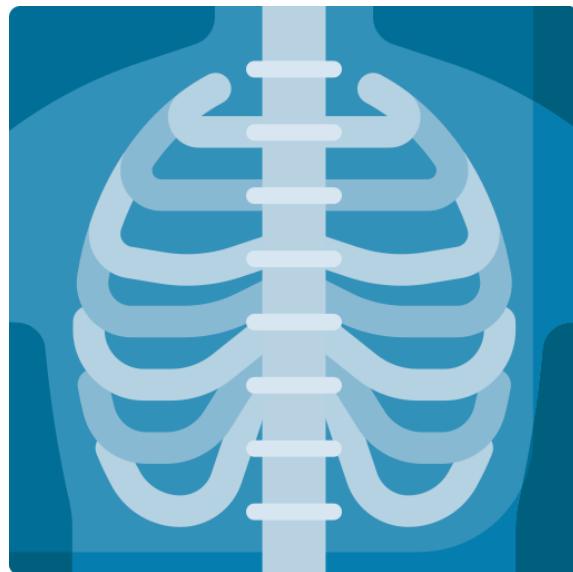


# Rapport Final du projet: Radiographie Pulmonaires

Bootcamp DS Sept24 : Chris HOZE, Mickael MELKOWSKI



# Table des matières:

<b>Contexte du projet.....</b>	<b>2</b>
Objectif.....	3
<b>Prise en main et découverte des données:.....</b>	<b>4</b>
Visualisation de quelques images du dataset:.....	4
Pre-processing des données.....	5
Création d'un dataset tabulaire pour l'exploration des données.....	5
Centrage et réduction des images.....	5
Statistiques descriptives sur les données.....	6
Répartition des données par catégorie.....	6
Répartition des données par source.....	6
Visualisation des différences entre catégories.....	7
Images moyennes par type et par source.....	9
Interprétation.....	10
Analyses en composantes principales.....	10
Part de variance expliquée par les composantes de l'ACP.....	10
Visualisation sur les deux premières composantes de l'ACP.....	11
Visualisation sur les troisième et quatrième composantes de l'ACP.....	11
Visualisation des composantes de l'ACP.....	12
Interprétation.....	12
Conclusion sur l'exploration des données.....	12
<b>Premiers modèles de classification.....</b>	<b>13</b>
Forêts aléatoires.....	13
Modèles de deep-learning sur dataset réduit.....	13
LeNet via Tensorflow : .....	14
Accuracy et Loss.....	14
Matrice de confusion.....	14
Dense121 via Pytorch : .....	15
Accuracy et Loss.....	15
Matrice de confusion.....	15
Conclusion : .....	15
<b>Modèles de deep-learning sur dataset complet.....</b>	<b>16</b>
<b>    Choix du modèle de base.....</b>	<b>16</b>
Modèle Resnet50 (177 couches).....	16
Accuracy et loss.....	16
Matrice de confusion.....	16
Modèle EfficientNetB2 (342 couches).....	17
Conclusion.....	17
<b>    Augmentation de données, choix du learning rate.....</b>	<b>18</b>
<b>    Fine tuning : Dégel des dernières couches.....</b>	<b>18</b>
Loss et accuracy.....	18
Matrice de confusion.....	19

Conclusion.....	19
Visualiser les régions utiles à la prédiction avec GradCam.....	20
Conclusion.....	20
<b>Utilisation de modèles plus avancés.....</b>	<b>20</b>
Choix du modèle.....	20
Modèle VGG19.....	21
Modèle EfficientNetB0.....	22
Modèle EfficientNetB4.....	23
Modèle DenseNet169.....	24
Modèle Resnet50.....	25
Modèle Resnet101.....	26
Choix de l'optimiseur.....	28
CosineDecay.....	28
ExponentialDecay.....	29
Conclusion.....	29
<b>Modèle définitif.....</b>	<b>30</b>
Paramètres d'entraînement.....	30
Résultat de l'entraînement.....	31
Conclusion.....	32
<b>Généralisation du modèle.....</b>	<b>33</b>
Prédiction de masques : modèle de segmentation.....	33
Données utilisées pour le modèle de segmentation.....	33
Modèle utilisé pour la segmentation.....	33
Performances du modèle.....	34
Prédiction du diagnostic COVID : modèle de classification.....	35
Données utilisées.....	35
Masquage des images.....	35
Classification des images.....	36
Modèle RESNET50.....	36
Modèle EfficientNetB4.....	37
Analyse des erreurs de classifications.....	37
<b>Conclusion.....</b>	<b>38</b>
<b>Annexe 1 : Résultat du modèle VIT.....</b>	<b>39</b>

## Contexte du projet

Une équipe de chercheurs de multiples universités du Moyen Orient et d'Asie ont assemblé un jeu de données de radiographie du thorax pour des patients, sain, atteint du Covid, de pneumonie et d'opacité pulmonaire.

Ce dataset a été constitué dans le but d'utiliser le deep learning pour le diagnostic du Covid-19 par X-ray plutôt que par RT-PCR<sup>1</sup> avant le déploiement des tests antigéniques.

Les données se répartissent en quatre classes comme indiqués ci-dessous :

- COVID-19 data (3615) : les radiographies des patients atteint du COVID
- Normal images (10192) : les radiographies des patients sans affection
- Lung opacity images (6012) : celles des patients présentant une opacité pulmonaire
- Viral Pneumonia images (1345) : celles des patients atteint d'une pneumonie virale

Chaque radiographie est fournie avec un masque pré-calculé, généré par apprentissage semi-automatique (i.e. créé en association entre experts en radiologie et experts en deep-learning). L'application du masque permet d'isoler les poumons ce qui permet de cibler les pixels des images les plus pertinents pour notre étude.



Les images des radios sont au format PNG en 299x299 pixels, celles des masques sont au même format mais en dimension 256x256 pixels.

## Objectif

On cherche à utiliser ce jeu de données pour développer un modèle de deep-learning permettant de diagnostiquer une affection pulmonaire. Ce sujet se présente comme un problème de classification multi-classes, on s'attachera ici à minimiser le nombre de faux-négatifs.

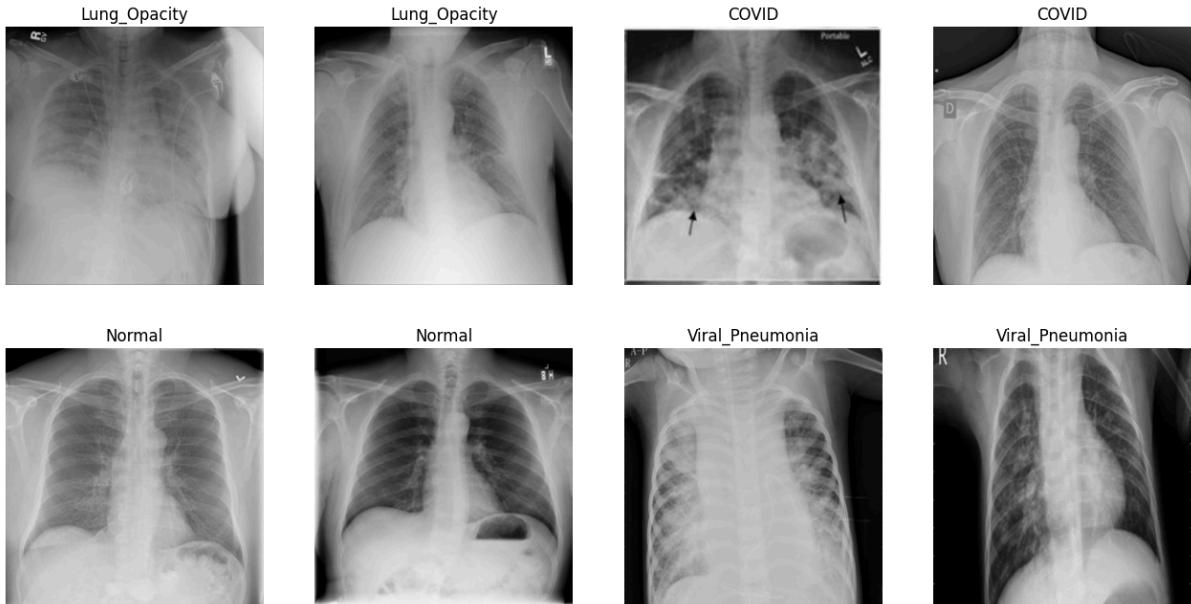
Dans un contexte de santé, il est préférable en première approche de ne pas risquer de considérer que la radiographie est normale alors que le patient est sujet à une infection.

---

<sup>1</sup> M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," in *IEEE Access*, vol. 8, pp. 132665-132676, 2020, doi: 10.1109/ACCESS.2020.3010287

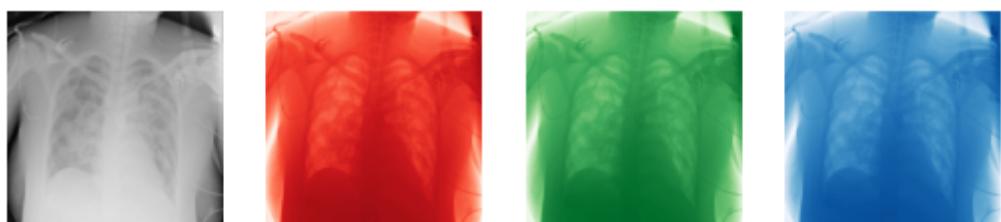
# Prise en main et découverte des données:

Visualisation de quelques images du dataset:



Les images sont des radiographies pulmonaires provenant de différentes sources, on remarque rapidement que seule une partie de l'image nous intéresse puisque le COVID est une maladie pulmonaire. On cherche à étudier uniquement sur les poumons alors que l'image laisse entrevoir également la colonne vertébrale et le cœur. Il faudra donc appliquer les masques fournis pour éliminer l'information non pertinente.

Les données sont en échelle de gris, une importation en couleur donne trois valeurs identiques pour les différents canaux. Chaque image correspond donc à une matrice de  $299*299*1$ . Pour vérification, on affiche ci-dessous le résultat d'une image importée en couleurs pour lesquels on chercherait à afficher chacun des trois canaux RGB..



Les seules métadonnées disponibles sur ce jeu de données concernent l'origine des radiographies. Aucune information n'est disponible sur les patients eux même (âge, sexe, fumeur ou non,...) malgré leur intérêt pour ce type d'analyse.

# Pre-processing des données

## Création d'un dataset tabulaire pour l'exploration des données

Dans un premier temps, on applique un preprocessing simple sur les données :

- Import
- Redimensionnement du masque à la taille de l'image de 256\*256 à 299\*299
- Application du masque
- Création d'un dataset tabulaire

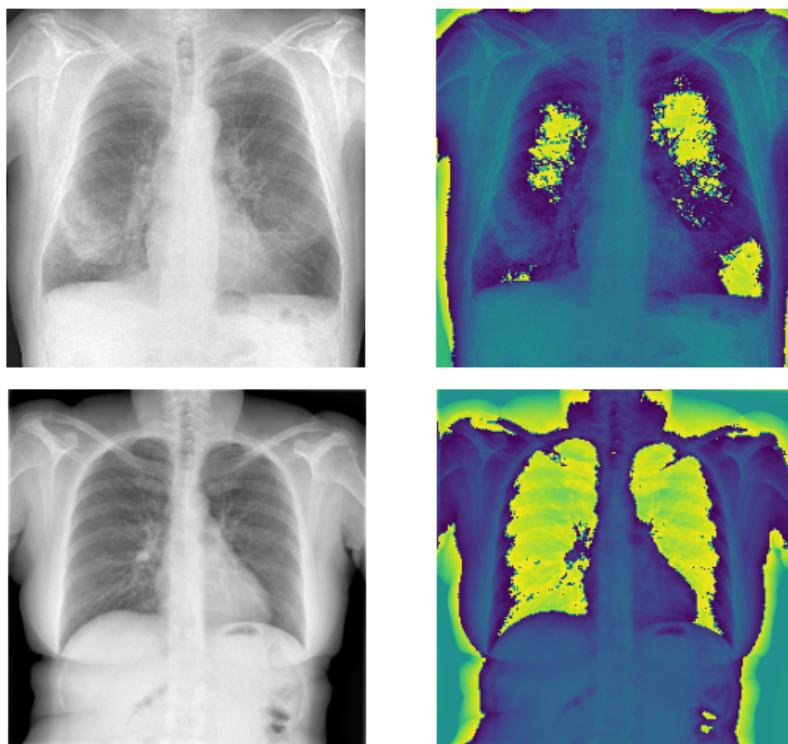
Un jeu de données tabulaire a été créé pour les analyses exploratoires, il contient 21 165 lignes et 89406 colonnes correspondant aux métadonnées et les 89400 (299 \* 299) pixels. Ce jeu de données sera utilisé pour l'exploration.

FILENAME	FORMAT	SIZE	URL	TYPE	num	0	1	2	3	...	89391	89392	89393	89394	89395	89396	89397	89398	89399	89400
Lung_Opacity-1	PNG	256*256	rnsa	Lung_Opacity	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Lung_Opacity-2	PNG	256*256	rnsa	Lung_Opacity	2	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Lung_Opacity-3	PNG	256*256	rnsa	Lung_Opacity	3	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Lung_Opacity-4	PNG	256*256	rnsa	Lung_Opacity	4	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Lung_Opacity-5	PNG	256*256	rnsa	Lung_Opacity	5	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

## Centrage et réduction des images

On remarque qu'une transformation des images pour que les valeurs des pixels se situent entre -1 et 1 (soustraction de 127 à la valeur du pixel et division par 127) permettent de mieux visualiser certains éléments des images et en particulier les poumons.

Ci-dessous un exemple de deux images avant et après ce processus :

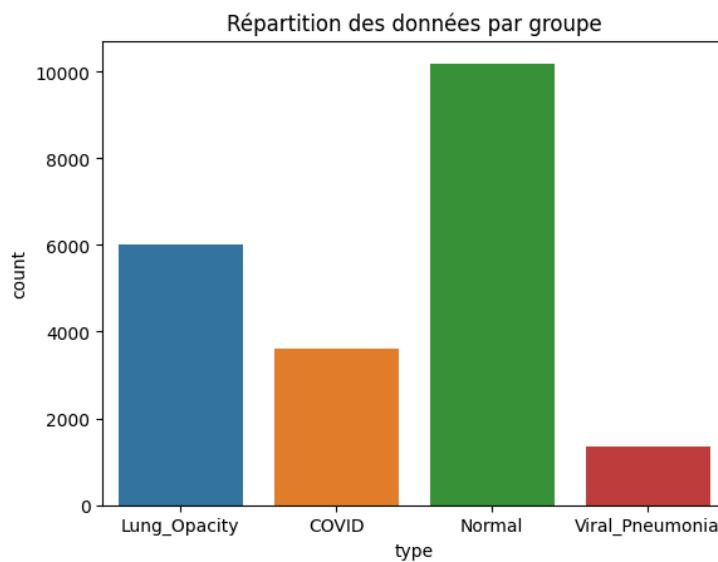


# Statistiques descriptives sur les données

Afin d'identifier un éventuel biais, on étudie en premier lieu comment se répartissent les images par catégorie et par source.

## Répartition des données par catégorie

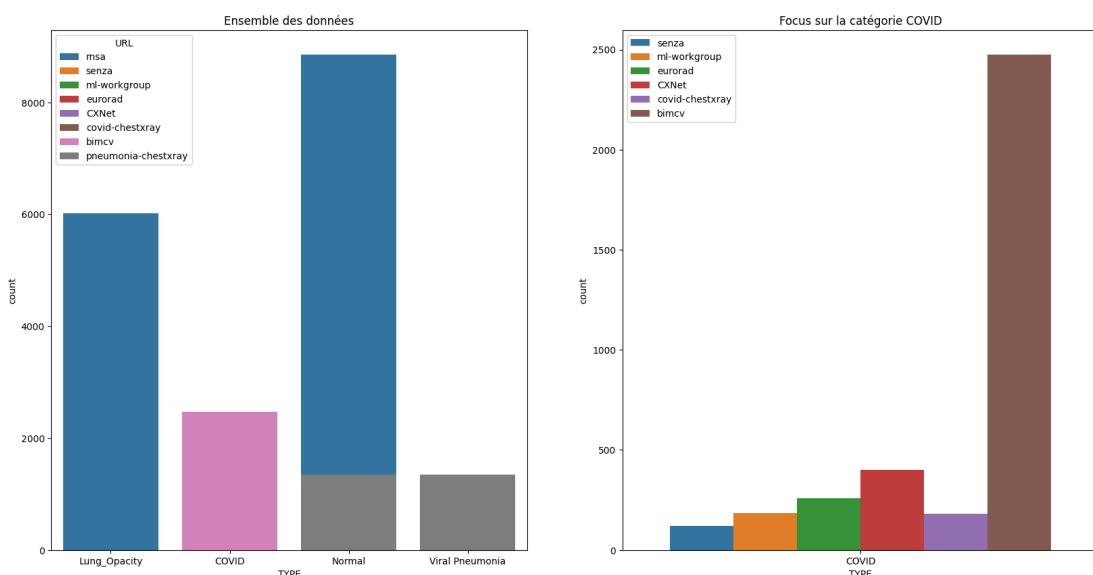
L'histogramme ci-dessous présente la répartition des images par catégories.



