |  |
| --- | --- | --- | --- |
| **Modèle testé** | **N° d’exp.** | **Caractéristiques de l’expérimentation** | **Métrique de justesse sur le jeu de validation** |
| **LeNet-5** | #1.1 | Sans optimisation d’hyperparamètres (HP) | **67 %** |
| #1.2 | Avec optimisation des hyperparamètres | **69%** |
| **VGG16** | #2.1 | Apprentissage par transfert sans dégel de couches – sans optimisation des HP | **85%** |
| #2.2 | Apprentissage par transfert sans dégel de couches – avec optimisation des HP | **86%** |
| #4.1 | Dégel et réapprentissage : 4 couches | **71%** |
| #4.2 | Dégel et réapprentissage : 8 couches | **74%** |
| #4.3 | Dégel et réapprentissage : 12 couches | **25%** |
| **VGG19** | #3.1 | Apprentissage par transfert sans dégel de couches | **85%** |
| #4.4 | Dégel et réapprentissage de la moitié des couches | **89%** |
| #4.5 | Dégel et réapprentissage de 100% des couches | **91%** |
| **XCEPTION** | #3.2 | Apprentissage par transfert sans dégel de couches | **78%** |
| #4.6 | Dégel et réapprentissage : 27 couches | **86%** |
| #4.7 | Dégel et réapprentissage : 67 couches | **87%** |
| #4.8 | Dégel et réapprentissage : 97 couches | **72%** |
| **ResNet50** | #3.3 | Apprentissage par transfert sans dégel de couches | **80%** |
| #4.9 | Dégel et réapprentissage : 32 couches | **74%** |
| **ResNet101** | #3.4 | Apprentissage par transfert sans dégel de couches | **82%** |
| **ResNet50V2** | #3.5 | Apprentissage par transfert sans dégel de couches | **81%** |
| **ResNet101v2** | #3.6 | Apprentissage par transfert sans dégel de couches | **81%** |
| **ResNet152V2** | #5.1 | Apprentissage par transfert sans dégel de couches | **82%** |
| #5.2 | Dégel et réentraînement de toutes les couches | **87%** |
| **DenseNet121** | #3.7 | Apprentissage par transfert sans dégel de couches | **82%** |
| **DenseNet201** | #3.8 | Apprentissage par transfert sans dégel de couches | **82%** |
| **MobileNet** | #3.9 | Apprentissage par transfert sans dégel de couches | **83%** |
| #4.10 | Dégel et réapprentissage : 19 couches | **80%** |
| #4.11 | Dégel et réapprentissage : 50 couches | **78%** |
| **MobileNetV2** | #3.10 | Apprentissage par transfert sans dégel de couches | **82%** |
| **EfficientNetB0** | #3.11 | Apprentissage par transfert sans dégel de couches | **84%** |
| #4.12 | Dégel et réapprentissage : 75 couches | **85%** |
| #4.13 | Dégel et réapprentissage : 118 couches | **86%** |
| #4.14 | Dégel et réapprentissage : 162 couches | **86%** |
| #4.15 | Dégel et réapprentissage : 191 couches | **91%** |
| **EfficientNetB7** | #3.12 | Apprentissage par transfert sans dégel de couches | **83%** |

Détail de quelques-unes des expérimentations effectuées

## Modèle LeNet-5

### Description du modèle

Le modèle LeNet-5, développé par Yann LeCun et ses collègues, est un réseau de neurones convolutionnel (CNN) classique. Nous avons utilisé une architecture simple avec des couches de convolution, de sous-échantillonnage, et de couches denses. Le modèle LeNet-5 est connu pour son architecture relativement simple, mais il est efficace pour des tâches de classification d'images simples.



**Représentation du modèle LeNet5 sur la base d’un input 32\*32 -** [**Source**](https://www.researchgate.net/figure/The-LeNet-5-Architecture-a-convolutional-neural-network_fig4_321586653)

### Entraînement du modèle LeNet-5 avec différentes tailles de jeu de données

Cette expérimentation visait à étudier la sensibilité de la performance du modèle LeNet-5 par rapport à la taille du jeu de données. Des jeux de données de tailles différentes ont été utilisés, allant de petits ensembles à des ensembles de données plus vastes.

En entraînant le modèle LeNet tuné sur des jeu de données de taille 1000, 2000, 3000, 4000, 4576 images, nous avons constaté que plus le jeu de données d’entraînement est important, meilleure est l’accuracy ; en revanche le temps de computation augmente fortement. Cela pourrait justifier le recours à l’augmentation de données afin d’augmenter la performance et réduire l’overfitting. Afin de prioriser l’expérimentation de modèles plus complexes, nous n’avons pas, à ce stade, appliqué d’augmentation d’image.

Une image contenant texte, ligne, capture d’écran, Tracé

Description générée automatiquement

## Optimisation des hyperparamètres du modèle LeNet

Nous avons effectué un tuning des hyperparamètres pour optimiser les performances du modèle LeNet-5. En utilisant la fonction Keras Tuner, nous avons fait varier les paramètres suivants :

* Nombre de filtres de la première couche de convolution (entre 16 et 128 par pas de 32)
* Nombre de filtres de la deuxième couche de convolution (entre 16 et 128 par pas de 32)
* Intensité du dropout à la sortie des couches de convolution (entre 0 et 0.5 par pas de 0.1)
* Nombre de neurones de la première couche dense (parmi 32, 64, 128, 256)
* Intensité du dropout à la sortie de la couche dense (entre 0 et 0.5 par pas de 0.1)
* Pas d’apprentissage (« learning rate ») pour l’optimiser Adam (parmi 0.1, 0.01, 0.001, 0.0001, 1e-05)

Dans le même esprit nous avons effectué un keras tuner sur l'architecture DNN de classification optimale. Nous avons fait varier :

* Nombre de couches de DNN précédant le dernière couche (de 0 à 10)
* Nombre de neurones pour chacune des couches (parmi 16, 32, 64,128, 256, 512, 1024, et 2076)
* La présence ou l'absence d'une couche Dropout entre chaque couche
* La fonction d'activation de chaque couche (parmi 'relu' ou 'tanh')

Le modèle de classificateur DNN final obtenue :

|  |  |  |  |
| --- | --- | --- | --- |
| Couches du classificateurs | Nombre de neurones | fonction d'activation | drop out ratio at 'output' |
| Dense 1 | 128 | 'relu' | 0% |
| Dense 2 | 32 | 'relu' | 20% |
| Dense 3 | 16 | 'tanh' | 0% |
| Dense 'Ouput' | 4 | 'softmax' | NA |

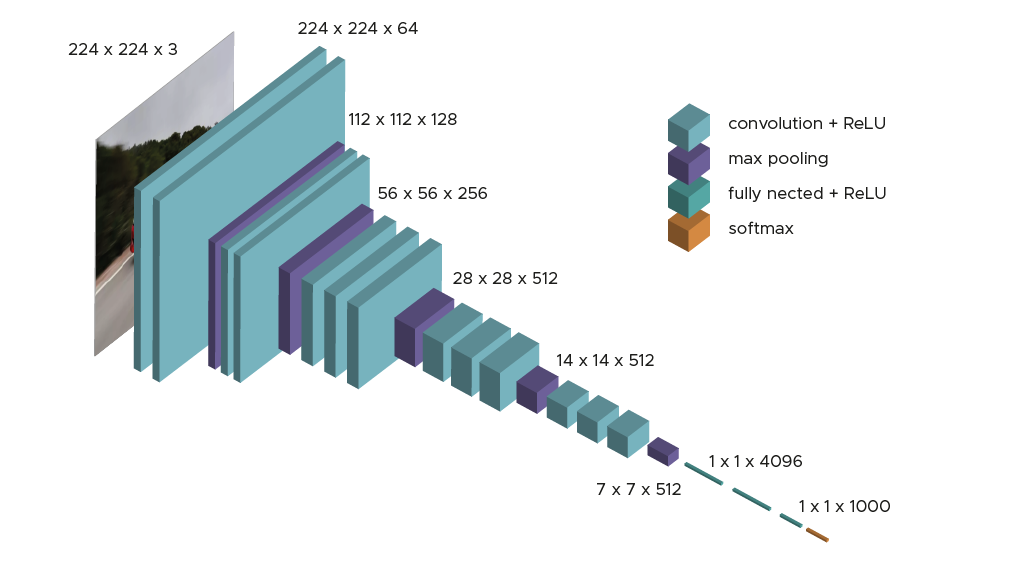
## Approche d’apprentissage par transfert à partir de modèles pré-entraînés

### Implémentation d’un modèle d’apprentissage par transfert utilisant le modèle VGG16

L’apprentissage par transfert a été appliqué en utilisant le modèle pré-entraîné VGG16. Nous avons gelé les couches convolutionnelles préexistantes et ajouté des couches denses spécifiques à notre tâche de classification.