On observe que nous avons majoritairement des données pour les patients ne présentant pas d'affection ou une opacité pulmonaire. Les données de COVID sont minoritaires et cela est d'autant plus vrai pour les données de pneumonie virale.

## Répartition des données par source.

On visualise ensuite la répartition des catégories par groupe avec deux histogrammes pour l'ensemble des données (à gauche) pour les données de la catégorie COVID (à droite)



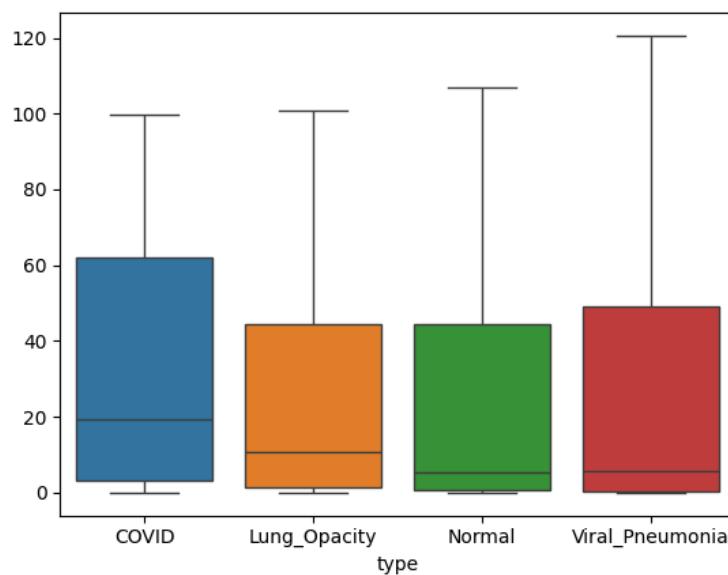
On remarque que les données de la catégorie COVID proviennent de sept sources différentes et en grande majorité de la source bimcv. Les données “Lung Opacity” et “Viral Pneumonia” proviennent chacune d'une source unique. Les données de catégorie “Normal” proviennent de deux sources différentes : les patients témoins des données d'opacité pulmonaire et de pneumonie virale. Ce recouplement partiel entre les sources devrait limiter la source de biais.

## Visualisation des différences entre catégories.

Dans une première approche, on crée une nouvelle variable contenant les valeurs moyennes pour l'ensemble des pixels pour chaque catégorie après masquage. Le tableau suivant résume les différences observées sur les valeurs des pixels, en fonction des jeux de données.

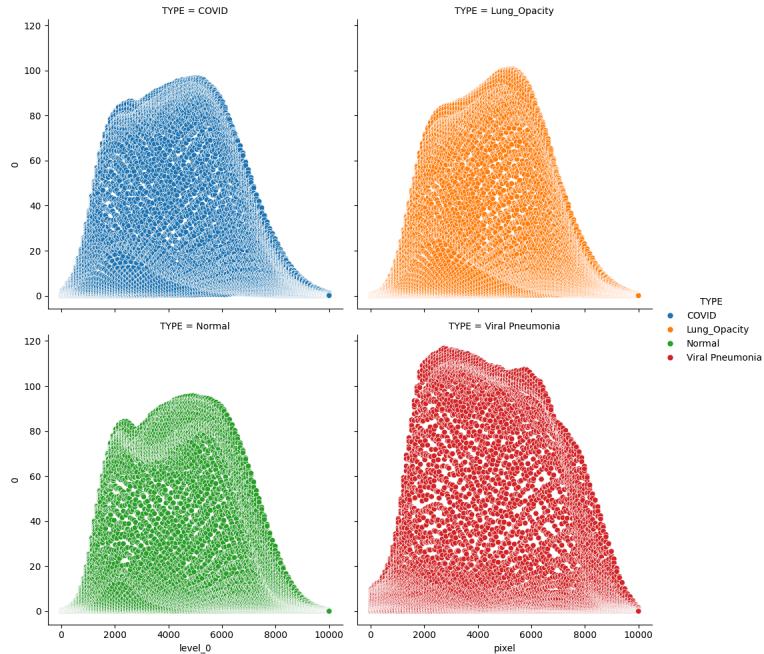
Catégorie	COVID	Lung Opacity	Normal	Viral Pneumonia
Moyenne	32,86	25,16	24,34	27,58
Ecart Type	32,94	29,31	32,22	37,52
Minimum	0,00	0,00	0,00	0,00
Médiane	19,41	10,64	5,18	5,54
Maximum	99,88	100,69	106,73	120,63

Les boîtes de dispersion ci-dessous permettent d'observer ces différences visuellement :

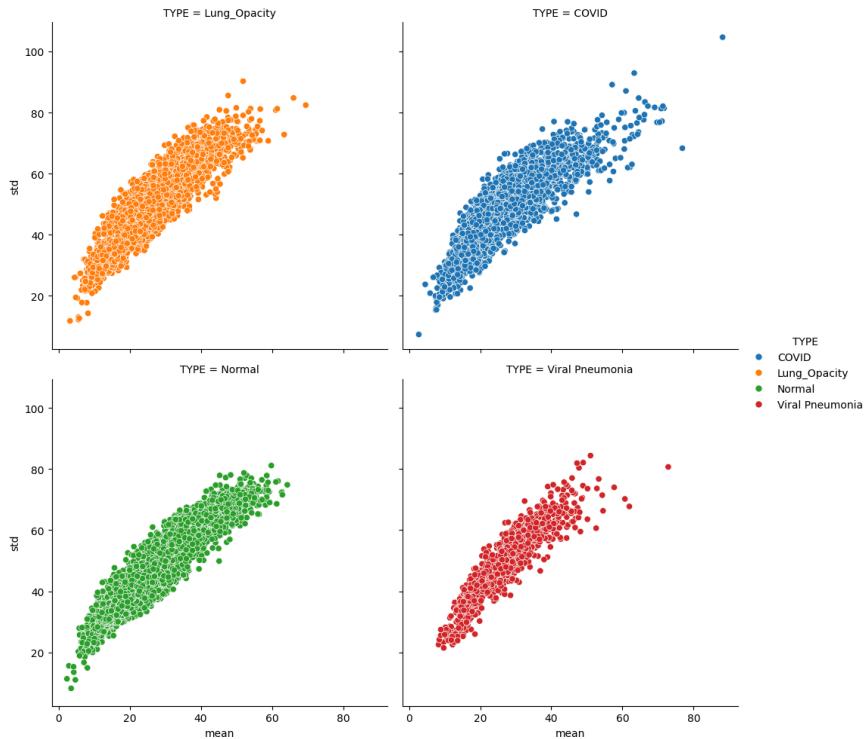


Il semble que la moyenne des données COVID soient plus élevée, ce qui suggère des images plus blanches et donc une cohérence avec la plus grande opacité attendue.  
Les données semblent plus extrêmes dans la catégorie Viral Pneumonia.

Une autre visualisation possible est de tracer le nuage de point de la valeur moyenne du pixel en ordonnées et l'ensemble des pixels en abscisse.



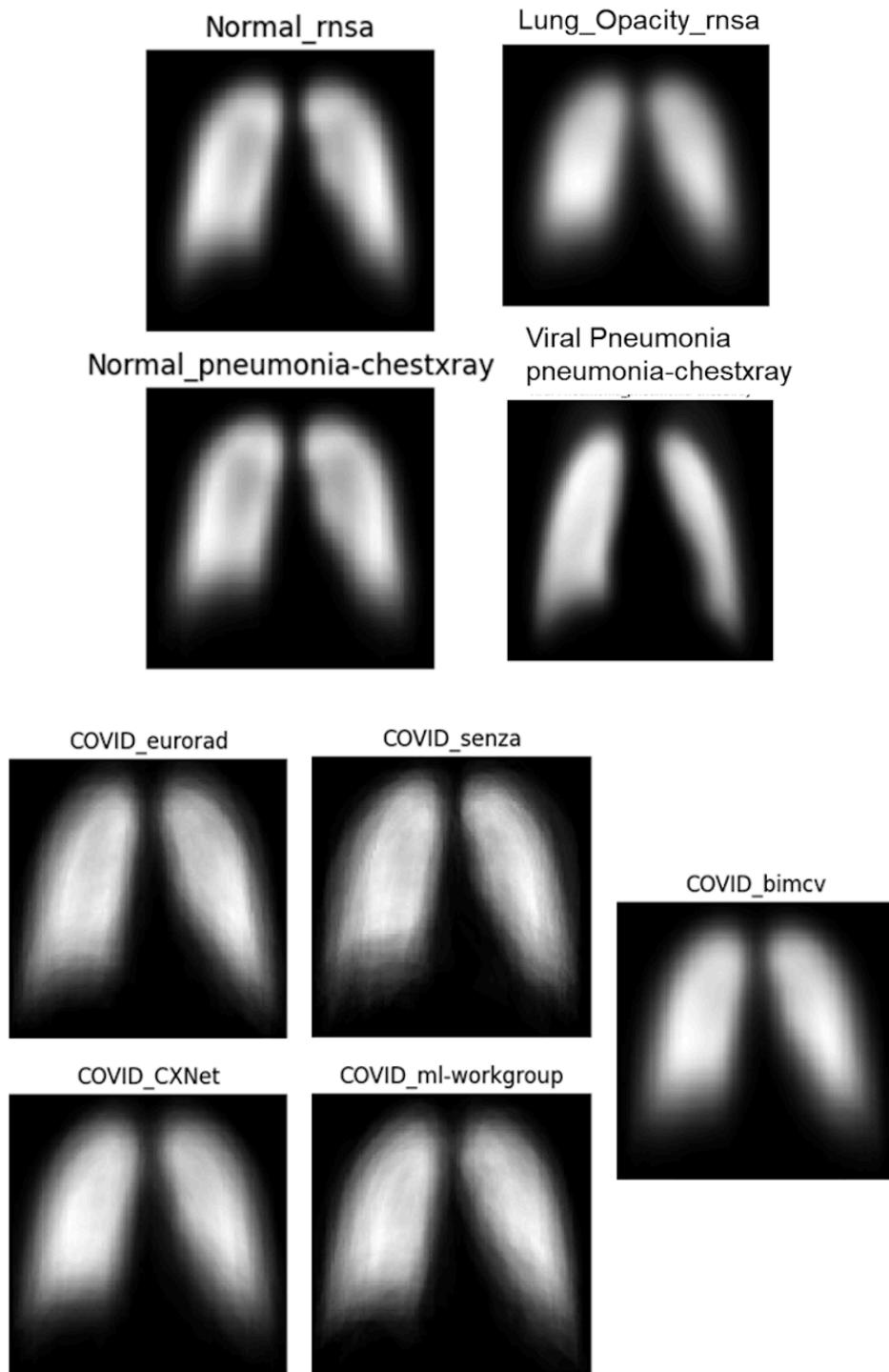
On peut également visualiser l'écart-type en fonction de la moyenne pour chaque catégorie.



Dans les deux représentations, on remarque que le comportement diffère en fonction des catégories. On distingue des formes plus sombres dans le cas des catégories COVID et Lung Opacity. Les données semblent également plus dispersées pour ces catégories.  
Pour mieux visualiser ces différences, on trace l'image moyenne par groupe.

## Images moyennes par type et par source

Dans un second temps, on visualise l'image moyenne par type et par sources pour y déceler un éventuel biais.



## Interprétation

Il semble y avoir un biais sur les données liées à la pneumonie virale : le cœur est proportionnellement plus important et l'application du masque efface la partie inférieure du poumon gauche. Ce biais est lié à l'origine des données issus du jeu de données "pneumonia-chestxray". Sans certitude, on émet l'hypothèse à la vue des images que ces données correspondent à des radios pulmonaires d'enfants.

Une partie des données issues de ce jeu de données étant également de catégorie "Normal" on retrouve le même phénomène de façon moins marquée sur cette catégorie.

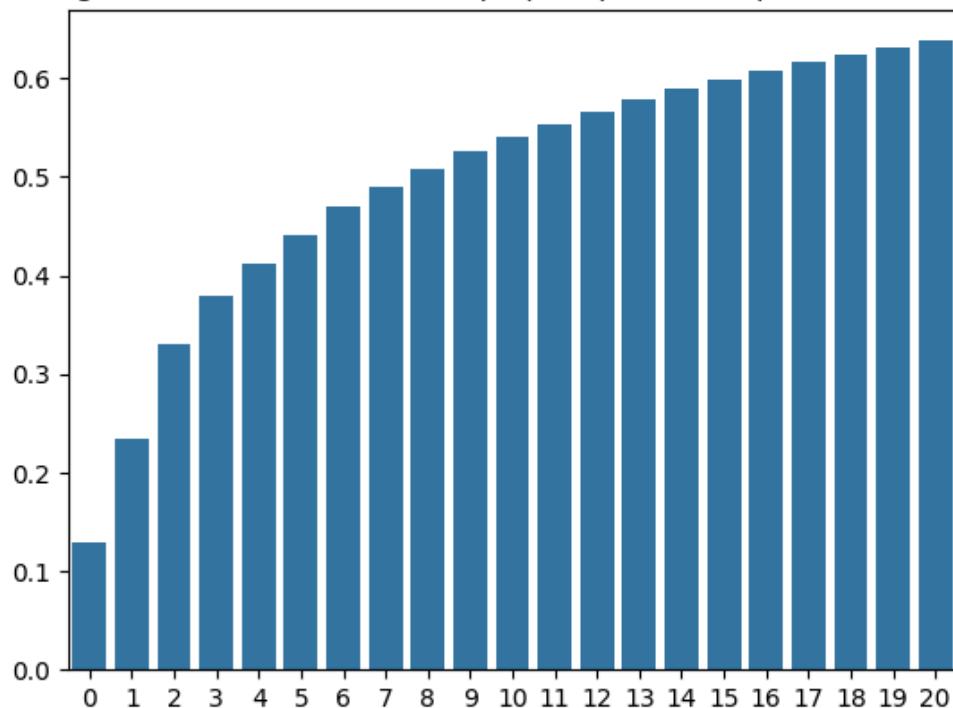
## Analyses en composantes principales

### Part de variance expliquée par les composantes de l'ACP

Pour des raisons de ressources informatique, l'analyse en composante principale a été réalisée en réduisant les images en dimension 100\*100.

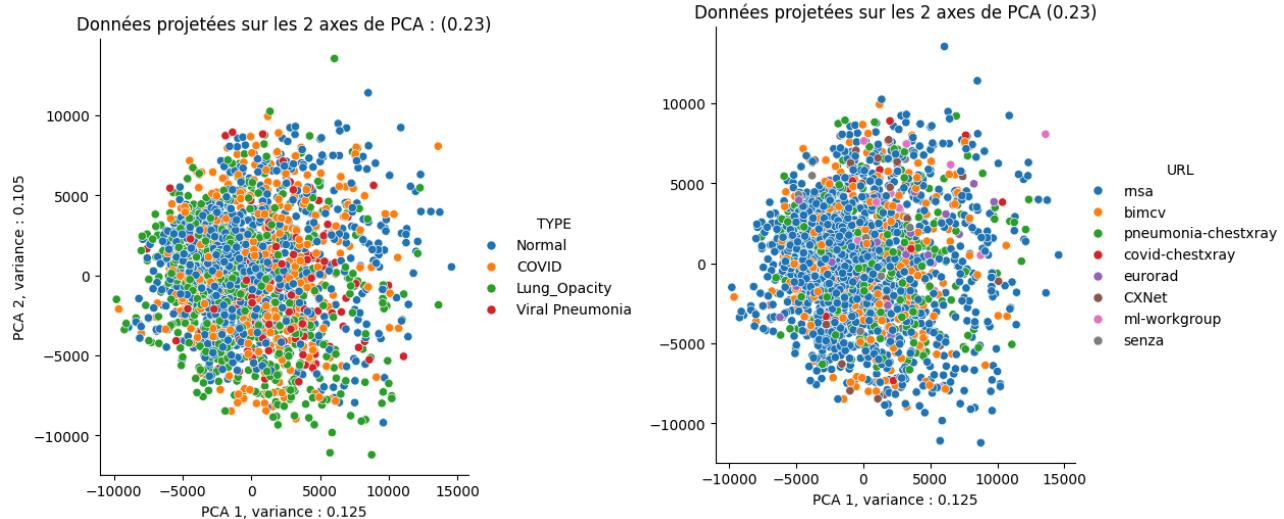
La première figure donne le pourcentage de variance sur les 20 premières composantes. On remarque que les dix premières composantes expliquent 51,6 % de la variabilité. Pour atteindre 95% de la variance initiale, il faut conserver 782 composantes.

Pourcentage de variance cumulée expliquée par les 20 premiers axes de l'ACP

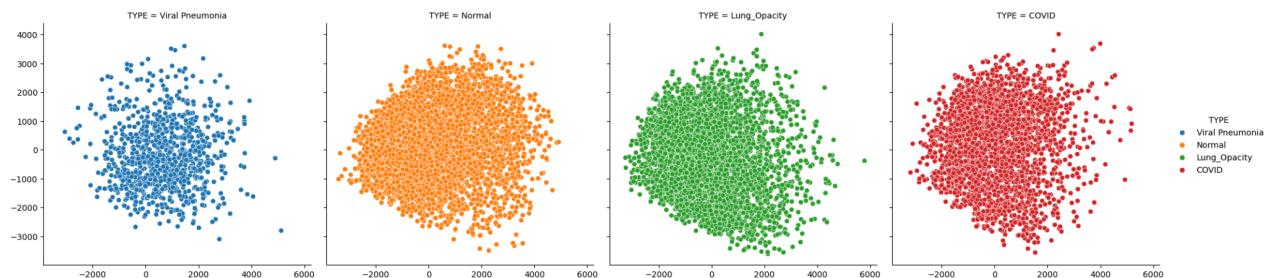


## Visualisation sur les deux premières composantes de l'ACP

On visualise les coordonnées des échantillons sur les plans 1 et 2 pour étudier s'il est possible de séparer les catégories selon un plan. On visualise également la répartition par source de données pour identifier une éventuelle source de biais. :

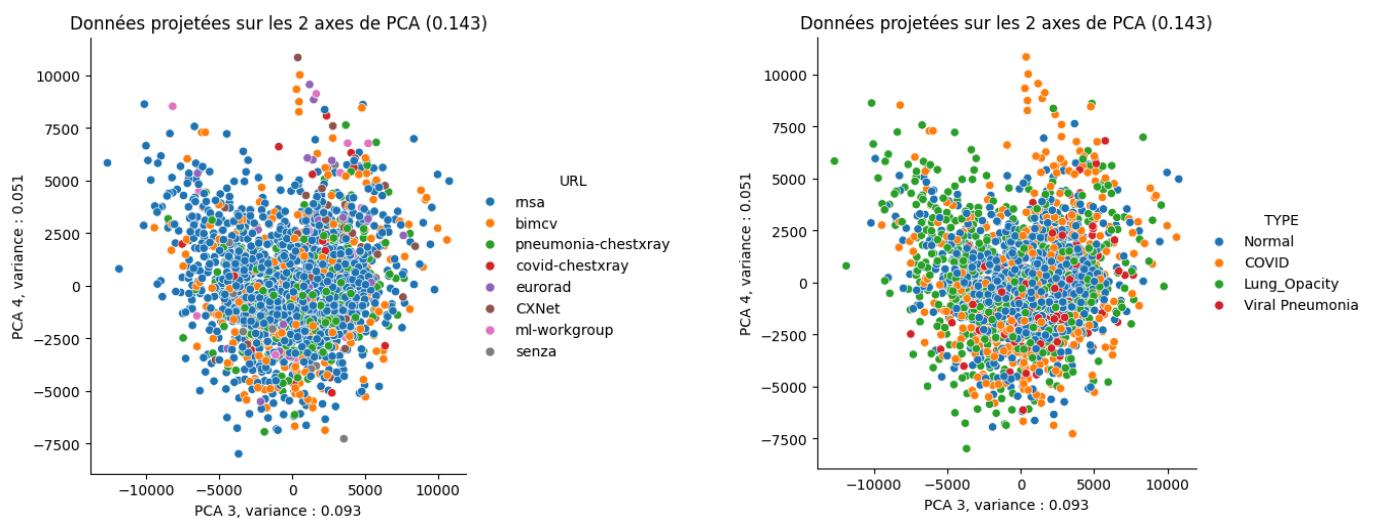


Pour plus de clarté, nous visualisons les différentes classes sur des graphiques distincts:



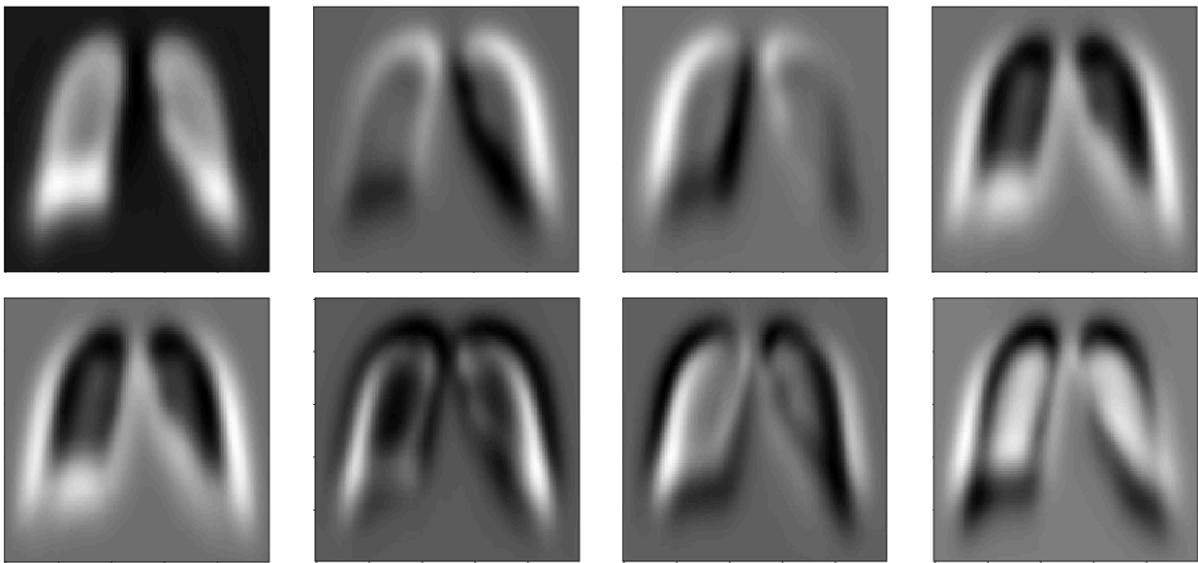
## Visualisation sur les troisième et quatrième composantes de l'ACP

On visualise les échantillons sur le plan créé par les composantes 3 et 4 de l'ACP.



## Visualisation des composantes de l'ACP

Les images ci-dessous correspondent à l'image des huit premières composantes de l'ACP.



On observe que les composantes sont essentiellement associées à la luminosité globale de l'image ainsi que celles des poumons plutôt qu'à des différences sur les poumons lui-même ce qui rejoint le fait que l'ACP ne soit pas capable de séparer les catégories d'images.

## Interprétation

Les axes ne permettent pas de séparer les échantillons par groupe, et on identifie pas de sous-population en fonction de la source de données.

L'orientation, le niveau de zoom et la taille des poumons diffèrent selon les patients et les radiographies ce qui peut expliquer pourquoi il est difficile de distinguer les classes avec les méthodes de statistiques classiques.

Un argument supplémentaire pour indiquer que l'emplacement des poumons sont variables au sein des radiographies sont variables est le faible nombre de variables qui sont identiques pour l'ensemble de l'échantillon (1321 soit 1.5%) après masquage alors qu'on aurait pu s'attendre à des zones masquées communes sur des régions non concernées par les poumons.

## Conclusion sur l'exploration des données

On ne constate donc pas de solution évidente à notre problème de classification avec une séparation dans un plan ce qui suggère que des modèles de classification classiques ne sont pas adaptés. Des solutions plus complexes permettant de prendre en compte un nombre plus important de caractéristiques seront donc à mettre en œuvre et en particulier l'emploi de modèle de Deep Learning et de Computer Vision.

Dans la suite du rapport, nous verrons les différentes étapes qui nous ont permis de choisir un modèle adapté à notre problématique explorant en plusieurs vagues des modèles déjà existants. Nous verrons d'abord un modèle de machine learning suivis des premiers essais en deep learning. Ensuite des modèles plus complexes de deep learning ont été employés jusqu'à pouvoir déterminer le modèle le plus adapté.

# Premiers modèles de classification

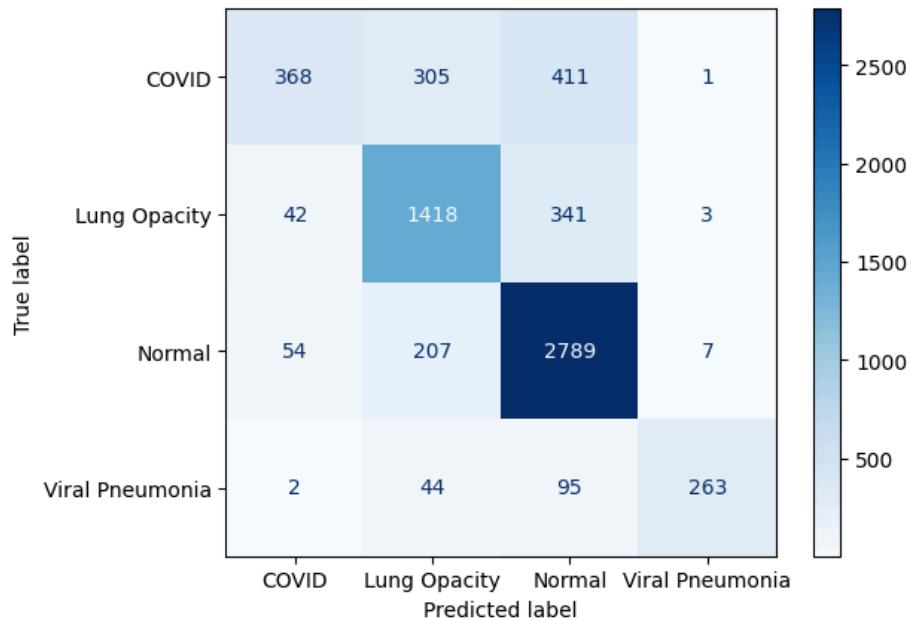
## Forêts aléatoires

En première approche, on applique une classification Random Forest sur les données transformées par l'ACP. L'ACP a été réalisée sur jeu de données complet après réduction du dataset en image de taille (100,100) soit 1000 variables correspondant aux pixels plus une correspondant à la source des données. Les données ont été séparées entre apprentissage (80%) et validation (20%) avant la réalisation de l'ACP. Le jeu de données transformé pour garder 95% de la variance contient 782 variables.

Une optimisation des paramètres est effectuée par validation croisée. On teste des valeurs entre 100 et 700 pour le nombre d'estimateurs, on cherche à optimiser le f1\_macro car le jeu de données est déséquilibré. Le meilleur modèle est celui avec un n\_estimators à 200.

Les résultats sont donnés dans le tableau de classification et la matrice de confusion suivante. On observe que malgré une précision moyennement élevée (0.76), le recall pour la classe COVID est particulièrement bas (0.34). Le modèle ne semble donc pas adapté pour cette analyse, ce qui est cohérent avec les résultats d'exploration.

	precision	recall	f1-score	support
COVID	0.79	0.34	0.47	1085
Lung_Opacity	0.72	0.79	0.75	1804
Normal	0.77	0.91	0.83	3057
Viral Pneumonia	0.96	0.65	0.78	404
accuracy			0.76	6350
macro avg	0.81	0.67	0.71	6350
weighted avg	0.77	0.76	0.74	6350



## Modèles de deep-learning sur dataset réduit

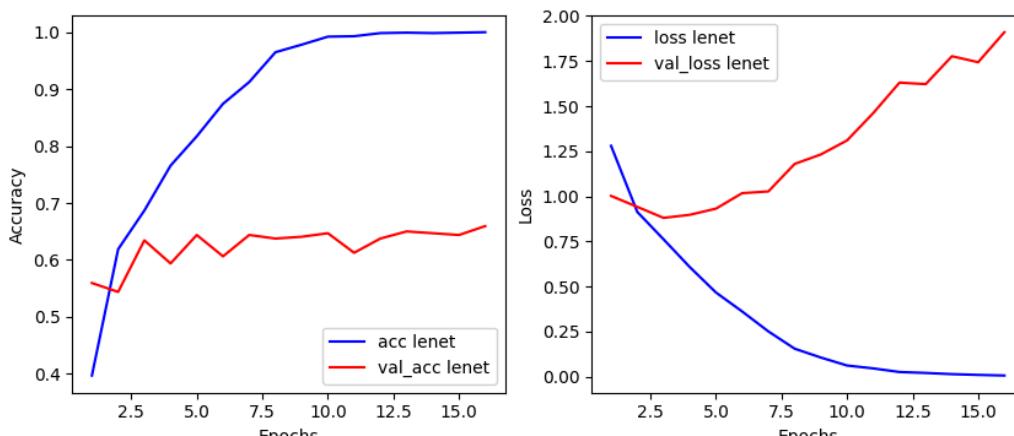
Des premiers modèles de deep-learning ont été réalisés avec un jeu de données réduit composés de 2000 images (500 par type) redimensionnées en 256\*256 pixels.

Deux modèles ont été testés, LeNet via Tensorflow et DenseNet121 via Pytorch dans le but de découvrir aussi les modules et leur fonctionnement.

Ce test a été réalisé avec 16 epochs et une taille de batch de 32.

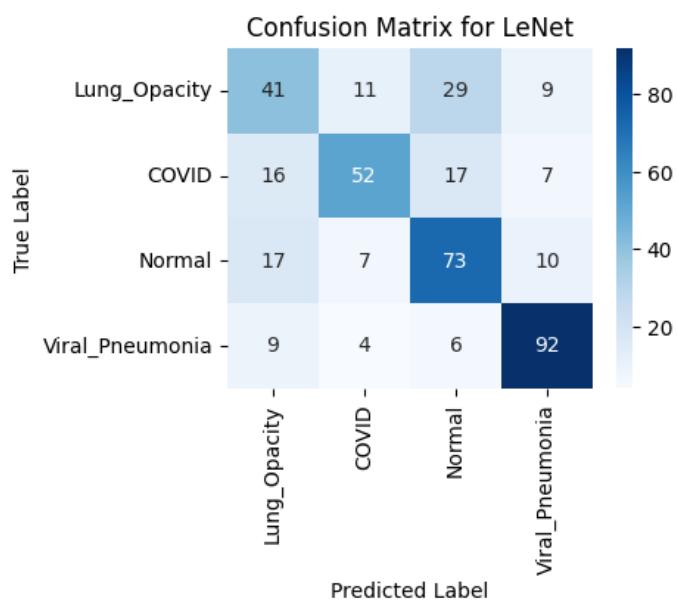
LeNet via Tensorflow :

Accuracy et Loss



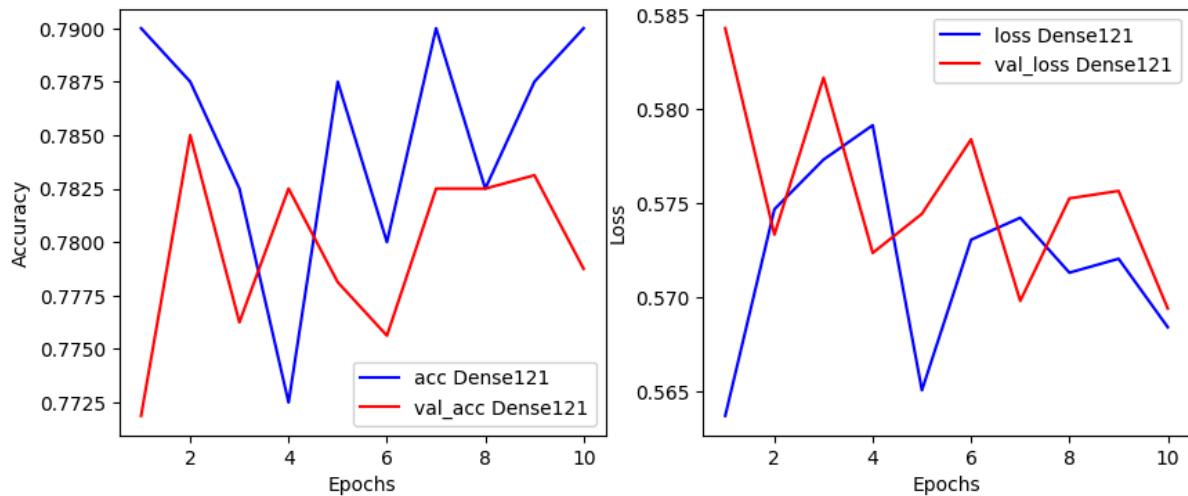
Il y a ici un overfitting sur les données, l'accuracy sur le jeu de validation ne s'améliorant que peu et la loss augmentant aussi.

Matrice de confusion



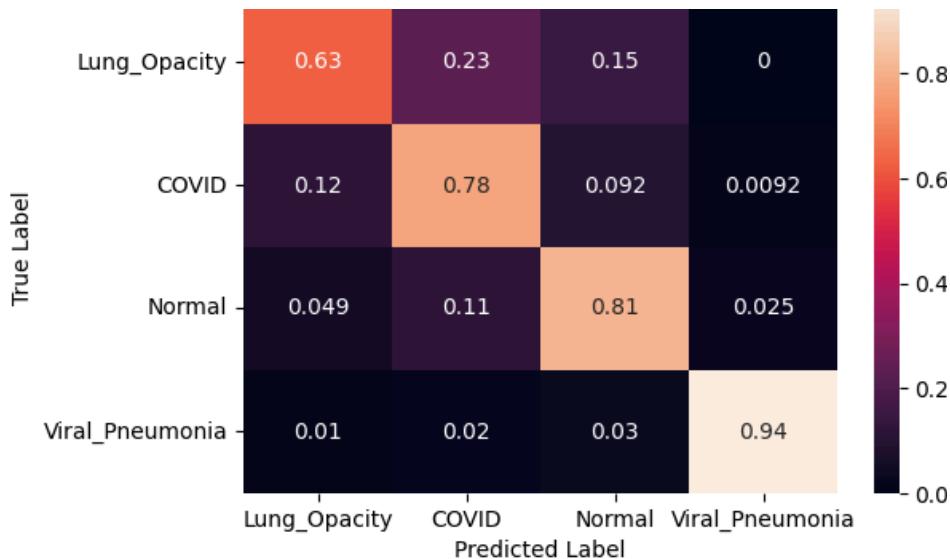
## Dense121 via Pytorch :

### Accuracy et Loss



La précision de ce modèle semble légèrement meilleure même si l'on est encore loin d'un modèle abouti.