VGG16 est un modèle de réseau de neurones convolutionnel qui a été développé par le Visual Graphics Group (VGG) à l'Université d'Oxford. Il a été créé pour participer au concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2014, où il a obtenu de très bonnes performances.

Le modèle VGG16 est caractérisé par sa profondeur, composée de 16 couches de convolution et de sous-échantillonnage (pooling) suivi de trois couches denses. Les couches de convolution sont toutes configurées avec des filtres de petite taille (3x3) et un pas (stride) de 1, ce qui donne une architecture très uniforme.



**Schéma de la structure de VGG16 – Source** [**Datascientest**](https://datascientest.com/quest-ce-que-le-modele-vgg)

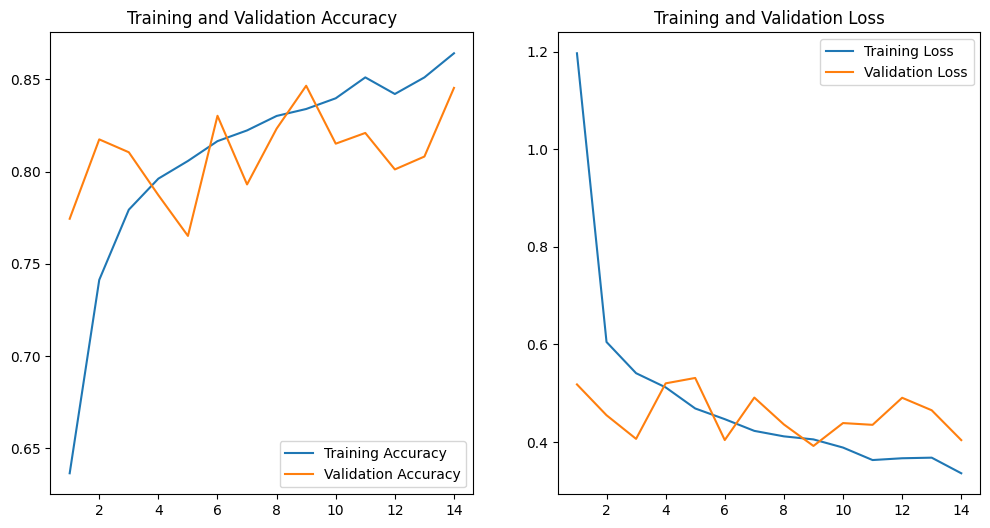
L’apprentissage par transfert permet de réutiliser un modèle pré-entraîné pour une tâche similaire. Dans le cas de VGG16, le modèle a été pré-entraîné sur la base de données ImageNet pour classifier une large variété d'images. Même si les images n’ont pas de rapport avec les images radio de notre jeu de données, les capacités d’extraction d’attributs à partir des images font de VGG16 un candidat intéressant.

Dans un premier temps, nous avons entraîné un modèle composé :

* D’une couche d’entrée permettant d’injecter des images au format (256, 256, 3). Le modèle VGG16 est paramétré pour des images au mode RGB.
* Du modèle VGG16 importé sans les couches denses de classification
* D’une couche de « GlobalAveragePooling » pour convertir la sortie du modèle VGG16
* De plusieurs couches denses pour la classification en 4 classes.

La fonction de pré-traitement fournie par la bibliothèque VGG16 de Keras a été appliquée au jeu de données d’entraînement, de validation et de test (centrage des valeurs de pixels notamment), en amont du modèle.

Les couches du modèles VGG16 ont été « gelées » pour pouvoir utiliser les poids liés au pré-entraînement sur le jeu « ImageNet ». Le modèle a été compilé avec l’optimiseur « Adam », la fonction de perte « sparse categorical crossentropy » adaptée à la classification multi-classes de données labellisées, et la mesure de justesse (« accuracy ») sur le jeu d’entraînement et de validation.

Courbes de justesse et perte pour le modèle d’apprentissage par transfert utilisant VGG16   
(sans finetuning)

### Optimisation des paramètres du classifieur de sortie du modèle VGG 16

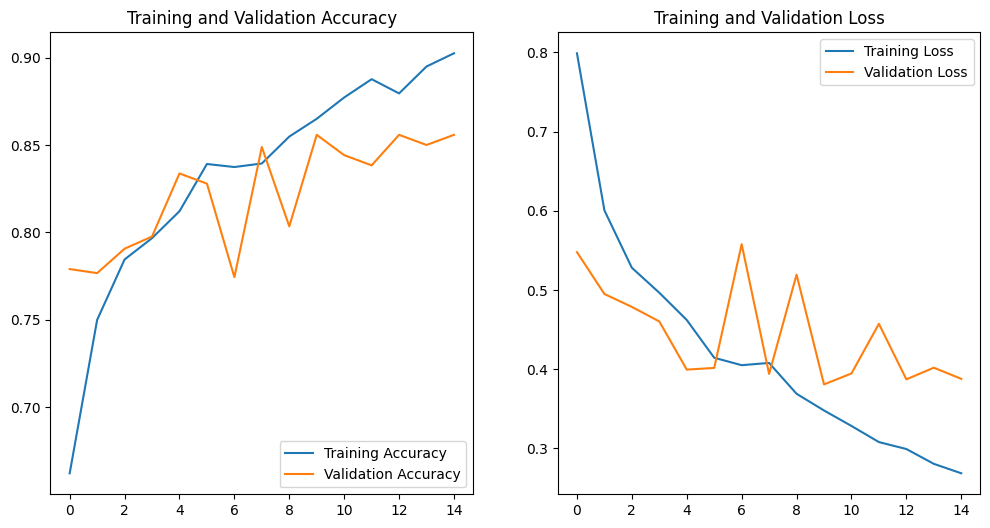
Nous avons effectué un tuning des hyperparamètres pour optimiser les performances de ce modèle basé sur VGG16. En utilisant la fonction de tuner « Hyperband » de la bibliothèque Keras Tuner.

|  |  |  |
| --- | --- | --- |
| **Hyperparamètre** | **Espace d’optimisation** | **Meilleur modèle retenu** |
| Intensité du dropout à la sortie du modèle VGG16 | Entre 0 et 0.5 par pas de 0.1 | **0** |
| Nombre de neurones de la première couche dense | 0, 32, 64, 128, 256, 512 | **128** |
| Nombre de neurones de la deuxième couche dense | 0, 32, 64, 128, 256, 512 | **32** |
| Intensité du dropout à la sortie de la couche dense | entre 0 et 0.5 par pas de 0.1 | **0.2** |
| Pas d’apprentissage (« learning rate ») pour l’optimiser Adam | 0.1, 0.01, 0.001, 0.0001, 1e-05 | **0.001** |

Les fonctions d’activation ont également fait l’objet de tests manuels (entre « relu » et « tanh »), mais n’affectent pas la performance du modèle de manière significative.