### Matrice de confusion



### Conclusion :

Ces premiers modèles présentent des résultats encourageants sur la pertinence de l'utilisation du deep learning avec en moyenne environ 80% de classification correctes pour Dense-121.

L'amélioration ou le choix de modèles plus adaptés pourra nous permettre d'assurer de meilleurs résultats pour les prochaines phases du projet. Pour des raisons de praticité d'utilisation, on préférera l'utilisation de Keras et Tensorflow à celle de Pytorch.

# Modèles de deep-learning sur dataset complet

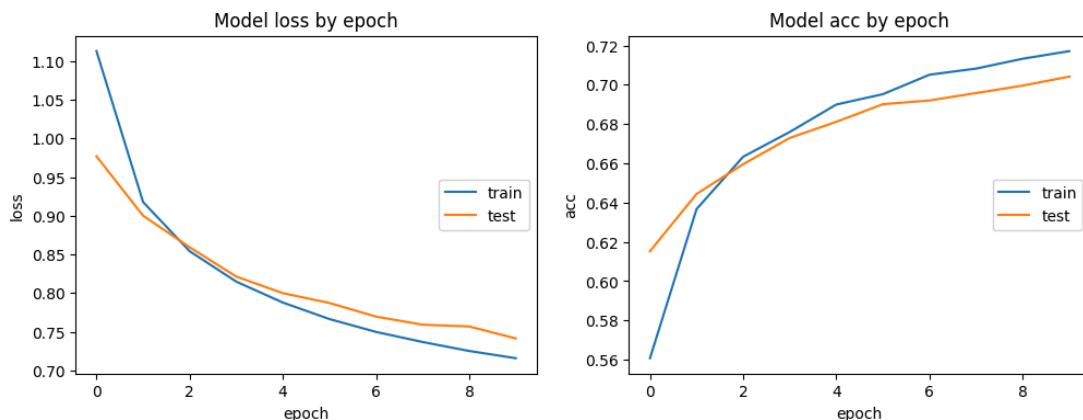
## Choix du modèle de base

Plusieurs réseaux de neurones de convolution (CNN) ont été testés. Les modèles définis par nos soins n'ayant pas des performances suffisantes nous nous sommes tournées vers des modèles plus complexes disponibles directement via keras ou d'autres sources en ligne.

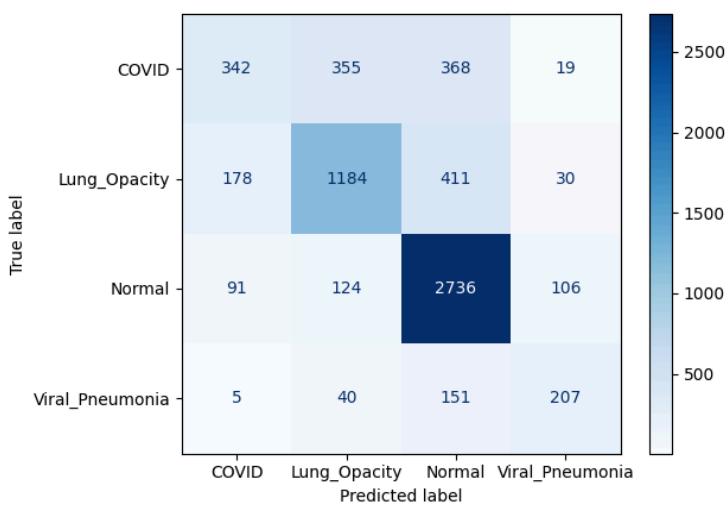
Les modèles choisis étaient pré-entraînés sur le jeu de données imangenet. Les modèles Resnet50 et EfficientB2 ont été testés en appliquant directement les poids issus de cet entraînement. Les masques ont été appliqués en amont et les images ont été pré-processées avec les fonctions ad hoc fournies avec le modèle. Les modèles Resnet50 et EfficientB2 ont été testés avec 70% des données, des tailles de lots de 64, un learning rate de 0.001 et 10 epochs. Les 30% de données restants sont utilisés en validation.

### Modèle Resnet50 (177 couches)

#### Accuracy et loss

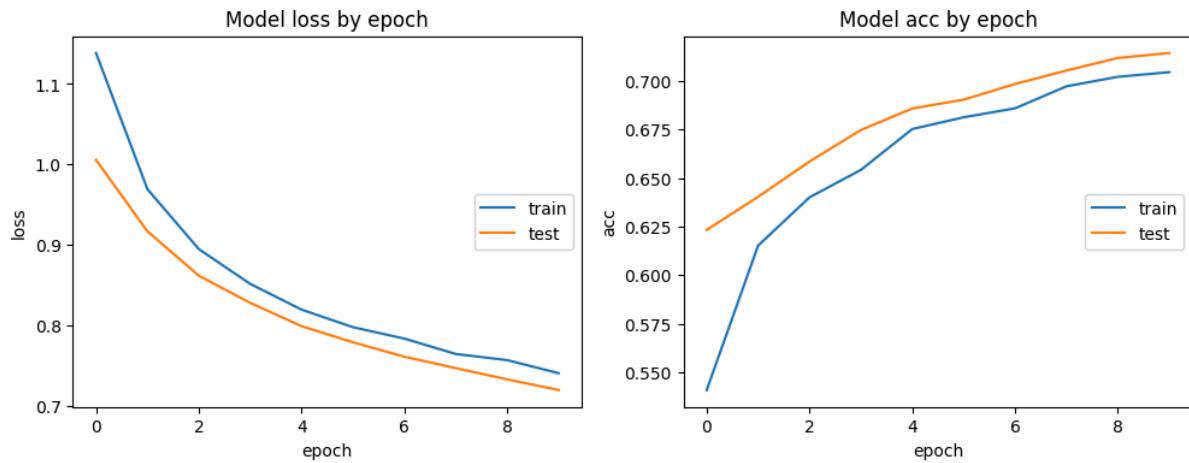


#### Matrice de confusion

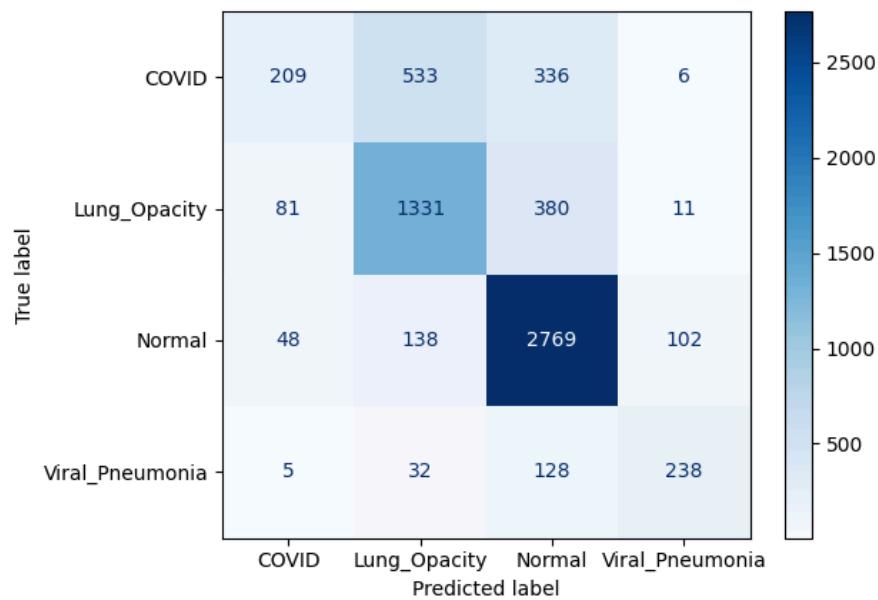


La précision du modèle ResNet50 est de 0.70, mais on observe que les radiographies COVID ne sont bien classées que dans un tiers des cas (recall = 0.33).

## Modèle EfficientNetB2 (342 couches)



La précision obtenue est de 0.72 avec un recall de 0.19 sur la classe COVID.



Les radiographies de la classe COVID sont souvent mal classées avec une confusion forte entre la catégorie COVID et l'opacité pulmonaire. Il faudra sans doute ajuster le poids des classes car les données COVID et Viral\_Pneumonia sont minoritaires dans le dataset.

## Conclusion

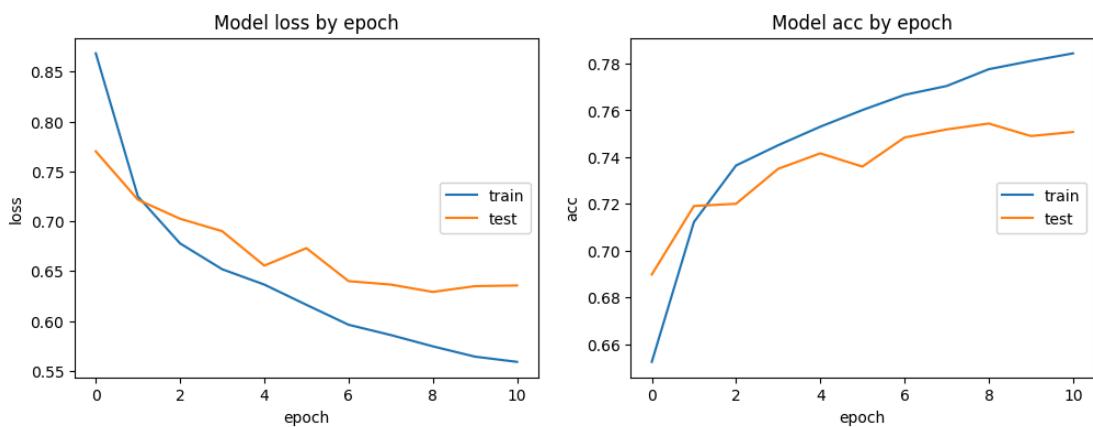
Le modèle ResNet50 conduisant à de meilleurs résultats sur ce premier essai, on choisira ce modèle pour approfondir le choix des paramètres pour le modèle final.

## Augmentation de données, choix du learning rate.

Les données ont été augmentées en utilisant les fonctions de zoom, d'éirement, de décalage vertical et horizontal ainsi que la modification de la luminosité. Cette augmentation a pour objectif d'améliorer la précision du modèle en diversifiant le jeu de données initial ce qui permet de fiabiliser l'analyse et faciliter sa généralisation.

Trois premières valeurs de learning-rate ont été testées : 0.01, 0.001 et 0.0001. On remarque que la précision augmente plus rapidement avec un learning rate de 0.01 sans pour autant empêcher la convergence de la loss.

On choisit donc de démarrer l'entraînement du modèle avec un learning-rate de 0.01 et de le diminuer ensuite au fur et à mesure des epochs. On intègre également un stop pour arrêter l'entraînement si la loss dans le jeu de données de validation varie de moins de 0.005. Avec ces modifications, l'accuracy est de 0.75 et le recall sur la catégorie COVID de 0.49.



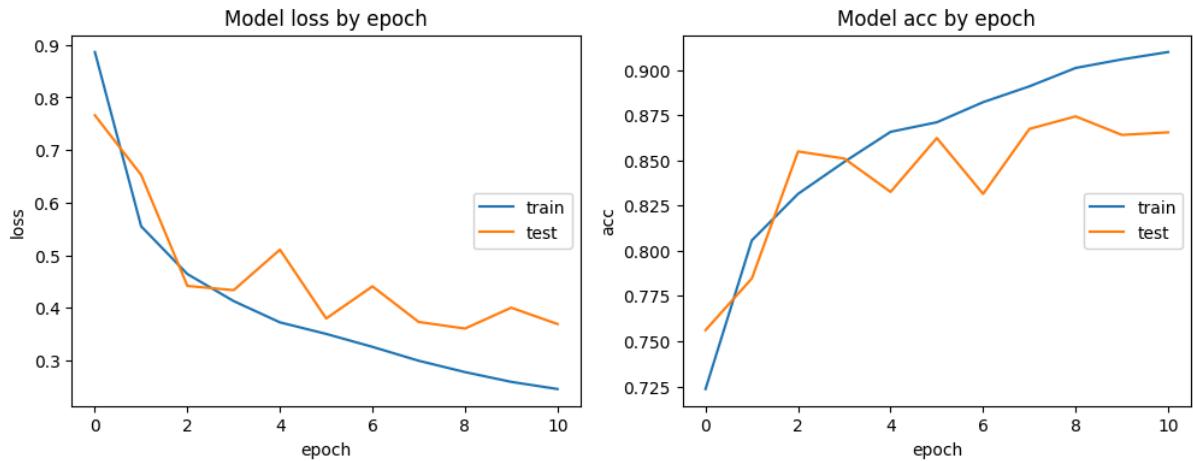
## Fine tuning : Dégel des dernières couches

Trois options ont été testées pour adapter plus finement le modèle à nos données. Les poids des dernières couches ont été dégelées de façon progressive d'abord les sept dernières, puis les quinze dernières et enfin les 30 dernières. Dans ce cas de figure, on cherche à maximiser la prédiction on sera donc plus strict sur l'arrêt anticipé de l'entraînement en utilisant un delta de variation de 0.001 sur la précision en validation.

On présente ici les résultats du degel des trente dernières couches qui donne la meilleure précision (0.86 de val\_accuracy, vs 0.83 vs 0.77).

## Loss et accuracy

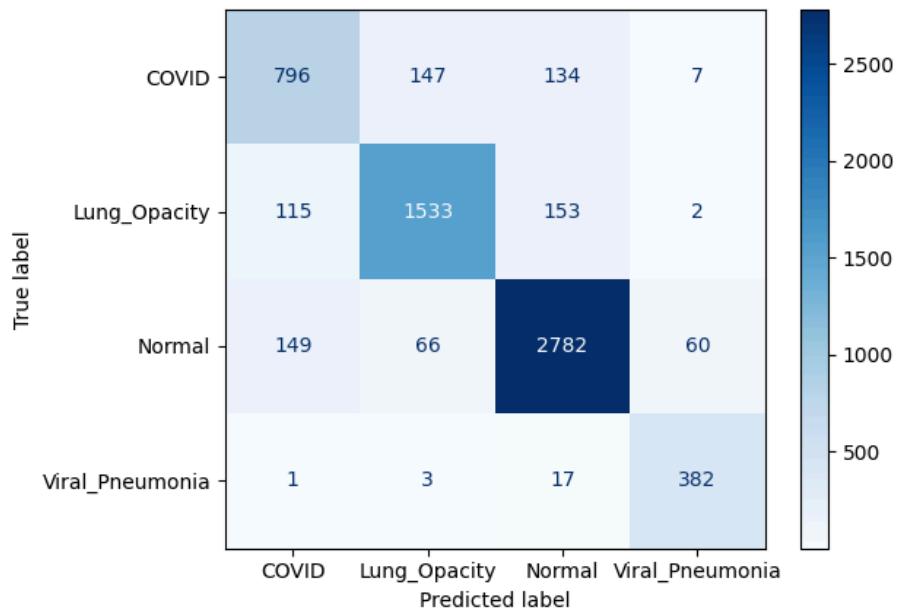
La fonction de perte et la précision sont affichées pour le jeu d'entraînement et de validation sont affichés dans le graphique ci-dessous.



On remarque un léger overfitting puisqu'à partir de l'epoch 7 la précision continue d'augmenter sur le jeu d'entraînement sans impact visible sur le jeu de validation.

## Matrice de confusion

La matrice de confusion sur le jeu de données test est la suivante :



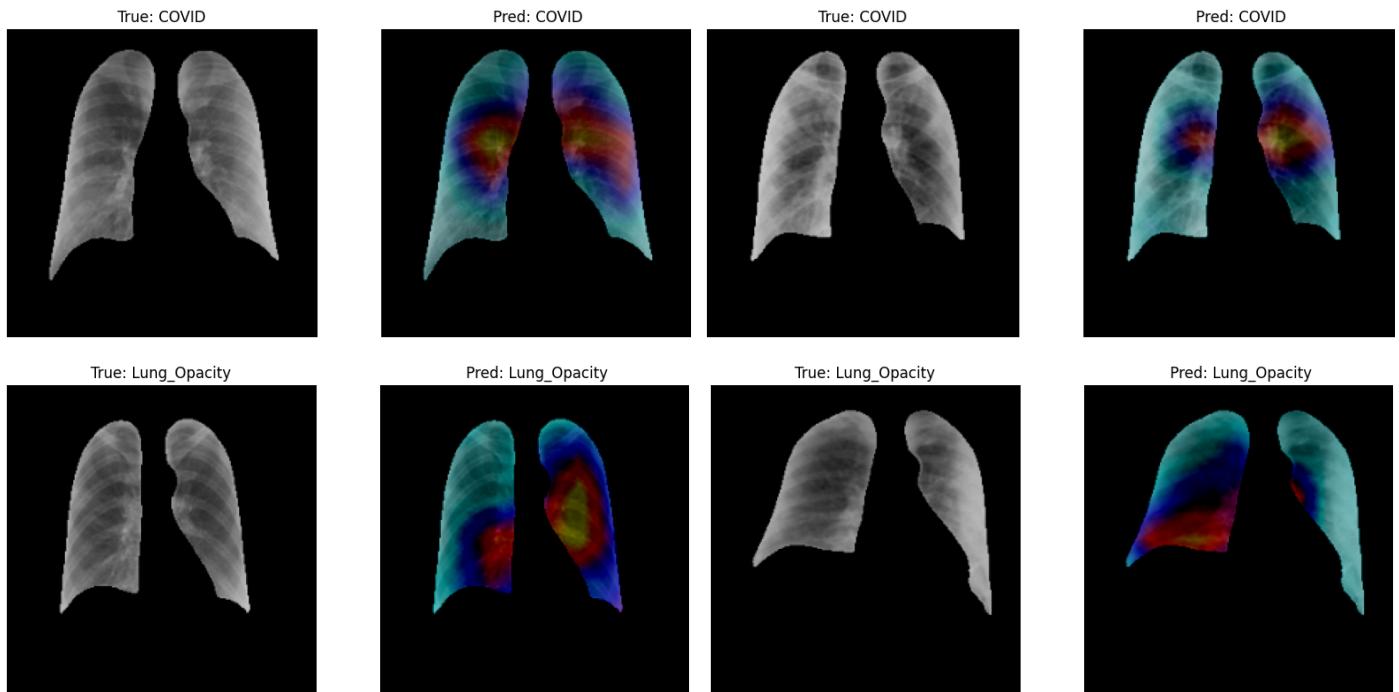
## Conclusion

On remarque qu'il y a encore une légère confusion entre les labels COVID et Lung Opacity mais elle est atténuée par rapport au modèle précédent.

La précision pour la catégorie d'intérêt COVID est de 0.75 avec un recall de 0.73. L'accuracy macro est de 0.85. On remarque également que plus le nombre de couches dégelées est important, plus la précision du modèle augmente. Il semble utile d'augmenter le nombre de couches dégelées.

## Visualiser les régions utiles à la prédiction avec GradCam

Les images ci-dessous permettent de superposer les régions qui ont été utilisées pour la prédiction et l'image initiale, y sont également indiqués les labels réels et prédits.



On retrouve ici la difficulté de repérer pour un non-spécialiste, les zones de l'image les plus pertinentes pour poser le diagnostic.

## Conclusion

Malgré une précision élevée, il persiste une confusion entre l'opacité pulmonaire et les cas de COVID. La précision est également plus faible pour ces deux catégories. Il semble nécessaire pour améliorer la fiabilité du modèle d'utiliser des modèles plus complexes.

## Utilisation de modèles plus avancés

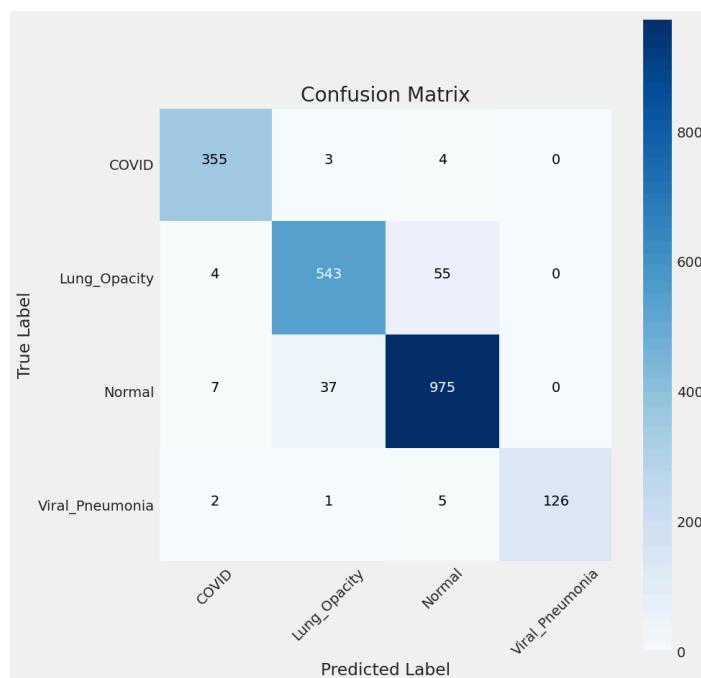
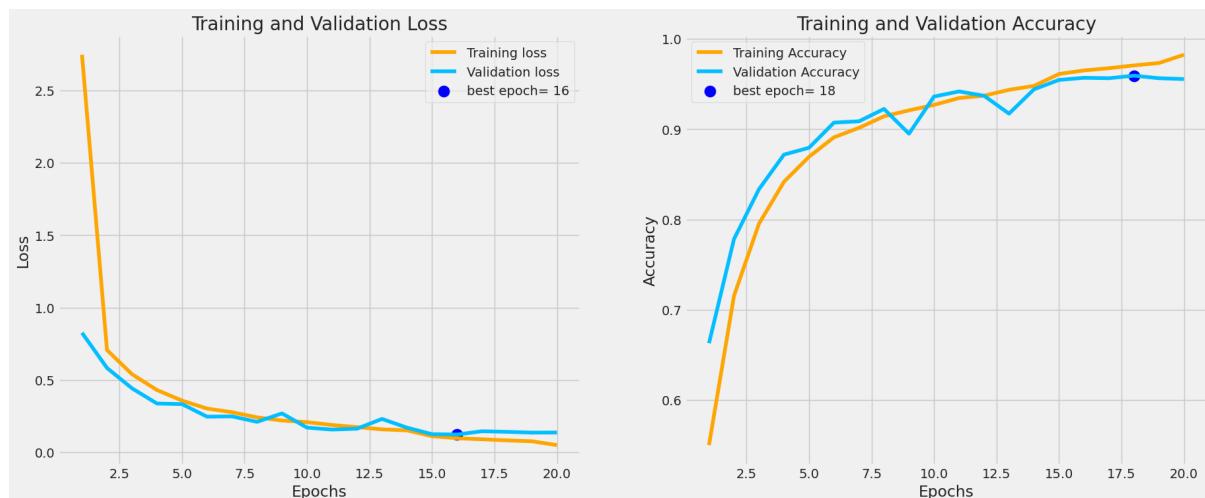
### Choix du modèle

La démarche étant satisfaisante, d'autres modèles ont été testés ensuite pour déterminer si un modèle plus adapté à notre sujet ou plus récent et dont les architectures diffèrent pourrait améliorer nos résultats.

Ce test a été fait sur 80% du jeu de données, 10% des données ont été utilisées pour estimer les paramètres (test) et 10% pour la prédiction finale (validation). La taille des batch a été fixée à 16. On utilise ici les images brutes sans application du masque.

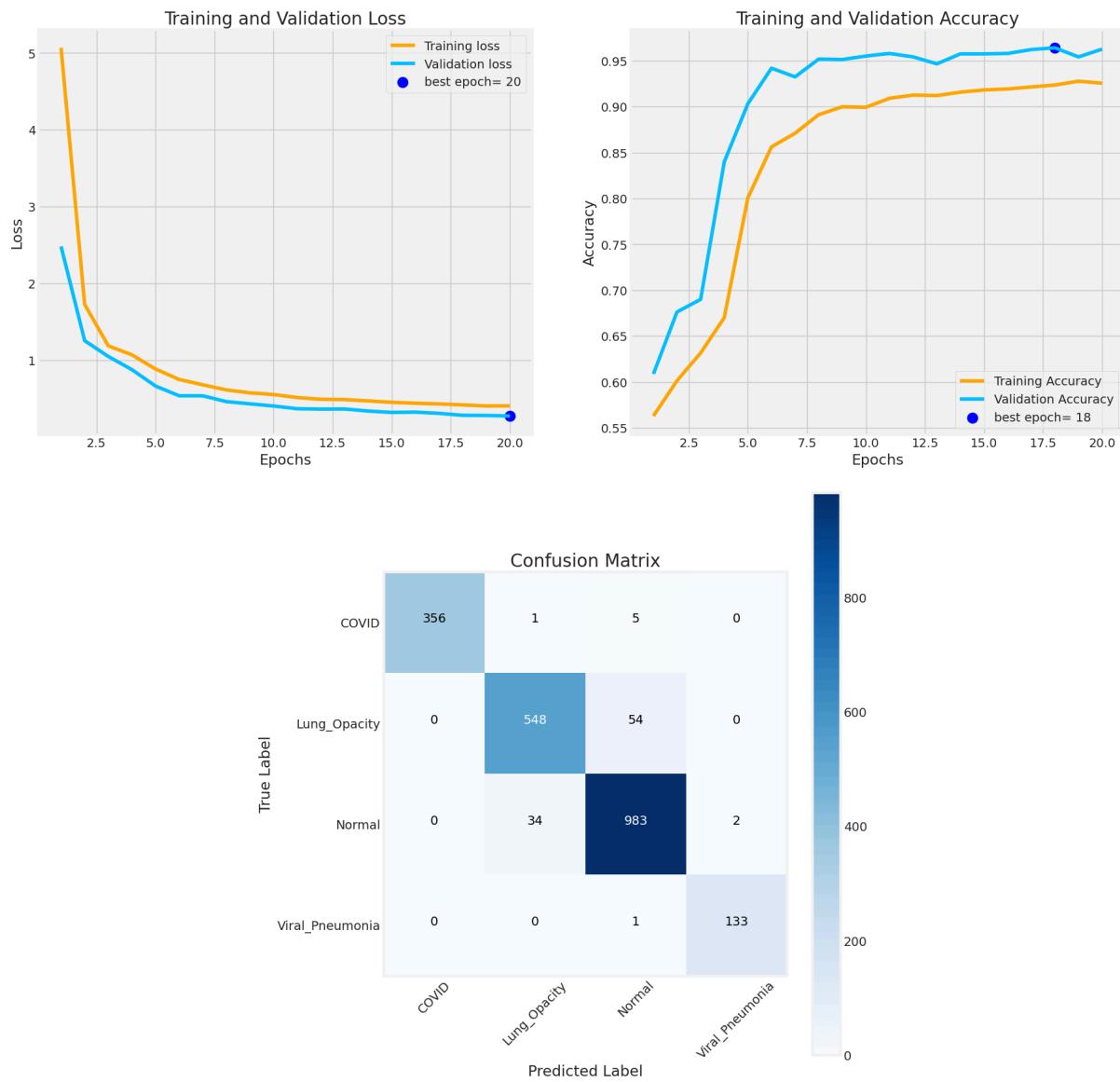
Les modèles testés sont VGG19, EfficientNetB4, EfficientNetB0 et DenseNet169 Resnet50 et Resnet101. Ils ont été testés en dégelant l'ensemble des couches.

## Modèle VGG19



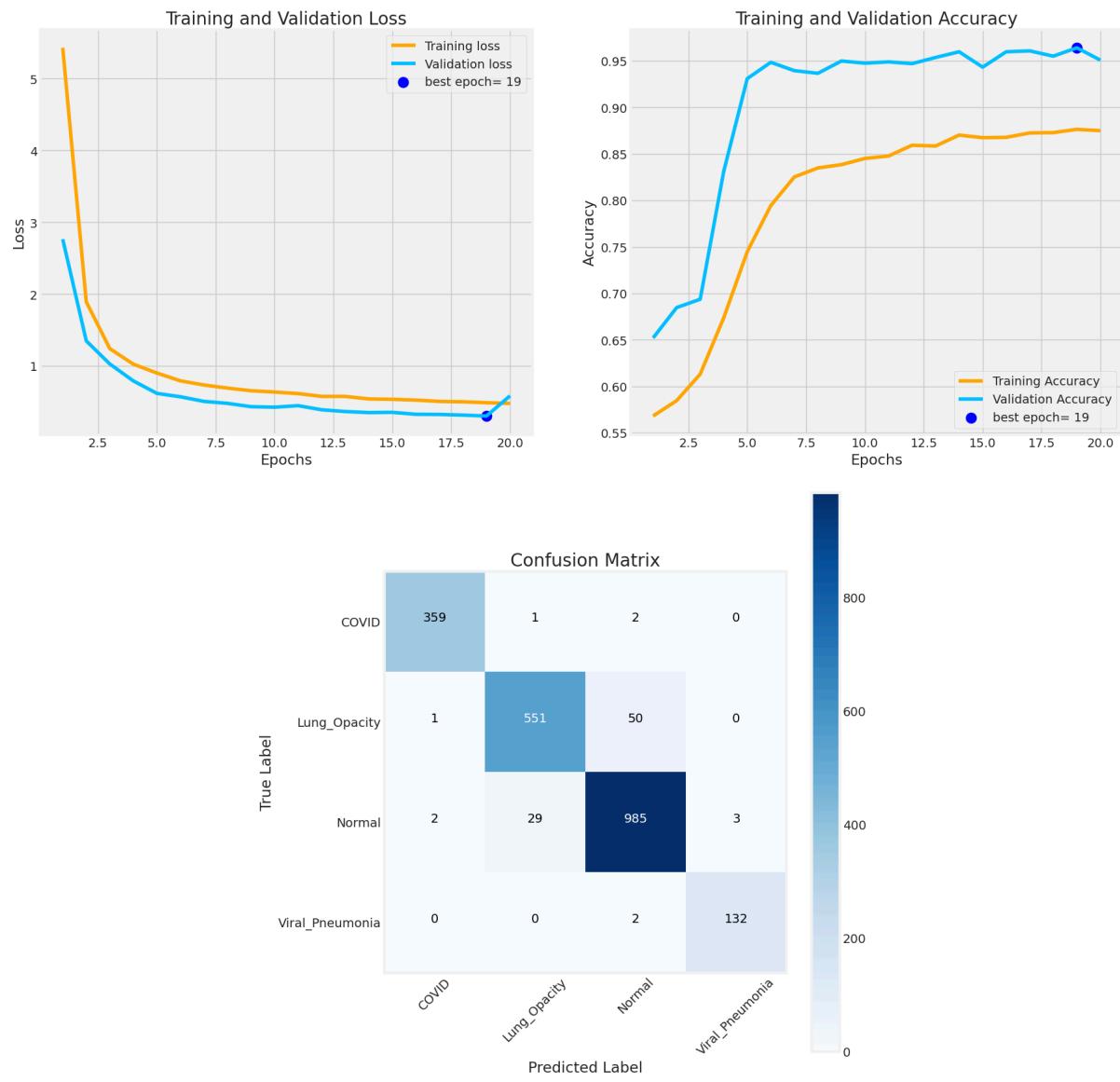
	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
COVID	0.96	0.98	0.97	362
Lung_Opacity	0.93	0.90	0.92	602
Normal	0.94	0.96	0.95	1019
Viral_Pneumonia	1.00	0.94	0.97	134
accuracy			0.94	2117
macro avg	0.96	0.94	0.95	2117
weighted avg	0.94	0.94	0.94	2117

## Modèle EfficientNetB0



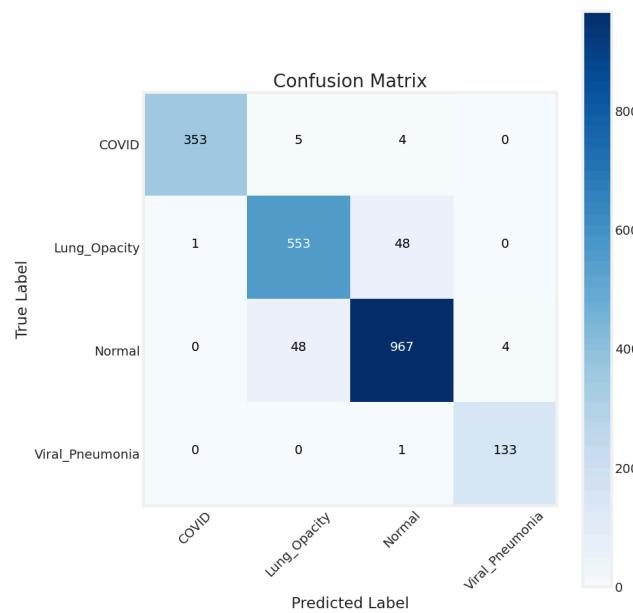
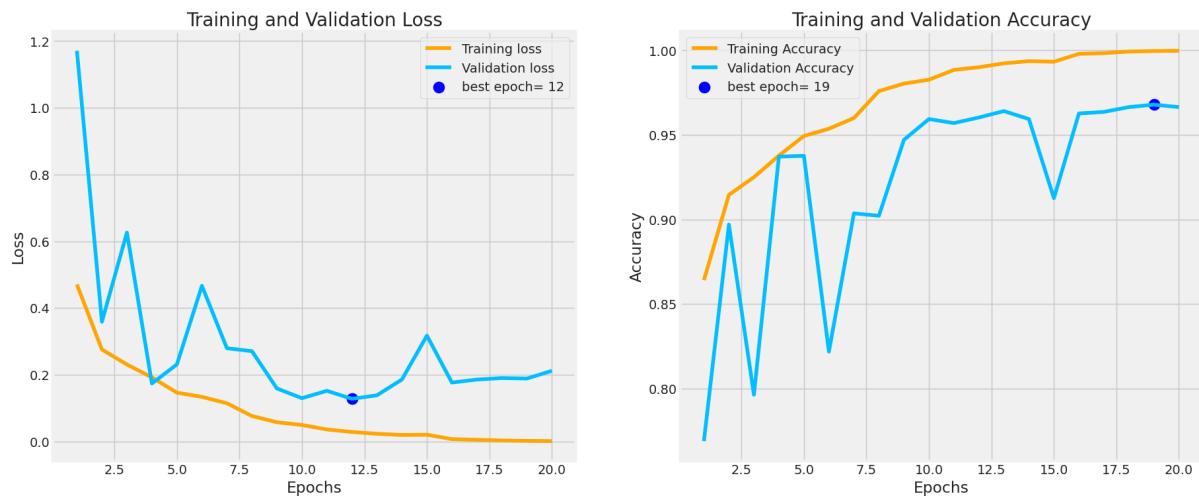
	precision	recall	f1-score	support
COVID	1.00	0.98	0.99	362
Lung_Opacity	0.94	0.91	0.92	602
Normal	0.94	0.96	0.95	1019
Viral_Pneumonia	0.99	0.99	0.99	134
accuracy			0.95	2117
macro avg	0.97	0.96	0.96	2117
weighted avg	0.95	0.95	0.95	2117