Le meilleur modèle a ensuite été implémenté et entraîné sur le jeu de 4503 images, et a atteint une performance de 86% sur les données de validation. Evalué sur le jeu de test, il atteint 83%. On remarque sur les courbes suivantes que la performance sur le jeu d’entraînement devient rapidement supérieure à celle sur le jeu de validation : le modèle sur-adapte au bout de quelques périodes.

Pour éviter cela, nous avons ajouté un paramètre « callback » lors de l’entraînement du modèle, permettant d’arrêter l’entraînement dès lors que le score de perte ne s’améliore plus, avec une latence de 5 périodes.



Courbes d’accuracy et loss pour le modèle d’apprentissage par transfert utilisant VGG16   
(avec optimisation des hyperparamètres et arrêt anticipé (« callback »).

Le rapport de classification calculé sur le jeu de données de test apporte des précisions : le f1-score global est de **82%**, et le modèle performe particulièrement bien sur la classe Pneumonie Virale avec **97%**. Des améliorations peuvent être attendues sur les 3 autres classes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Précision** | **Rappel** | **Score F1** | **Population (jeu de validation)** |
| COVID | 0.73 | 0.81 | 0.77 | 269 |
| Lung Opacity | 0.79 | 0.74 | 0.76 | 269 |
| Normal | 0.81 | 0.77 | 0.79 | 269 |
| Viral Pneumonia | 0.97 | 0.98 | 0.97 | 269 |
| **Global** | 0.83 | 0.82 | 0.82 | 1076 |

Par la suite, nous avons expérimenté avec d’autres modèles d’apprentissage par transfert existants, en s’inspirant de modèles qui ont prouvé de bonnes performances dans des publications scientifiques. Ces modèles ont été entraînés sur le **même jeu de données** d’entraînement et testés sur le même jeu de validation. La séparation du jeu d’entraînement et de validation a utilisé le même paramètre (« seed ») dans tous les tests. Nous pouvons donc comparer les résultats entre modèles.

Nous avons testé l’approche par apprentissage avec les modèles suivants :

* VGG16 (présenté au dessus)
* VGG19
* XCEPTION
* ResNet50
* ResNet101
* ResNet50V2
* ResNet101v2
* RestNet152V2
* DenseNet121
* DenseNet201
* MobileNet
* MobileNetV2
* EfficientNetB0
* EfficientNetB7

Pour chacun des modèles, la fonction de **pré-traitement** adaptée, disponible dans la bibliothèque Keras, a été utilisée. Ces fonctions permettent d’appliquer aux images en entrée le même pré-traitement que les images qui ont été utilisées pour le pré-entraînement (ImageNet). En voici deux exemples (source Keras.io) :

* Fonction “preprocessing” de VGG16 : Les images sont converties de RVB à BGR, puis chaque canal de couleur est centré sur zéro par rapport au jeu de données ImageNet, sans mise à l’échelle
* Fonction “preprocessing” de ResNetV2 : Les valeurs des pixels sont mises à l’échelle entre -1 et 1.

Dans la première salve de tests, nous avons entraîné les modèles en « gelant » les couches du modèle de base, qui ont été pré-entraînées sur le jeu de données « ImageNet ». Nous avons atteint un score de justesse **entre 80% et 85%** avec la plupart des modèles.

Nous avons également tenté d’entraîner ces modèles en “dégelant” une partie des couches du modèle de base afin de les réentraîner sur le jeu de données fourni. Nous avons effectué plusieurs tests en faisant varier le modèle et le nombre de couches dégelées. Nous avons atteint un score de justesse de **91%** avec les modèles basés sur VGG19 et EfficientNetB0.

Les combinaisons testées sont présentées dans le tableau de synthèse en début de chapitre.

Nous retenons plusieurs leçons générales de ces expérimentations :

* L’approche par transfert permet d’obtenir de très bons résultats, avec une approche plus sobre (économie de la construction et du premier entraînement du modèle). Ceci est d’autant plus impressionnant que le jeu de données sur lequel ont été entraînés ces modèles, ImageNet, ne **contient pas d’images de radiologie**.
* Dans l’ensemble, le dégel et le réentraînement de couches du modèle de base permet d’obtenir une meilleure performance. Cela ne s’est pas toujours vérifié dans nos tests, et nous avons constaté que le choix du pas d’apprentissage était très influent. Il semble que l’utilisation d’un pas d’apprentissage plus petit, ou adaptable en fonction de la

Graphique avec les résultats

## Interprétation des résultats grâce à la méthode « Grad-CAM »

Pour chacun des modèles testés, l’analyse de la matrice de confusion et du rapport de classification, calculés sur un **jeu de test** qui n’a pas servi pour l’entraînement**,** nous ont permis de constater :

* Que la plupart des modèles d’apprentissage par transfert atteignent une bonne performance de classification sur 4 classes, **entre 85% et 90%**
* Que la performance n’est pas homogène en fonction des classes : la classe « Viral Pneumonia » a généralement un score F1 plus haut que les autres (95 à 98%).

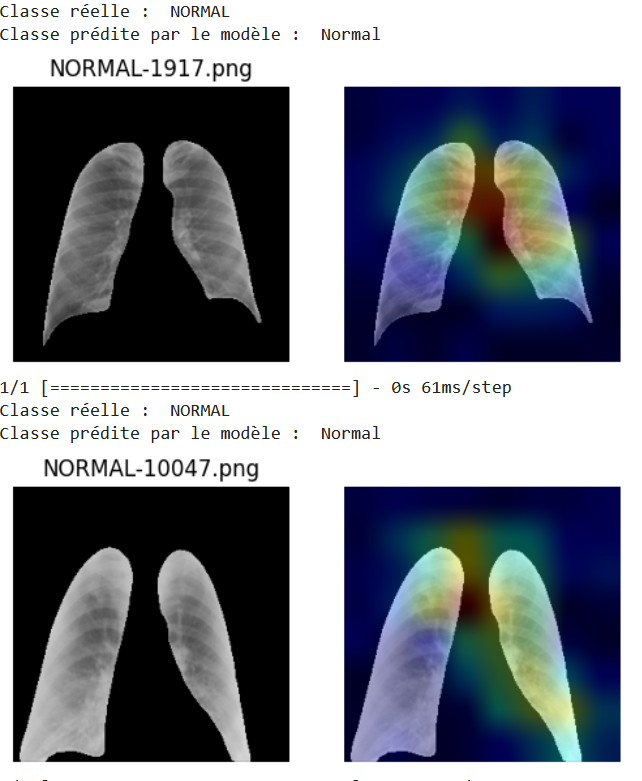
Il semble donc que les modèles parviennent mieux à extraire des signes caractéristiques de la pneumonie virale à partir des radios de poumons.

Pour aller plus loin, nous avons implémenté l'algorithme **Grad-CAM** (*« Gradient-weighted Class Activation Mapping »),* inspiré des exemples fournis par François Chollet sur le site de Keras[[5]](#footnote-5) ainsi que de nombreux projets Github.

Cette méthode consiste à extraire la dernière couche de convolution du modèle, avant les couches denses, et de calculer les gradients de la sortie de cette couche par rapport à l'activation de cette même couche. Ces gradients sont ensuite utilisés pour générer une carte d'activation, mettant en évidence les régions de l'image qui ont contribué le plus à la prédiction de notre modèle.