## Modèle EfficientNetB4



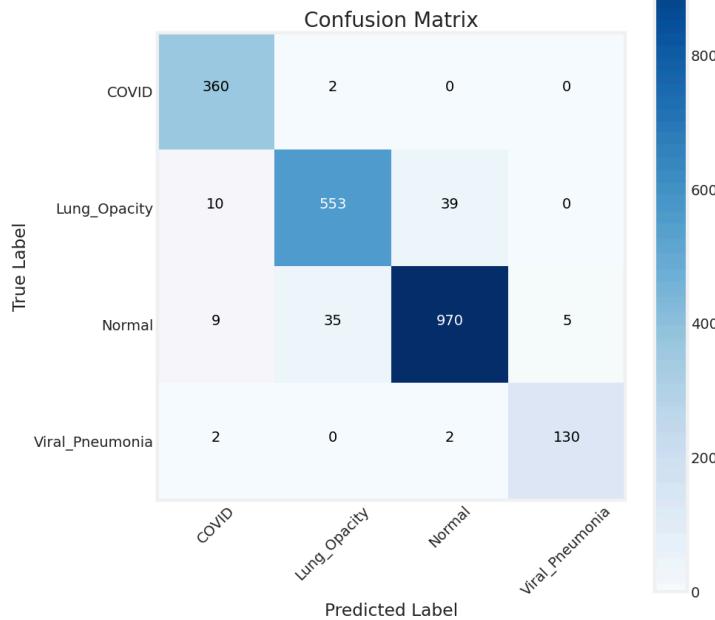
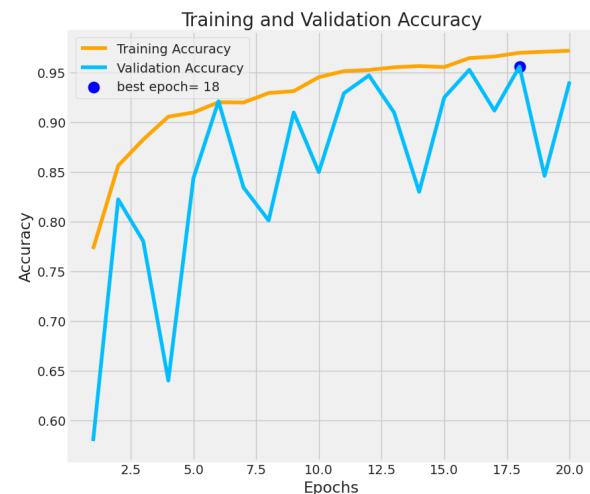
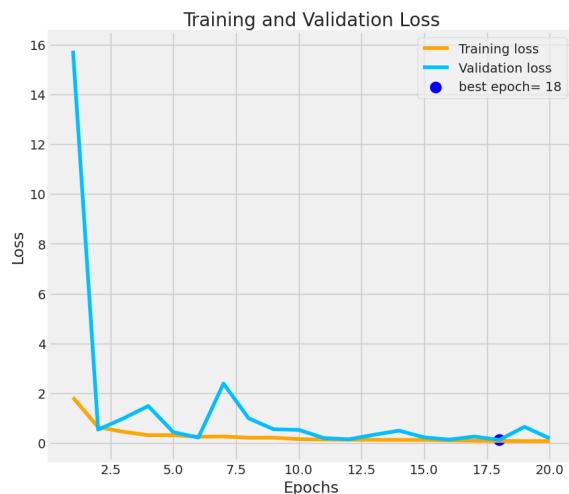
	precision	recall	f1-score	support
COVID	1.00	0.98	0.99	362
Lung_Opacity	0.94	0.91	0.92	602
Normal	0.94	0.96	0.95	1019
Viral_Pneumonia	0.99	0.99	0.99	134
accuracy			0.95	2117
macro avg	0.97	0.96	0.96	2117
weighted avg	0.96	0.96	0.96	2117

## Modèle DenseNet169



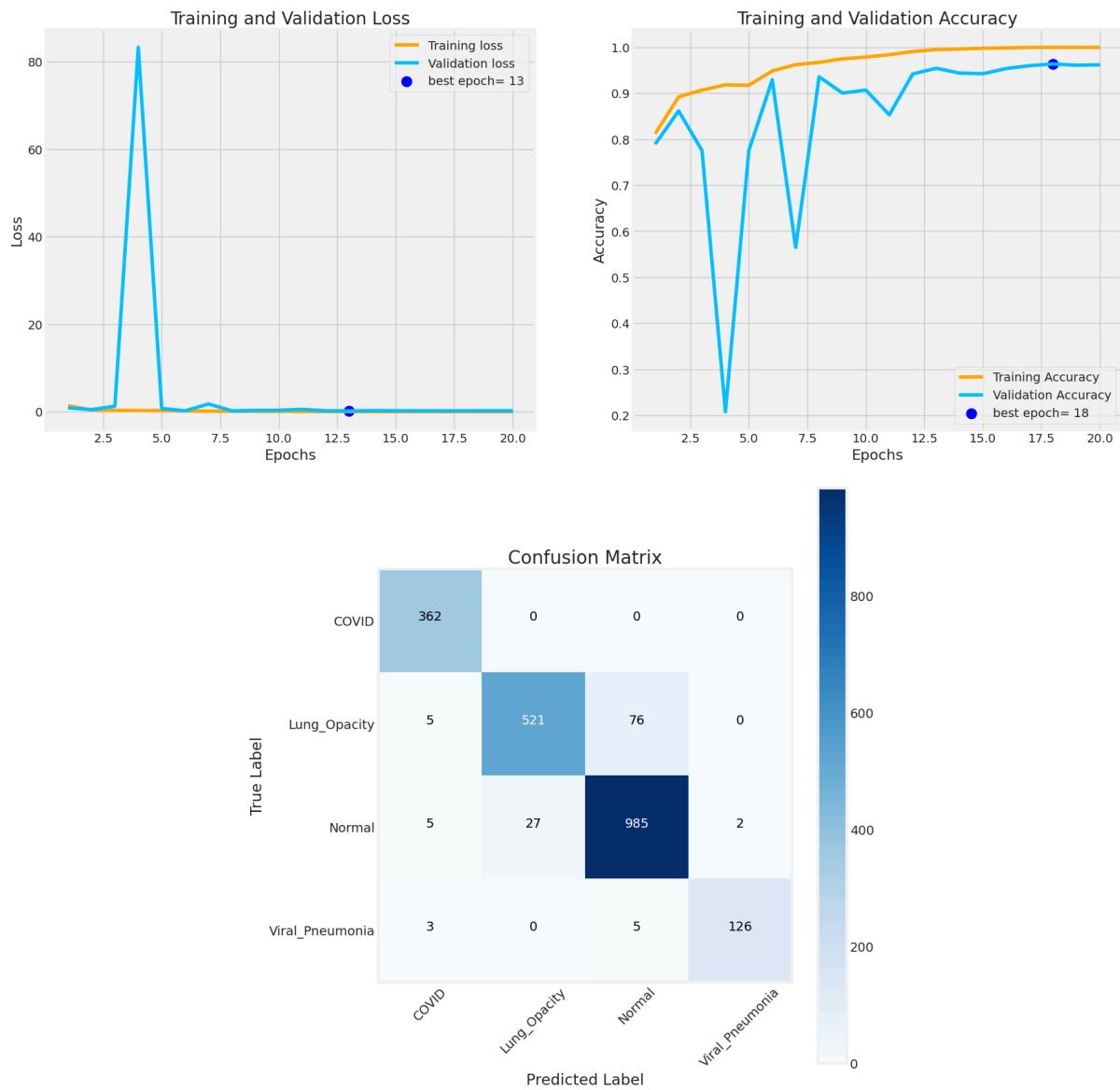
	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
COVID	1.00	0.98	0.99	362
Lung_Opacity	0.91	0.92	0.92	602
Normal	0.95	0.95	0.95	1019
Viral_Pneumonia	0.97	0.99	0.98	134
accuracy			0.95	2117
macro avg	0.96	0.96	0.96	2117
weighted avg	0.95	0.95	0.95	2117

## Modèle Resnet50



	precision	recall	f1-score	support
COVID	0.94	0.99	0.97	362
Lung_Opacity	0.94	0.92	0.93	602
Normal	0.96	0.95	0.96	1019
Viral_Pneumonia	0.96	0.97	0.97	134
accuracy			0.95	2117
macro avg	0.95	0.96	0.95	2117
weighted avg	0.95	0.95	0.95	2117

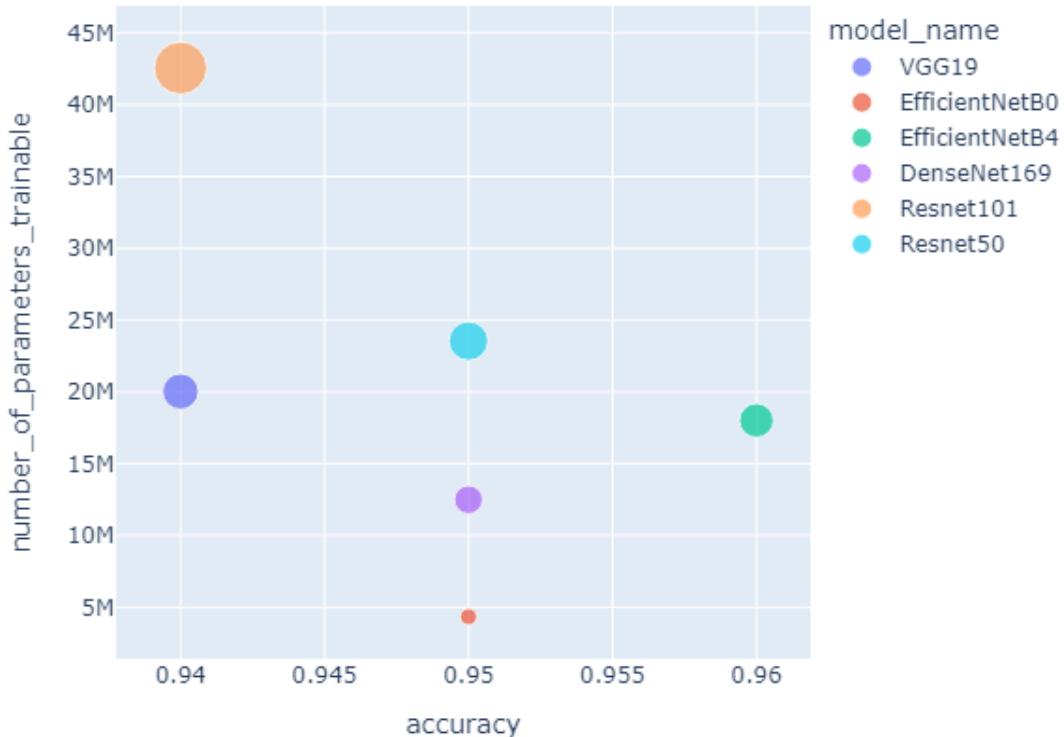
## Modèle Resnet101



	precision	recall	f1-score	support
COVID	0.97	1.00	0.98	362
Lung_Opacity	0.95	0.87	0.91	602
Normal	0.92	0.97	0.94	1019
Viral_Pneumonia	0.98	0.94	0.96	134
accuracy			0.94	2117
macro avg	0.96	0.94	0.95	2117
weighted avg	0.94	0.94	0.94	2117

Les précisions obtenues en ré-estimant l'ensemble des paramètres sont très élevées quel que soit le modèle.

Model comparison: Accuracy VS Number of parameters trainable



Le meilleur modèle est EfficientNetB4 avec une précision globale de 0.96 et de 0.99 pour la catégorie COVID aussi bien en accuracy qu'en recall.

Si on le compare aux autres modèles un EfficientNetB0 serait aussi un candidat très intéressant pour des tests plus rapides mais donnant aussi de bon résultats. Il contient 4 fois moins de paramètres pour une précision inférieure de 0.01 par rapport à EfficientNetB4.

La meilleure précision du modèle où l'ensemble des couches ont été ré-entraînés peut s'expliquer par la spécificité des données de radiographies par rapport aux images qui ont été utilisées dans image\_net.

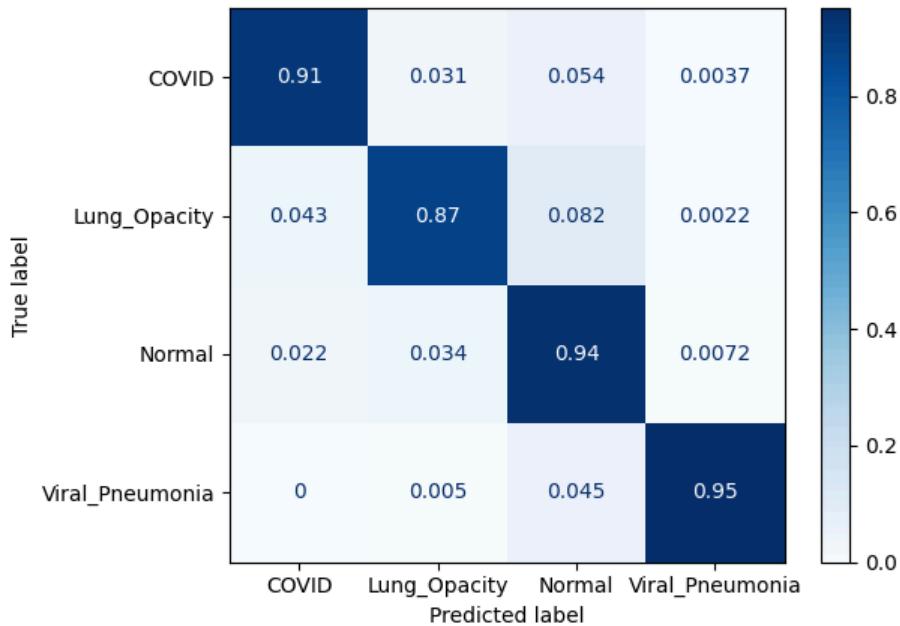
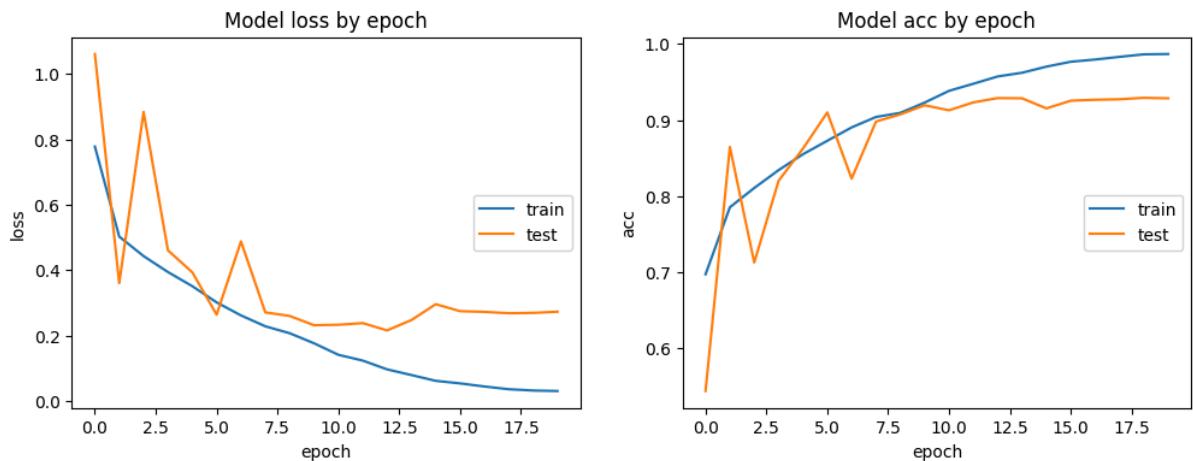
## Choix de l'optimiseur

Les tests des modèles précédents ont été faits avec l'algorithme d'optimisation Adamax et une valeur initiale de learning rate de 0.001.