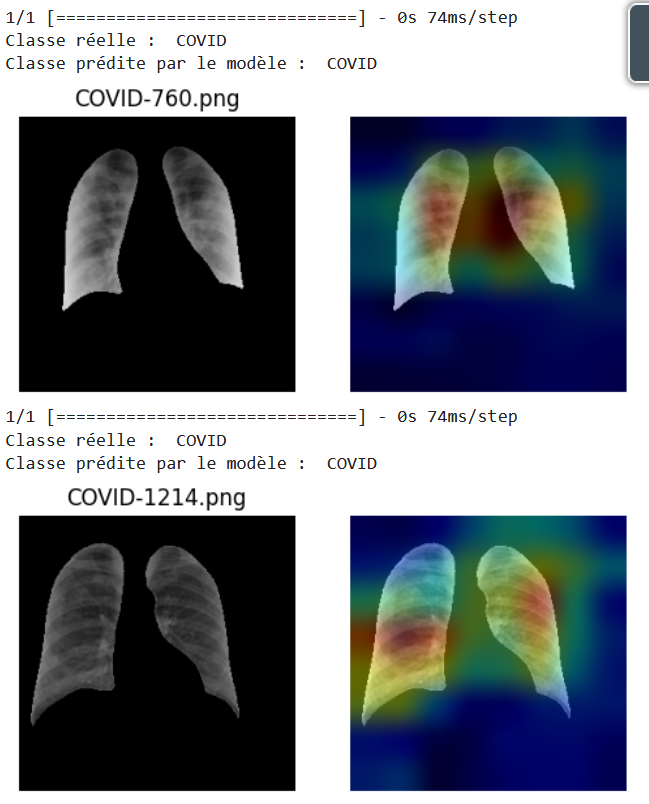
Nous présentons dans les pages suivantes plusieurs exemples de Grad-CAM appliqué au modèle d’apprentissage par transfert construit à partir de VGG16 et tuné précédemment, sur des images tirées au hasard dans chacune des classes du jeu de test.

### Grad-CAM sur une image de la classe Normale



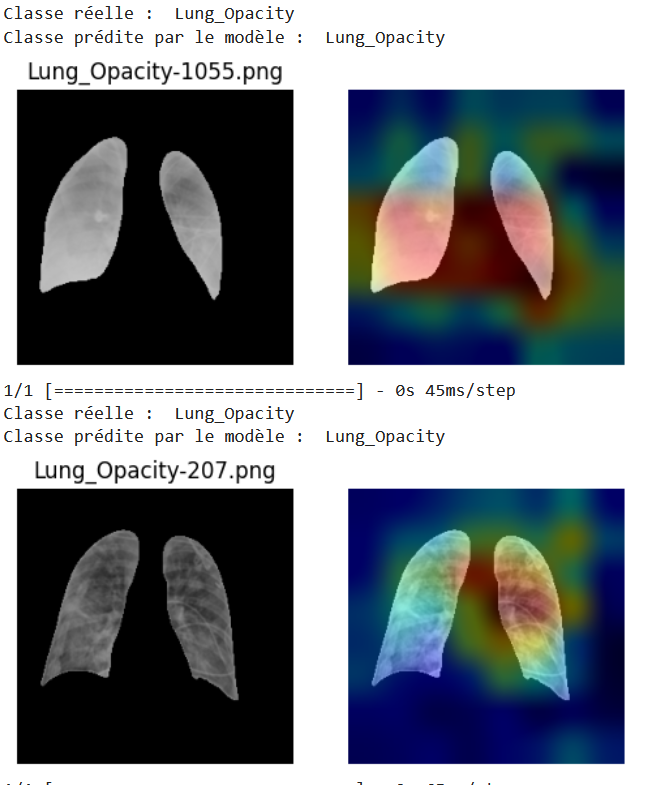
On constate que la zone activée par le modèle est centrée entre les poumons et sur les bords intérieurs dans la partie supérieure des poumons.

### Grad-CAM sur une image de la classe COVID



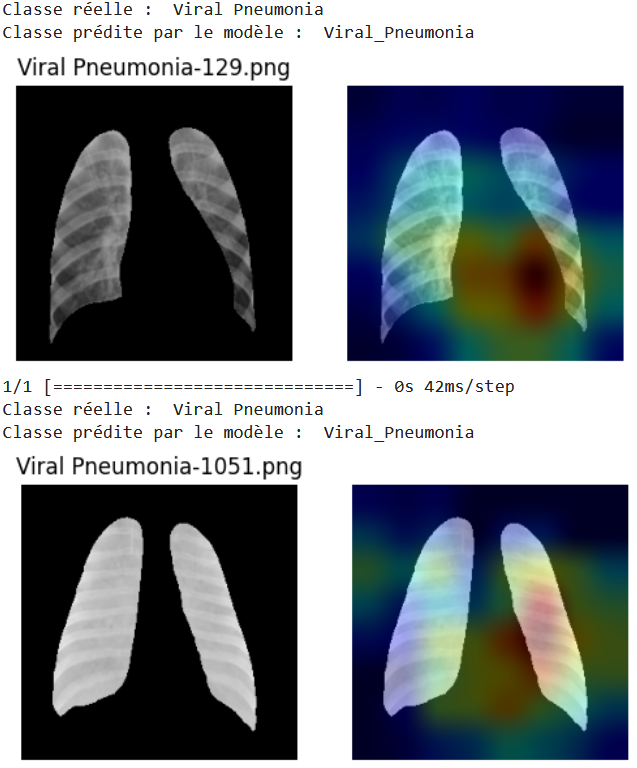
On constate que la zone activée par le modèle est plus localisée à l’intérieur des poumons.

### Grad-CAM sur une image de la classe « LUNG OPACITY »



On constate que la zone activée par le modèle est plus étendue que pour le Covid. Toutefois, elle est aussi assez différente d’une image à l’autre, et nous n’avons pas réussi à déterminer les caractéristiques particulières à l’œil nu.

### Grad-CAM sur une image de la classe « VIRAL PNEUMONIA »



On constate que la zone activée par le modèle semble localisée entre les poumons, dans la zone « masquée ».

Sans connaissance en analyse d’imagerie médicale, notre capacité à interpréter les sorties de nos modèles est limitée. Pour tenter d’aller plus loin, nous avons sollicité une radiologue professionnelle (qui n’était plus en activité au moment du Covid) et lui avons soumis notre approche et nos résultats Grad-CAM sur deux modèles (VGG19 / ResNet152V2).

**Il en ressort que les zones activées par les modèles (et mises en évidence par le Grad-CAM) correspondent peu aux signes pathologiques utilisés par les professionnels.**

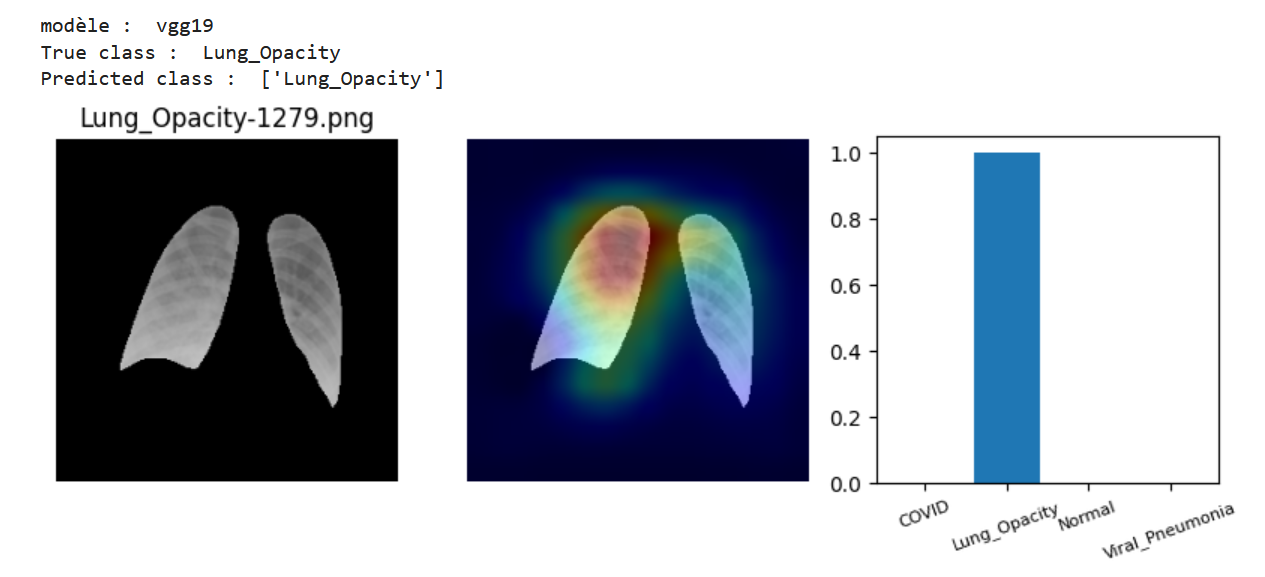
En effet, les radiologues se basent sur des anomalies visibles par rapport à l’image radio d’un patient sain. Les variations d’intensité d’une image radio sont dues aux obstacles rencontrés par les rayons-X sur leur trajectoire. Les poumons d’un patient sain contiennent uniquement de l’air, leur texture est donc homogène, et on visualise bien les frontières des côtes et des organes et vaisseaux sanguins.

Sur la radio d’un patient atteint d’une pathologie pulmonaire, un radiologue cherche des signes d’anomalie comme par exemple :

* Des « tâches » visibles à l’œil nu
* Des zones opaques qui traduisent la présence de fluide dans les poumons
* Des frontières floues entre les poumons et les organes (autour du cœur, du diaphragme par exemple)

Les informations en dehors de la zone pulmonaire fournissent aussi des indices sur la gravité de la maladie : intubation, présence de cathéters,…

### Interprétation croisée sur une image de la classe Lung Opacity (modèle VGG19)



Ici, notre radiologue a repéré la base des deux poumons comme étant en anomalie (opacité et côtes moins visibles). Or le Grad-CAM montre que le modèle a activé la zone qui paraît « normale » à l’œil du radiologue. L’attribut extrait par le modèle pour la classification est difficilement interprétable d’un point de vue médical.

De plus, il manque sur la radio masquée une partie du diaphragme (à droite sur l’image). Le masque fourni exclut donc une zone utile à l’analyse. Dans les approfondissements (chapitre suivant), nous étudierons l’impact du masquage sur la performance du modèle.

* Expliquer ici comment l’interprétation nous a menés à effectuer plusieurs approfondissements

# Approfondissements

## Expérimentation 1 : Approche réduite sur la base des distributions d’intensité

### Classification du problème

De même que la classification directe des images, la classification de la réduction d'image réduites à leur distribution d'intensité est complexe et a été expérimentée comme une branche parallèle de nos modélisations.

Nous avons tout d’abord expérimenté un modèle de classification simple sur la base des vecteurs d’intensité des images. Comme attendu, la performance a été aussi bonne que 'pile ou face'.

Ce problème réduit à un choix binaire Malade/Pas Malades est typiquement un problème de diagnostic.

Cependant selon l'usage médical, il existe deux situations de diagnostic qui nécessite de privilégier la performance selon certaines métriques :

* le diagnostic d'inclusion: privilégie la justesse (*« accuracy »)* et la spécificité
* le diagnostic d'exclusion privilégie la justesse (*« accuracy »)* et la sensibilité

La métrique de performance principale de cette classification binaire sera la justesse, mais nous considérerons également la sensibilité et spécificité dans un deuxième temps

Cette approche malade/normal, nous permettra de disposer d'un jeu de données plus larges. Nous disposons ainsi d'un jeu de 5164 images d'entraînements, 1292 pour la validation et enfin 1614 pour le jeu de test

Une image contenant capture d’écran

Description générée automatiquement

### Choix du modèle et optimisation

1. Expérimentation #1 : KNN, SMV, RandomForest.  Justesse < 60%
2. Expérimentation #2 : Apprentissage profond. Réseau type Le net en dimension 1.
3. Expérimentation #3 : Affinage d'une Le Net avec Keras Tuner
4. Optimisation de l'architecture et des paramètres avec Keras Tuner

*Par manque de temps nous avons stoppé cette branche d'exploration à ce niveau.*

Une image contenant texte, diagramme, Police, ligne

Description générée automatiquement

### Interprétation des résultats

L'amélioration du modèle au sein de chaque expérimentation est souvent venue de l'utilisation des bons paramètres dans Keras, et surtout de l'utilisation de Keras Tuner qui nous a permis de sélectionner la bonne architecture présenter à gauche

Les résultats de l'expérimentation 3 montre un entraînement optimisé du modèle, mais également une excellente performance en sensibilité de 94,5% tout en conservant une justesse acceptable de 80%.

|  |  |  |
| --- | --- | --- |
|  | Prédiction  MALADE | prédiction  NORMAL |
| MALADE réel | VP :764 | FN :43 |
| NORMAL REEL | FP :279 | VN :528 |

Une image contenant Tracé, capture d’écran, diagramme, ligne

Description générée automatiquement

Avec un faible taux de faux négatif, soit une bonne sensibilité, le modèle devient acceptable pour l'intégrer dans un protocole d'exclusion. Cependant nous pouvons et devons encore l'améliorer

## Expérimentation 2 : Augmentation de données (pré-traitement)

Pour tenter de réduire la sur-adaptation des modèles au jeu d’entraînement, nous avons réalisé plusieurs tests en ajoutant une phase **d’augmentation de données** en amont du réseau de neurones convolutif. Dans la littérature consultée pour le projet, cette approche a permis d’obtenir de bons résultats[[6]](#footnote-6).

Par exemple, nous avons reproduit le test d’apprentissage par transfert mené avec le modèle ResNet152V2, cette fois en ajoutant en amont du modèle une couche d’augmentation de données avec les opérations suivantes :

* Translation aléatoire +/-10%
* Rotation aléatoire +/-10°
* Retournement horizontal aléatoire
* Zoom aléatoire +/-15%
* Variation d’intensité aléatoire +/-10%

Ce test n’a pas obtenu un meilleur résultat que celui sans augmentation de données. La performance était même inférieure, toutes choses égales par ailleurs (voir tableau ci-dessous).

Nous pouvons émettre plusieurs hypothèses :

* Le jeu de données initial présente déjà une **forte variabilité** et l’augmentation de données n’apporte rien au processus d’apprentissage
* Les transformations appliquées n’ont **pas de sens d’un point de vue métier** (par exemple, la place des organes n’est pas symétrique, et moins d’une personne sur 10 000 a les organes inversés ; il n’est peut-être pas pertinent d’appliquer un retournement horizontal)

Résultats obtenus :

|  |  |  |  |
| --- | --- | --- | --- |
| **Modèle testé** | **N° d’exp.** | **Caractéristiques de l’expérimentation** | **Métrique de justesse sur le jeu de validation** |
| **ResNet152V2** | #5.1 | Apprentissage par transfert sans dégel de couches | **82%** |
| Appr2.1 | Expérience 5.1 avec augmentation de données | **77%** |

## Expérimentation 3 : Entraînement sur un jeu de données 3 fois plus important

Thibaut

## Expérimentation 4 : Pré-traitement supplémentaire des images en entrée

Lors de notre phase de pré-traitement, nous avions seulement choisi d’opérer une conversion en niveaux de gris et l’application des masques fournis. Or d’après la littérature scientifique parcourue pour le projet[[7]](#footnote-7), l’application de méthodes de pré-traitement des images peut améliorer la performance des modèles d’apprentissage par transfert classiques. Nous nous sommes donc inspirés d’une de ces publications pour comparer la performance d’un modèle **sans** pré-traitement supplémentaire et **avec pré-traitement** **: égalisation d’histogramme et flou gaussien.**

L’égalisation d’histogramme a été choisie suite à une discussion avec un radiologue, car le niveau de contraste est un facteur déterminant pour l’analyse des images radio, et pourrait donc constituer un biais pour le modèle (par exemple si toutes les images COVID ont nécessité un réglage particulier lors de l’examen radio).