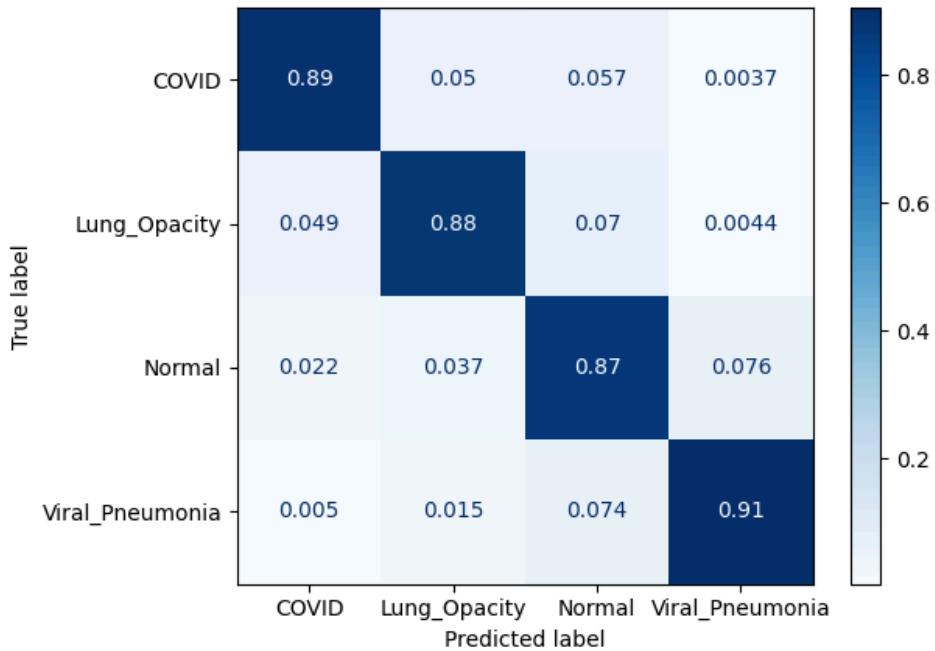
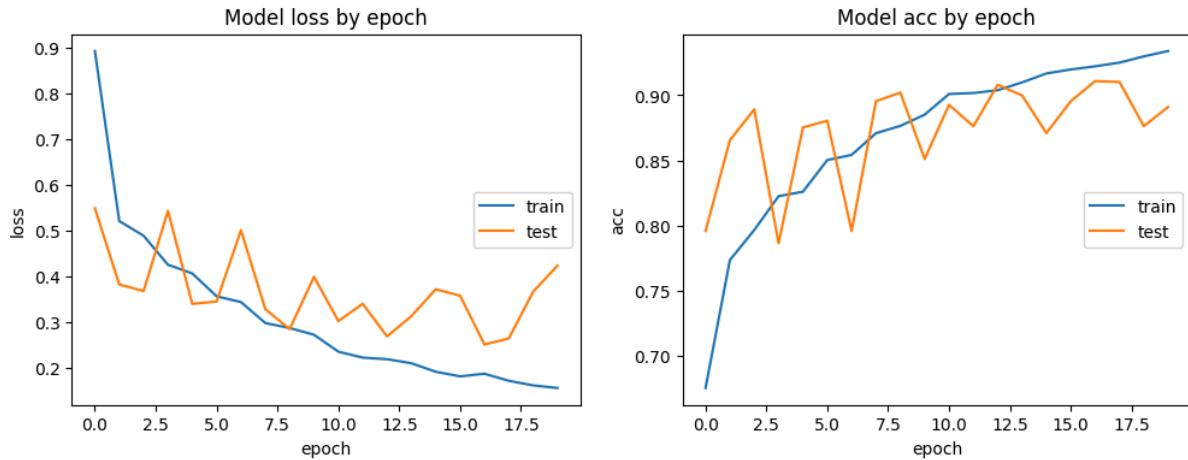
Deux optimiseurs différents ont également été testé sur le modèle EfficientNetB4 : CosineDecay et ExponentialDecay, ils permettent d'ajuster le taux d'apprentissage au fur et à mesure des epoch de façon sinusoïdale ou exponentiel.

Les graphiques d'évolution de la fonction de perte, de la précision ainsi que les matrices de confusion sont présentés ci-dessous pour les deux optimiseurs :

### CosineDecay



## ExponentialDecay



## Conclusion

Dans les deux cas testés, les modèles présentent une bonne précision avec 0.92 pour le CosineDecay et 0.88 pour l'ExponentialDecay. L'évolution de la fonction de perte sur la validation est cependant très fluctuante contrairement à ce qui avait été observé sur le modèle avec uniquement l'optimiseur Adamax. Nous choisissons donc de conserver uniquement l'optimiseur Adamax.

Il pourrait être intéressant d'approfondir la stratégie de réduction du taux d'apprentissage en modifiant le paramétrage (decay\_rate) ou en ne commençant la réduction qu'après avoir laissé passé les epochs initiaux (warm-up). Cela nous permettrait d'étudier s'il est encore possible d'optimiser le modèle..

# Modèle définitif

## Paramètres d'entraînement

Le modèle EfficientNetB4 choisit à donc été entraîné avec les paramètres suivants:

Parameter	Value
batch size	16
epoch	40
Learning rate initial	0.001
patience ( <i>nb epoch before adjusting lr</i> )	3
threshold ( <i>if train accuracy is &lt; threshold adjust monitor accuracy, else monitor validation loss</i> )	0.9
factor	0.5
optimizer	Adamax
loss function	Categorical Crossentropy
Training dataset ratio	0.8
Test dataset ratio	0.1
Evaluation dataset ratio	0.1
Image Dataset size	21165
Image Dataset type	Masked

Les images sont masquées pour garantir que l'apprentissage n'utilise que des informations concernant les poumons.

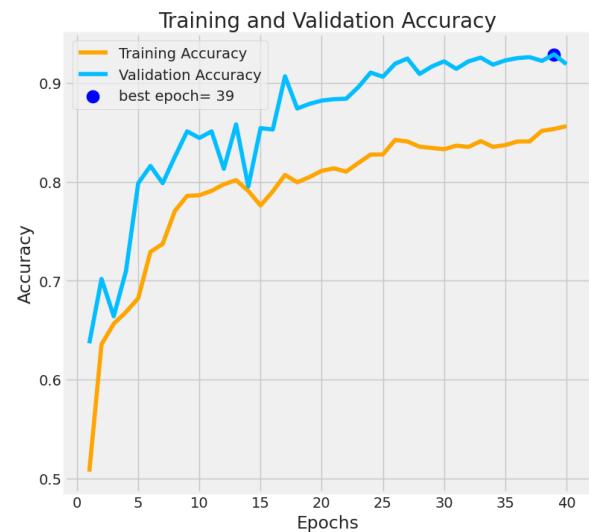
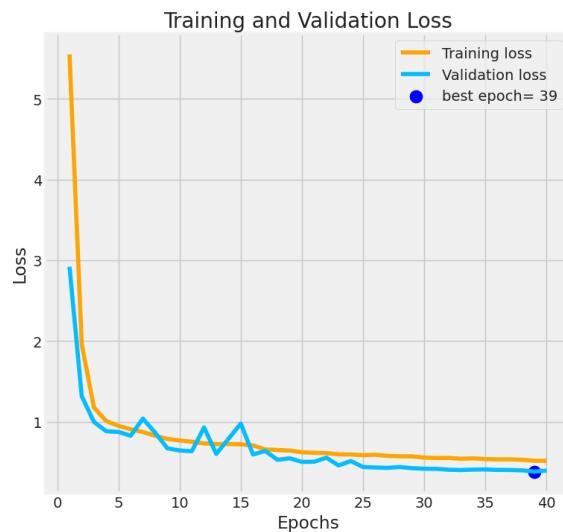
L'entraînement sur données non masquées donne d'aussi bons voir de meilleurs résultats mais on choisit de travailler sur les données masquées pour s'assurer que les prédictions sont uniquement basées sur les poumons et pas des éléments du background. Ce biais lié à des informations de background pourrait être d'autant plus important dans notre cas où une confusion existe entre l'étiquetage des images et l'origine des données.

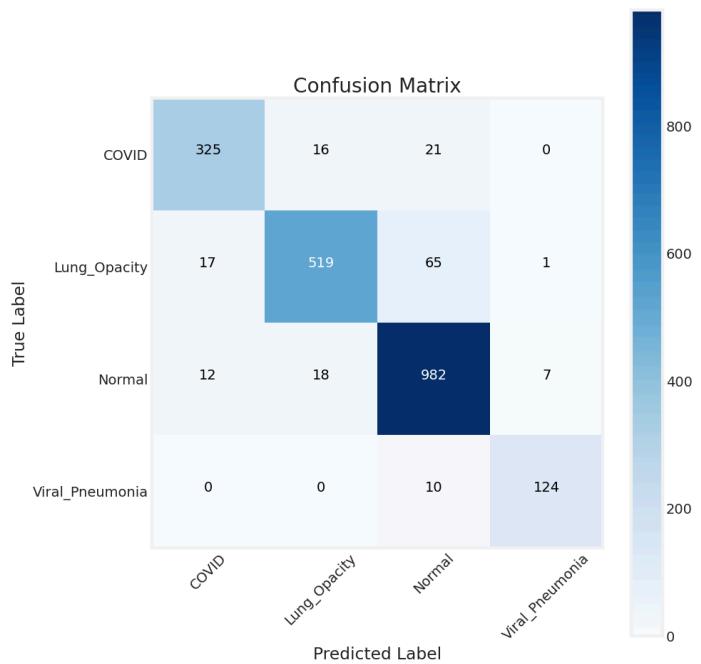
L'augmentation des données a été réalisée avec les paramètres suivants:

Parameter	Value
horizontal_flip	True
width_shift_range	0.1
height_shift_range	0.1
rotation_range	5
fill_mode	nearest
brightness_range	[0.8,1.2]

Les données sont augmentées de manière à simuler des radiographies sur lesquelles les poumons ne seraient pas totalement centrés ou réalisées de dos afin de mieux généraliser le problème pour de futures images venant d'autres sources.

## Résultat de l'entraînement





	precision	recall	f1-score	support
COVID	0.93	0.90	0.91	362
Lung_Opacity	0.93	0.84	0.88	602
Normal	0.90	0.97	0.94	1019
Viral_Pneumonia	0.98	0.94	0.96	134
accuracy			<b>0.92</b>	2117
macro avg	0.94	0.91	0.92	2117
weighted avg	0.92	0.92	0.92	2117

## Conclusion

Le modèle EfficientNetB4 fine-tuned à notre jeu de données nous permet d'obtenir 92% d'accuracy. Une précision plus importante pourrait probablement être obtenue avec des modèles de type transformers qui sont les plus performants aujourd'hui.

Un test a été réalisé sur un sous-jeu de données de 2000 images avec le modèle VIT mais la demande en ressources informatiques étaient trop importantes pour être réalisée sur jeu de données complets (Annexe 1).

# Généralisation du modèle

Nous cherchons ici à savoir dans quelle mesure notre modèle est généralisable à un jeu de données plus important.

Le modèle ayant été entraîné sur des données masquées afin de se focaliser sur les poumons, il faut donc pouvoir générer des masques sur une nouvelle image.

## Prédiction de masques : modèle de segmentation

Afin de générer un masque à partir d'une base de données existante, nous utilisons un modèle de segmentation. A la différence des modèles classiques, on cherche ici à attribuer une valeur pour chacun des pixels de l'image et non pas à l'image dans sa globalité.

### Données utilisées pour le modèle de segmentation

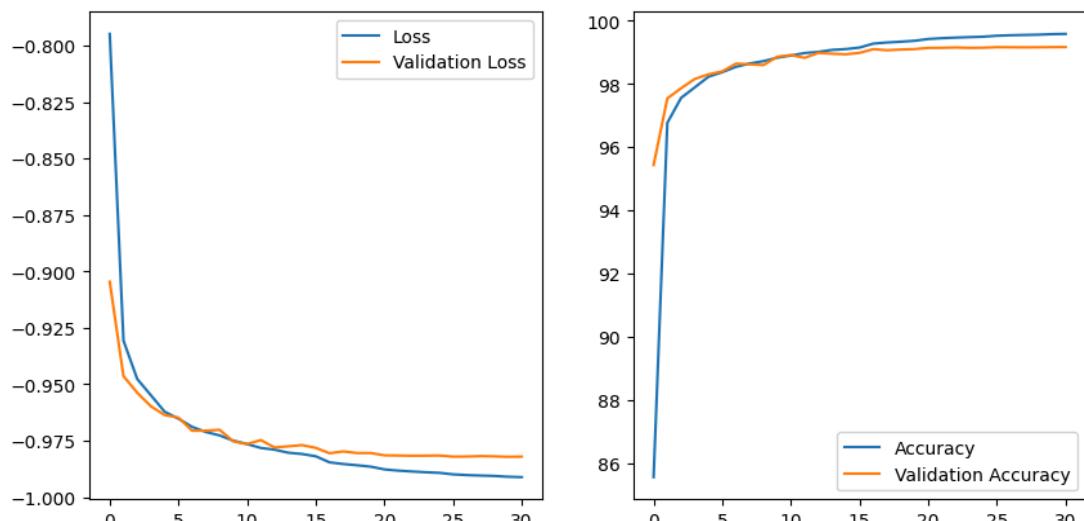
Pour répondre à cette question, un modèle de segmentation a été implémenté. En raison de limitation de mémoire et de temps de calcul, le modèle a été appliqué à un jeu de données contenant au maximum 3000 échantillons de chaque classe soit un total de 8690 images. L'entraînement a été fait sur 80% du jeu de données, le paramétrage sur 10% des données et la validation sur 10% également.

Les variables explicatives sont ici les images en format 256\*256 et on cherche à prédire les masques. Aucune augmentation de données n'a été réalisée pour ce modèle.

Il aurait été possible de diminuer la résolution des images afin de pouvoir étudier un jeu de données plus important mais cela n'a pas été retenu car le reste des travaux s'appuie sur des images en 256 \* 256. Les valeurs des images ont été centrées, réduites autour de 127.

### Modèle utilisé pour la segmentation

On utilise ici, un modèle de type Unet avec 10 blocs et 27 couches pour un total de 7,8 millions de paramètres. Pour des raisons de ressources informatiques, la taille des lots a été réduite à 8. Le learning\_rate initial est de 0.0001, il est divisé par 2 à chaque plateau. Le modèle est entraîné sur 50 epochs avec un arrêt en cas de stagnation de la fonction de perte pour avec la fonction de perte "dice loss" et "binary\_accuracy". Le modèle a été entraîné sur 31 epochs et s'est arrêté pour cause de stagnation de la fonction de perte.



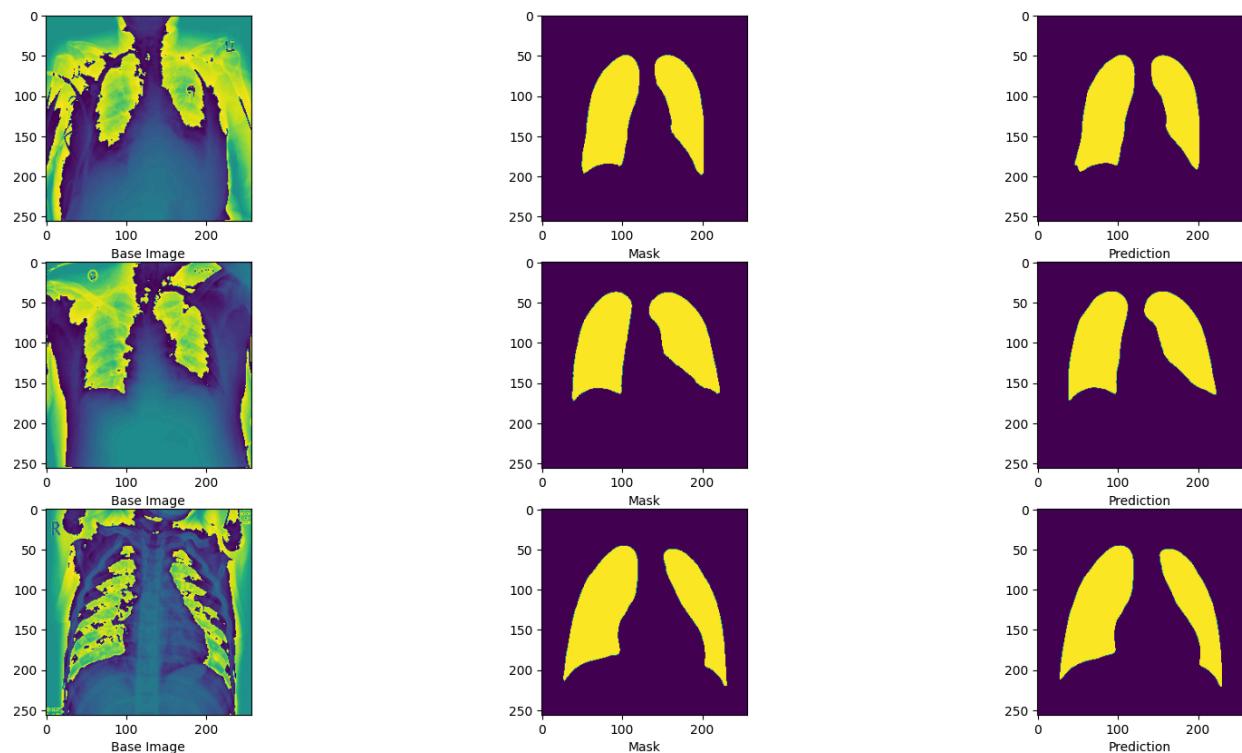
On observe que le modèle permet d'atteindre une bonne précision à la fois dans le jeu de données d'entraînement que de paramétrage.