Le flou gaussien a été choisi car il permet de réduire le « bruit » de l’image, qui peut aussi constituer un biais pour l’apprentissage.

**Toutefois, le pré-traitement supplémentaire n’a pas permis d’obtenir une meilleure performance.**

Résultats obtenus :

|  |  |  |  |
| --- | --- | --- | --- |
| **Modèle testé** | **N° d’exp.** | **Caractéristiques de l’expérimentation** | **Métrique de justesse sur le jeu de validation** |
| **ResNet152V2** |  |  |  |
|  |  |  |

## Expérimentation 5 : Classification 3 classes

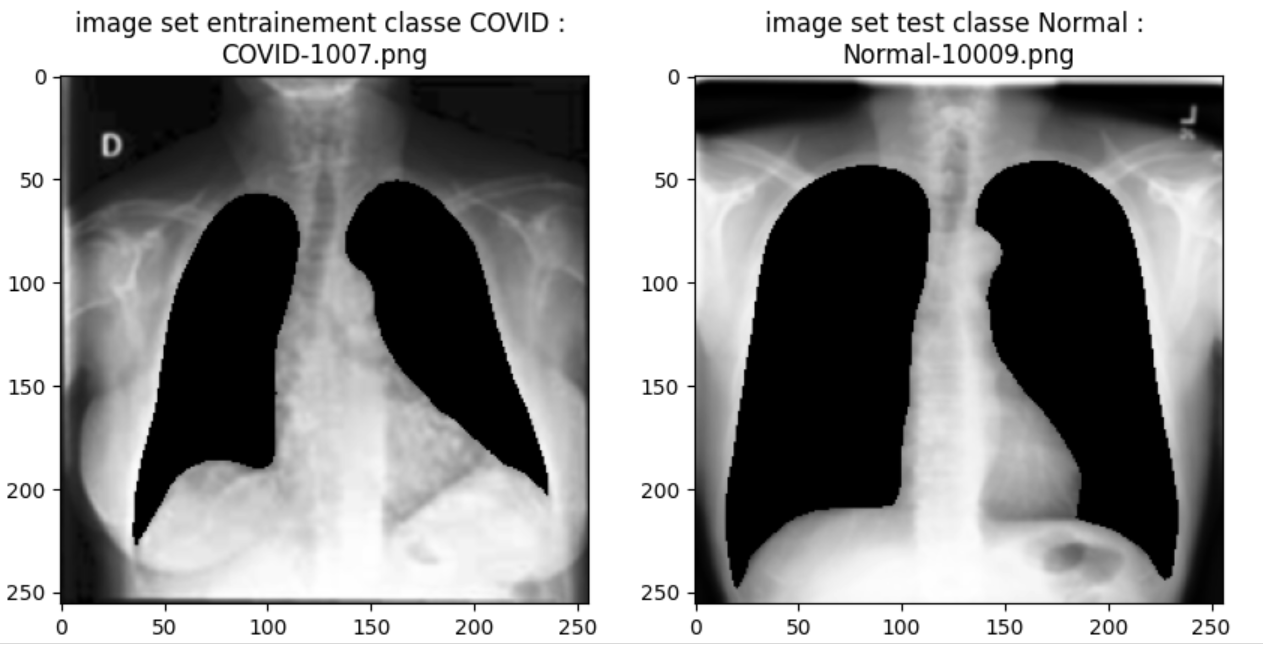
Comme la classe des autres pneumopathies ('Lung Opacity') est composé d'images de diverses pathologies, il pourrait en effet être considérer que cette classe est inhomogègne par rapport aux autres classes.

Nous avons donc conduits une dernières évaluation pour tester l'impact de cette classe sur le modèle. Ainsi nous avons crée un dernier de jeu de données à trois classes: Normal, Covid et pneumopathie virale.

L'impact est assez léger mais non négligeable car il permet d'uagmenter la justesse des modèles d'environ 1pts.

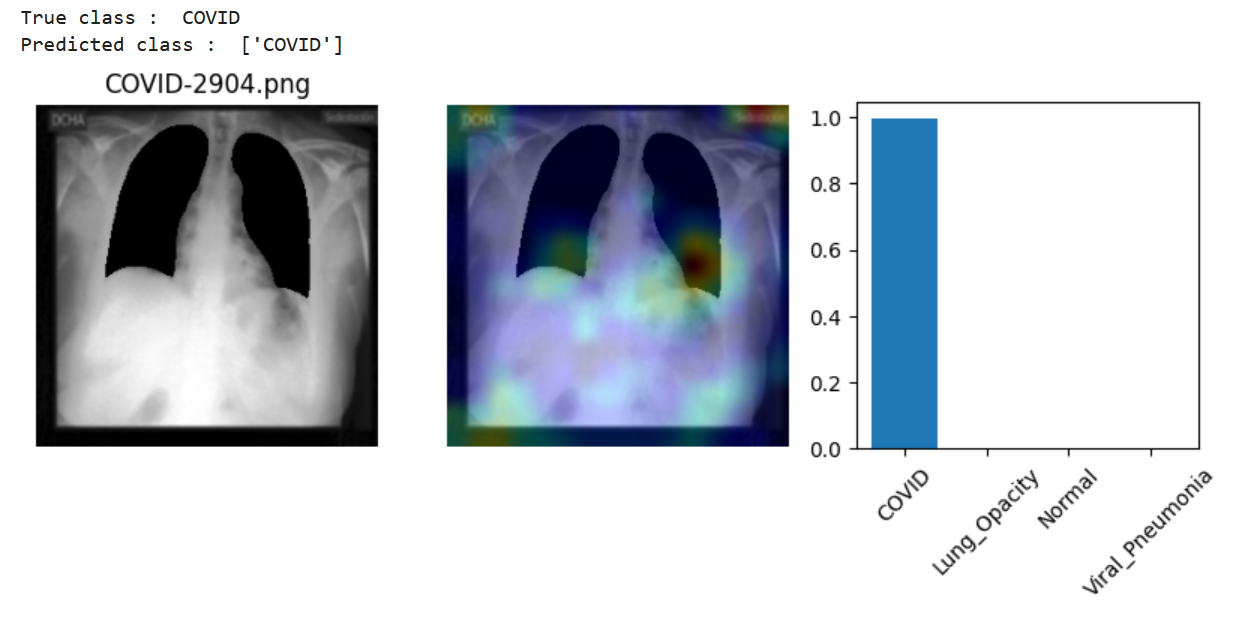
## Expérimentation 5 : entraînement d’un modèle sur 3 zones d’images : radio complète / zone pulmonaire uniquement / zone hors des poumons

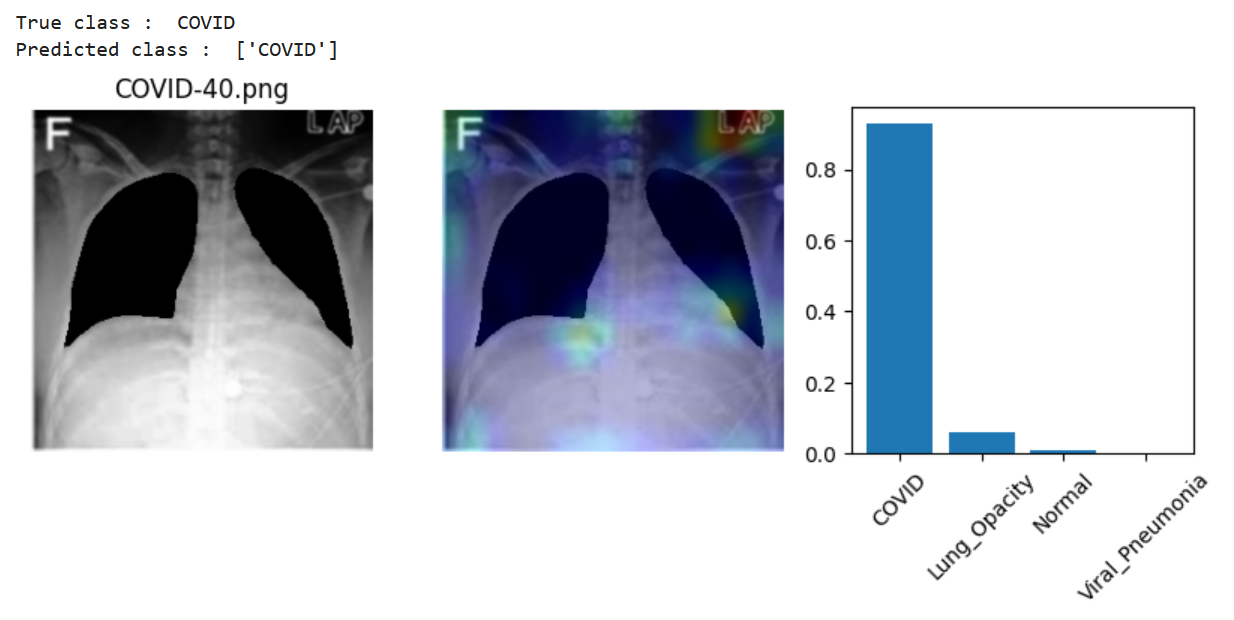
Thibaut

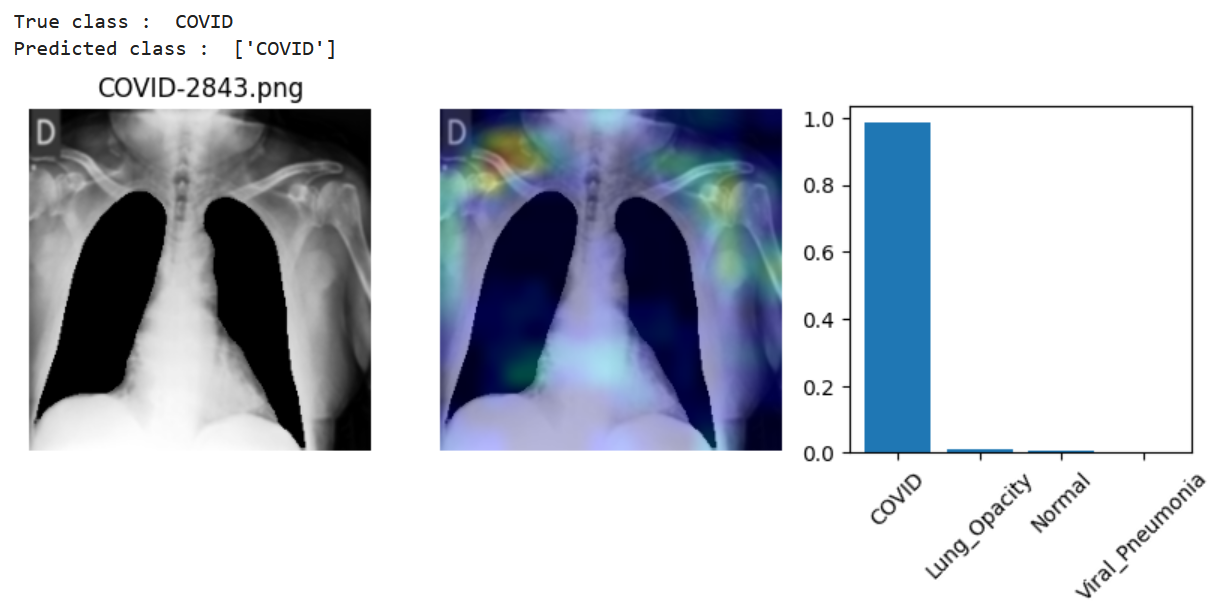


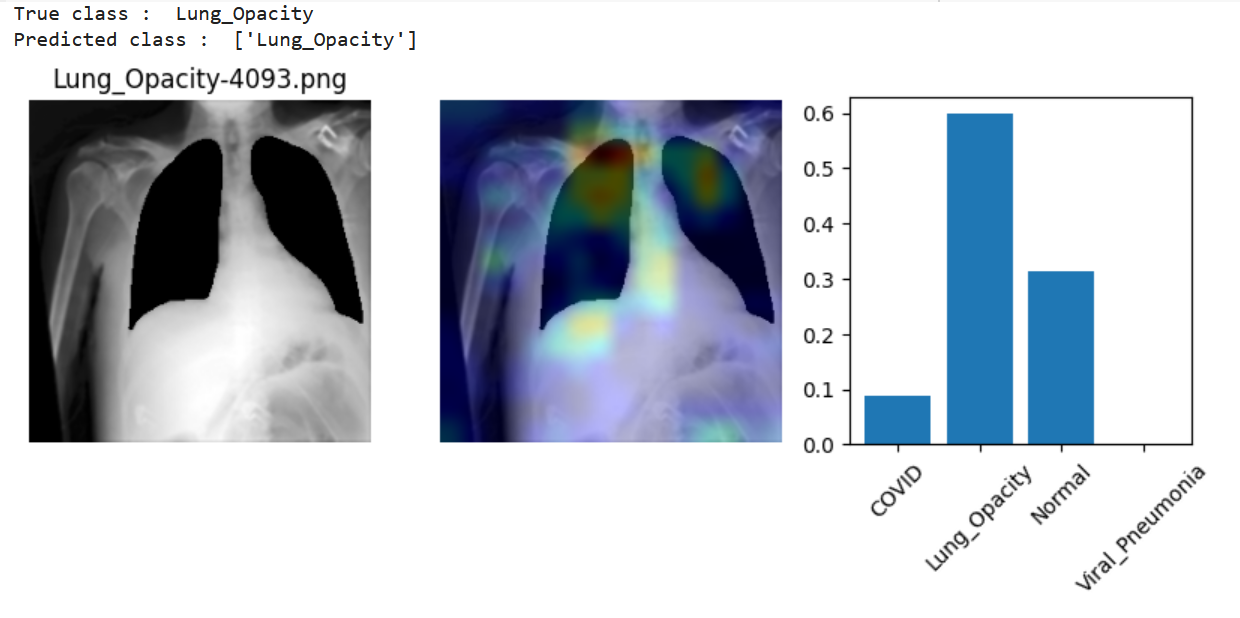
|  |  |  |
| --- | --- | --- |
| **Modèle testé** | **Caractéristiques de l’expérimentation** | **Métrique de justesse sur le jeu de validation** |
| **VGG19** | Apprentissage par transfert, sans dégel de couches, sur jeu de données à 12193 images : **zone pulmonaire uniquement** | **82%** |
| Apprentissage par transfert, sans dégel de couches, sur jeu de données à 12193 images : **images hors** **zone pulmonaire** | **89% !!!** |

Interprétation GradCAM :









# Conclusion et perspectives

## Pertinence de la solution

* Résultats
* Benchmarks
* Objectifs atteints : …

L'exploration du problème de diagnostic du Covid sur la base de radiographie, nous a amener de modéliser plusieurs approches et variantes au cours du projet :

* + La principale nommée **'approche directe'** consiste à classifier en 4 classes les radiographies avec leur masque tel que proposé par le jeu de donnée initiale.
    - Les principaux 'benchmarks' effectués le furent :
      * Entre LeNet les différents modèles de Transfert Learning, avec et sans réentrainement.
      * Et sur les différents hyperparamètres.
    - Les meilleurs modèles furent EfficientNet avec réentrainement, avec une justesse de classification de 91%
  + Une approche dite **'reduite'** consiste à trier les patients malades de patients sains sur la base exclusive des profils d'intensités des radiographies.
    - L'objectif de cette approche était d'obtenir des performances acceptables pour le tri des patients avec une économie de ressources de calcul et de stockage
    - Les différents modèles comparés variaient sur l'architecture des RNN et leurs hyperparamètres
    - Le meilleur modèle présente une justesse modeste de 80.1% mais une sensibilité de 94.7%
  + D'autres variantes ont été testé pour évaluer leur impact sur les modèles tel que:
    - L'augmentation de donnée qui n'a pas permis d'améliorer les performances Thibault, Nicolas
    - L'ajout de prétraitement supplémentaire en entrée. Thibault
    - L'exclusion d'une classe, en raison de l'hétérogénéité des ces pathologies, a permis d'augmenter très légèrement les performances de la justesse, soit environ 1pts pour les deux approches : 'directe' et 'réduite'