## Performances du modèle

Les métriques sur le jeu de données d'entraînement, de paramétrage et de validation sont données dans le tableau ci-dessous. On confirme les bons résultats du modèle et ce même sur un jeu de données qui n'a pas été vu lors de l'entraînement.

Jeu de données	Accuracy	Dice coefficient	Dice Loss
Entrainement* (*réduit à 1000 échantillons)	0,9958	0,9911	0,0089
Paramétrage (783 échantillons)	0,9917	0,9822	0,0178
Validation (869 échantillons)	0,9918	0,9822	0,0178

L'image ci-dessous représente un exemple pour trois images de jeu de données de validation avec à gauche l'image initiale, au centre le masque réel, à droite le masque prédict.



Ici, encore on observe les bons résultats du modèle de segmentation ce qui prouve qu'il sera possible de créer des masques sur de nouvelles données afin d'évaluer notre modèle de classification sur un jeu de données externe.

## Prédiction du diagnostic COVID : modèle de classification

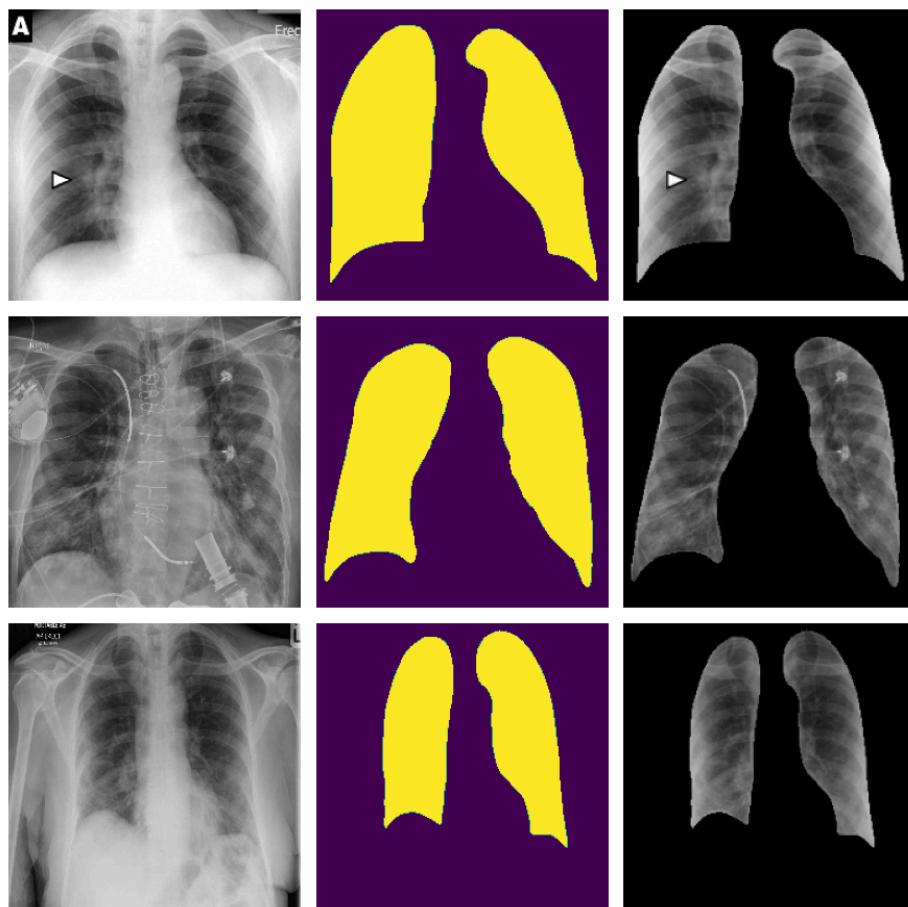
Pour cette partie on se base sur le jeu de données COVID\_NET, une initiative visant à rassembler un maximum de radiographies pulmonaires liées au COVID. Le jeu de données est accessible au lien suivant : [COVIDx CXR-4 Kaggle](#)

### Données utilisées

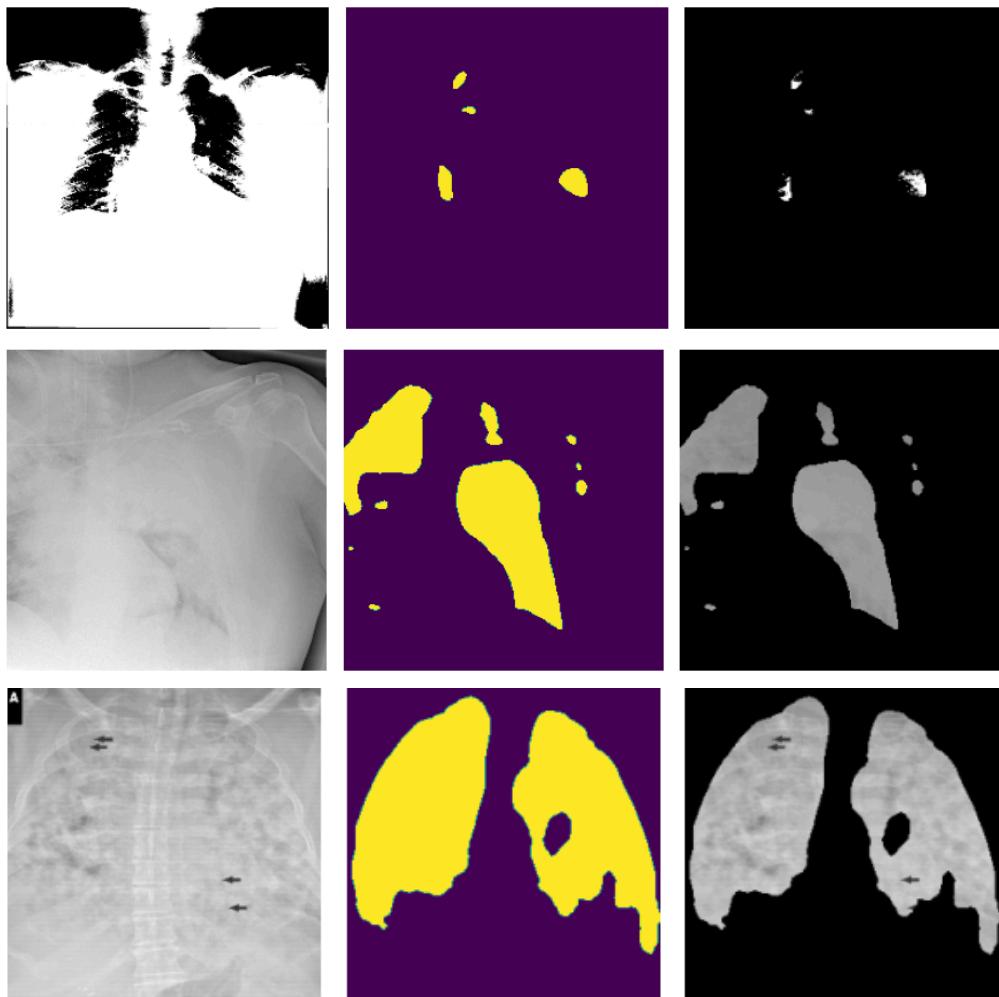
Ce jeu de données contient plus de 80 000 images dont une partie sont redondantes avec notre jeu de données initial. Pour s'assurer que ce jeu de données de validation soit bien externe à notre dataset initial, nous conserverons uniquement les sources “actmed”, “cohen” et “fig1”. Les données sont réparties en deux classes selon la positivité ou non au COVID. Au total 723 images et leur labels sont conservés.

### Masquage des images

Les images n'étant pas masquées, la première étape consiste à créer un masque à partir de modèle de segmentation présenté précédemment puis à l'appliquer. Les masques et les images masquées sont présentés ci-dessous pour trois exemples. Visuellement, les résultats semblent satisfaisants ce qui conforte les bons résultats du modèle de segmentation.



Dans certains cas particuliers avec une image de départ de mauvaise qualité ou du moins très différentes des données initiales, le modèle de segmentation ne semble pas fonctionner ce qui conduit à des masques inappropriés.



## Classification des images

### Modèle RESNET50

Après avoir pré-processé les images, on applique le modèle développé à partir de RESNET50. On obtient la matrice de confusion suivante (normalisée par ligne).

	COVID	Lung_Opacity	Normal	Viral Pneumonia
Negatif	42,1 %	23,5 %	33,8 %	0,6 %
Positif	72,1 %	12,3 %	14,7 %	0,9 %

On remarque que 72% des images issues de patients positifs sont prédites comme COVID, le reste se classe à 12% en opacité pulmonaire et 14,7% normal. Le taux de faux-positif est élevé avec 42% de radiographies de patients négatifs prédites comme COVID, 23% comme présentant une opacité pulmonaire et uniquement 34% comme normal. données.

## Modèle EfficientNetB4

Avec le modèle le plus performant, sur le jeu de données de validation interne les résultats sont les suivants :

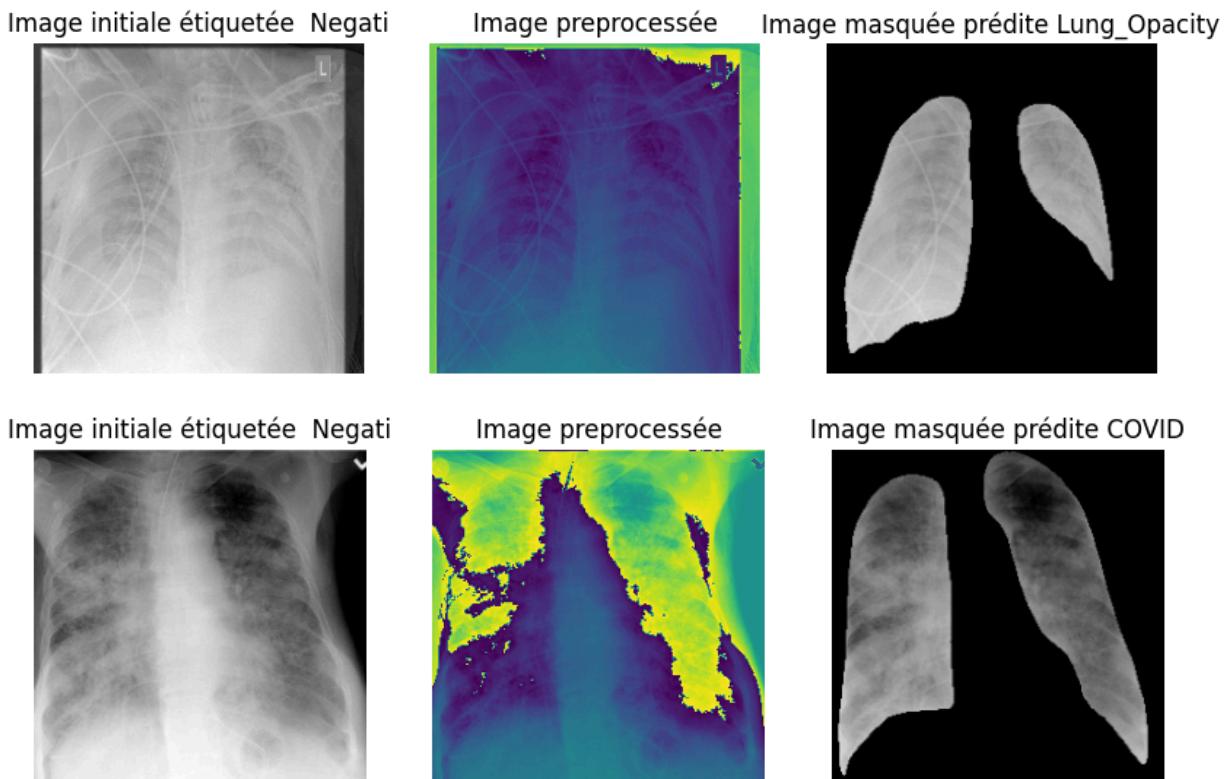
	COVID	Lung_Opacity	Normal	Viral_Pneumonia
Negatif	62,7 %	19,7 %	17,6 %	0,0 %
Positif	83,1 %	11,3 %	5,3 %	0,3 %

Conformément à ce qui a été observé sur le jeu de données initial, les performances du modèle EfficientNetB4 semblent meilleures avec 83% d'images de patients positifs prédites comme COVID et uniquement 5% classés comme normales.

En revanche, cela se fait au prix d'un plus grand nombre de faux positifs : des images de patients négatifs classés comme COVID dans 63% des cas et Normal dans 17%.

### Analyse des erreurs de classifications

Ci-après deux exemples d'images pour lesquelles les prédictions sont fausses.



Sans l'expertise d'un radiologue il est difficile de juger mais il semble toutefois que les nouvelles images soient d'assez mauvaise qualité ce qui pourrait expliquer au moins en partie les mauvais résultats observés. Il faudrait également s'assurer que les images provenant de patients négatifs au COVID correspondent réellement à des radiographies de patients sans affection et non pas souffrant d'une autre pathologie.

## Conclusion

Cette étude a été réalisée sur un jeu de données composé de 21 165 images réparties en quatre catégories. L'objectif principal était de développer un modèle de deep learning pour faciliter le diagnostic des affections pulmonaires, en particulier du COVID-19. Les données ont été préalablement traitées à l'aide des masques fournis, puis divisées en trois sous-ensembles : apprentissage (80%), validation (10%) et test (10%). Cette séparation a permis d'entraîner le modèle sur les données d'apprentissage, d'optimiser les paramètres sur le jeu de validation, et de mesurer la performance du modèle sur le jeu de test.

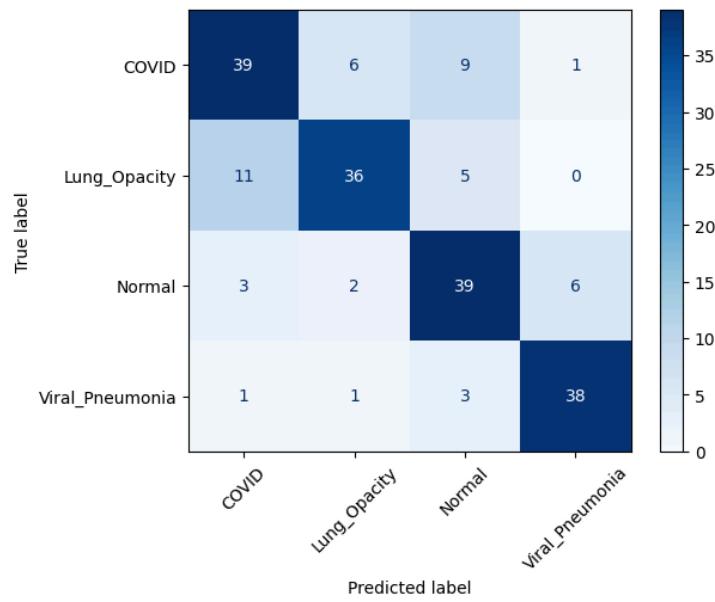
Plusieurs architectures et configurations d'hyperparamètres ont été testées. Le modèle ayant donné les meilleurs résultats est EfficientNetB4, développé en 2019, avec 19 millions de paramètres. Ce modèle a atteint une précision de 0.92, ce qui est très satisfaisant. De plus, aucun problème majeur de performance n'a été observé pour la catégorie COVID, qui, bien que minoritaire, est la plus cruciale du point de vue diagnostic.

Un autre objectif de cette étude était de généraliser le modèle à un jeu de données externe. Après avoir identifié des nouvelles données que nous pourrions utiliser, un modèle de segmentation a été mis en place pour générer des masques pour ces images externes. L'architecture choisie pour cette tâche est U-Net, qui a obtenu une précision supérieure à 0.99, ce qui est très encourageant. Ensuite, le modèle de segmentation a été appliqué aux données externes, suivi du modèle de classification sur ces images après application des masques. Bien que la précision observée soit inférieure à celle obtenue sur le jeu de données interne, il a été constaté une tendance à classer les images de patients positifs au COVID dans la catégorie COVID. Ces résultats doivent toutefois être confirmés sur des images de meilleure qualité et sur des données mieux étiquetées (avec une classification identique à celle présente sur le jeu de données initial).

En conclusion, ce travail démontre l'efficacité du modèle EfficientNetB4 pour le diagnostic du COVID-19 à partir de radiographies pulmonaires, ainsi que l'intérêt du modèle de segmentation pour générer des masques permettant de prétraiter de nouvelles données. Pour généraliser davantage ce travail, il serait pertinent d'affiner encore le modèle, notamment en explorant des modèles de type transformers, qui, bien que gourmands en ressources, semblent prometteurs, ou en augmentant le jeu de données initial à partir de la base COVID-NET utilisée ici pour la validation externe.

## Annexe 1 : Résultat du modèle ViT

Test du model ViT sur 2 epochs et un échantillon de 2000 images, 200 ont été utilisés en validation. Le modèle a été arrêté après 63min d'entraînement



	precision	recall	f1-score	support
COVID	0.72	0.71	0.72	55
Lung_Opacity	0.80	0.69	0.74	52
Normal	0.70	0.78	0.74	50
Viral_Pneumonia	0.84	0.88	0.86	43
accuracy			0.76	200
macro avg	0.77	0.77	0.76	200
weighted avg	0.76	0.76	0.76	200