## Limites identifiées

La principale limite vient avant tout de l'imprécision de l'objectif Diagnostic du projet. En effet, rien que sur ce projet, nous pouvons définir 7 objectifs différents: 3 objectifs d'inclusions et 3 d'exclusions correspondant à chaque pathologie, mais aussi un objectifs de tri des patients

* Limite du jeu de données :
  + Incertitude vis-à-vis des sources
  + Aucune métadonnée disponible concernant les patients
  + Les symptômes du COVID évoluent au cours du temps : classification impactée par le timing de la radio
* Concernant le choix du modèle et son interprétabilité :
* Concernant l’équipe :
  + Manque la compétence d’analyse médicale pour évaluer la pertinence des « caractéristiques » retenues par le modèle et visualisées grâce au Grad-CAM

## Pistes d’amélioration et solutions alternatives à envisager

* Utiliser plusieurs modèles combinés, chacun étant performant sur une pathologie en particulier, avec un système de vote (source : Li & Li 2022 : deep learning voting algorithms combining multiple Convolutional neural networks)
* Pour l'approche réduite, cette classification peut être abordé comme un problème de signal stationnaire, l'utilisation d'architecture et de modèle de transfert learning plus adapté pourrait éventuellement améliorer la performance. Mais l'autre axes d'amelioration serait la combinaison à d'autres modèles

## Potentiel de généralisation pour un usage réel

* Aide pour le triage rapide des patients en cas de pandémie
* Mais préférences données par les praticiens médicaux aux modèles qui leur permettent de justifier un diagnostic ou de compléter un ensemble d’analyses. Certains modèles trouvés dans la littérature sont entraînés pour annoter les images radio avec les anomalies (opacités, nodules, …) qui permettent au radiologue de confirmer son diagnostic ou d’effectuer des analyses supplémentaires
* Risque de grande variabilité dans la qualité des données
* Besoin d’un pré-traitement des données en entrée vs grande résolution des images radio = enjeu de puissance de calcul pour pré-traiter les images et fournir un résultat instantané

## L’avis d’un radiologue professionnel

…

…

## Bénéfice du projet pour l’équipe

* Courbe de croissance rapide sur l’apprentissage profond, en avance par rapport au cours
* Les outils disponibles (TF, Keras, OpencV, Scikit-learn….) et la capacité de calcul à faible coût (Colab) permettent d’obtenir des résultats significatifs en quelques expérimentations
* Approche expérimentale par essai-erreur, forte autonomie pour trouver et tester des solutions disponibles sur Internet

## Difficultés rencontrées

A faire :

* Prévisionnel : tâches qui ont pris plus de temps que prévu, etc.
* Jeux de données : acquisition, volumétrie, traitement, agrégation, etc.
* Compétences techniques / théoriques : timing d’acquisition des compétences, compétence non proposée en formation, etc.
* Pertinence : de l’approche, du modèle, des données, etc.
* IT : puissance de stockage, puissance computationnelle, etc.

# Annexes

## Bibliographie

* Yadav, Ruchi & Sahoo, Debasis & Graham, Ruffin. (2020). **Thoracic imaging in COVID–19**. Cleveland Clinic Journal of Medicine. 87. 10.3949/ccjm.87a.ccc032.
* Wong HYF, Lam HYS, Fong AH, et al. **Frequency and distribution of chest radiographic ﬁndings in COVID-19 positive patients** [published online ahead of print, 2019 Mar 27]. Radiology 2019;201160.doi:10.1148/radiol.2020201160
* Narin A., Kaya C., Pamuk Z. **Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks**. arXiv preprint arXiv:2003.10849. 2020 [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=arXiv+preprint+arXiv:2003.10849&title=Automatic+detection+of+coronavirus+disease+(covid-19)+using+x-ray+images+and+deep+convolutional+neural+networks&author=A.+Narin&author=C.+Kaya&author=Z.+Pamuk&publication_year=2020&)] [[Ref list](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7510591/#bib0010)]
* Appasami G, Nickolas S. **A deep learning-based COVID-19 classification from chest X-ray image: case study**. Eur Phys J Spec Top. 2022;231(18-20):3767-3777. doi: 10.1140/epjs/s11734-022-00647-x. Epub 2022 Aug 18. PMID: 35996535; PMCID: PMC9386662.
* Sadre, R., Sundaram, B., Majumdar, S. *et al.* **Validating deep learning inference during chest X-ray classification for COVID-19 screening**. *Sci Rep* **11**, 16075 (2021). <https://doi.org/10.1038/s41598-021-95561-y>
* Talaat, M.; Si, X.; Xi, J. **Multi-Level Training and Testing of CNN Models in Diagnosing Multi-Center COVID-19 and Pneumonia X-ray Images**. *Appl. Sci.* **2023**, *13*, 10270. https://doi.org/10.3390/app131810270
* Heidari M, Mirniaharikandehei S, Khuzani AZ, Danala G, Qiu Y, Zheng B. **Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms**. Int J Med Inform. 2020 Dec;144:104284. doi: 10.1016/j.ijmedinf.2020.104284. Epub 2020 Sep 23. PMID: 32992136; PMCID: PMC7510591.
* Wang L, Lin ZQ, Wong A. **COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images**. Sci Rep. 2020 Nov 11;10(1):19549. doi: 10.1038/s41598-020-76550-z. PMID: 33177550; PMCID: PMC7658227.

1. Source : 2. Narin A., Kaya C., Pamuk Z. Détection automatique de la maladie à coronavirus (covid-19) à l'aide d'images radiographiques et de réseaux de neurones convolutifs profonds. Prépublication arXiv arXiv :2003.10849. 2020 [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=arXiv+preprint+arXiv:2003.10849&title=Automatic+detection+of+coronavirus+disease+(covid-19)+using+x-ray+images+and+deep+convolutional+neural+networks&author=A.+Narin&author=C.+Kaya&author=Z.+Pamuk&publication_year=2020&)] [[Liste de](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7510591/#bib0010) références] [↑](#footnote-ref-1)
2. Les établissements de santé > édition 2020 > DREES (fiche 23 : L’équipement en imagerie des établissements de santé publics et privés à but non lucratif) : [lien](https://drees.solidarites-sante.gouv.fr/sites/default/files/2021-01/Fiche%2023%20-%20L%E2%80%99%C3%A9quipement%20en%20imagerie%20des%20%C3%A9tablissements%20de%20sant%C3%A9%20publics%20et%20priv%C3%A9s%20%C3%A0%20but%20non%20lucratif%20.pdf) [↑](#footnote-ref-2)
3. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/> [↑](#footnote-ref-3)
4. Source : 1. Wong HYF, Lam HYS, Fong AH, et al. Frequency and distribution of chest radiographic ﬁndings in COVID-19 positive patients [published online ahead of print, 2019 Mar 27]. Radiology 2019;201160.doi:10.1148/radiol.2020201160 [↑](#footnote-ref-4)
5. Grad-CAM class activation visualization, F. Chollet, <https://keras.io/examples/vision/grad_cam/> [↑](#footnote-ref-5)
6. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7658227/> [↑](#footnote-ref-6)
7. “Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms” <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7510591/> [↑](#footnote-ref-7)