Projet Analyse Pulmonaire - Rapport de Modélisation

1. Formalisation du problème de classification

1.1 Rappel sur les données

Pour rappel, les données à disposition sont des images radios labellisées selon le type de patients auxquelles elles correspondent. Elles appartiennent à 4 classes :

- Les images Covid ;
- Les images de patients atteints de pneumonie virale ;
- Les images étiquetées *lung opacity* qui correspondent à des patients souffrant d'autres infections pulmonaires.
- Les patients sains, labellisés *normal*

Chaque image est appairée à une image masque qui délimite la zone des poumons sur la radio. La superposition des deux permet de réduire l'image à sa portion signifiante pour le problème cible et d'éliminer le bruit apporté par les autres zones des radios.
L'ensemble des images radio a une résolution de 299 pixels, contre 256 pour les masques.
On dispose d'un peu plus de 21000 observations (couples image-masque-label) dont la moitié provenant de la classe *normal.* Les répartitions des patients malade est largement déséquilibrée avec environ 3000 *Covid*, 6000 *lung opacity* et 1345 *viral pneumonia.*

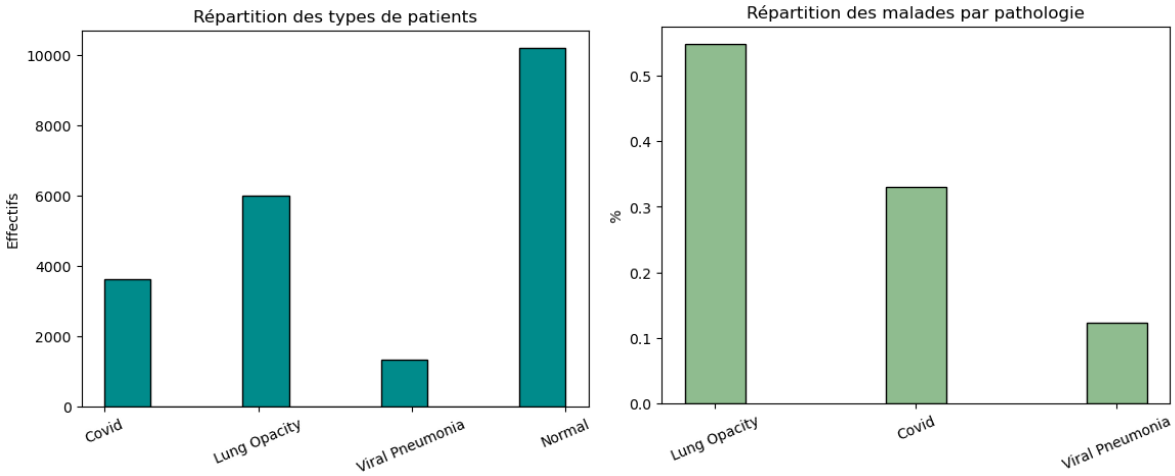


Figure 1 Répartition des données selon le type de maladie

1.2 Problème de classification

L'objectif de cette étude est de développer un modèle de classification permettant d'identifier efficacement les malades du covid. La difficulté et de distinguer précisément d'une part les patients malades des patients sains, et d'autre part les patients covid des patients atteints d'autres pathologies pulmonaires.

Le choix de modélisation que nous avons réalisé est motivé par l'enjeu métier, mais également par des contraintes computationnelles. En effet, il nous est impossible de traiter l'ensemble des images sur nos

machines et les options de serveurs de calculs gratuites telles que Google Collab sont également insuffisantes en termes de mémoire pour gérer ce volume de données.

C'est pourquoi nous choisissons de nous réduire à un sous-ensemble équilibré de données dans la suite. Bien que les premières explorations aient pu porter sur des sous-ensembles de tailles sensiblement plus ou moins grandes, on se limite en fin de course à 7500 observations au total.

Le problème a été investigué sous plusieurs angles :

- Un problème à trois classes distinguant :
    - Les malades Covid ;
    - Les malades atteints de pathologies pulmonaires autres que le Covid (agrégation des classes pneumonie virale et opacité pulmonaire) ;
    - Les patients sains.
- Une classification sur 4 classes.

Les données étant déséquilibrées, nous nous sommes assurés d'obtenir des classes équilibrées lors de l'échantillonnage afin d'éviter d'introduire des biais dans nos modèles et d'assurer la précision de leurs estimations.

Dans un premier temps, des modèles simples de Machine Learning ont été investigués. Face aux limites rencontrées lors de ces premières modélisations, des modèles plus avancés de Deep Learning ont été testés.

1.3 Critères de performances retenus

Les modèles sont évalués selon une métrique donnée. On se concentre d'une part sur le taux de bonnes prédictions du modèle (précision). En effet, ce choix peut être transposé à tous les types de modèle et toutes les répartitions en classes. D'autre part, la classe cible étant le groupe Covid, on s'intéresse également à son rappel. C'est pourquoi le f1-score est aussi une métrique de comparaison des modèles importante.

2. Modèles de Machine Learning

2.1 K-Nearest Neigbors

La méthode des k-plus proches voisins fournit un premier outil simple de classification des images qui repose sur le calcul de la distance entre des vecteurs d'entrées. Dans notre cas, en raison de la taille des images, nos capacités de calcul ne nous permettent pas d'utiliser directement les images comme variables explicatives. C'est pourquoi la première étape consiste en une réduction de la dimension via l'utilisation d'une PCA capturant 90% de la variance des données.

L'algorithme k-NN a été utilisé sur le problème de classification à 3 classes, d'abord sur les images sans masques, puis sur les images masquées.

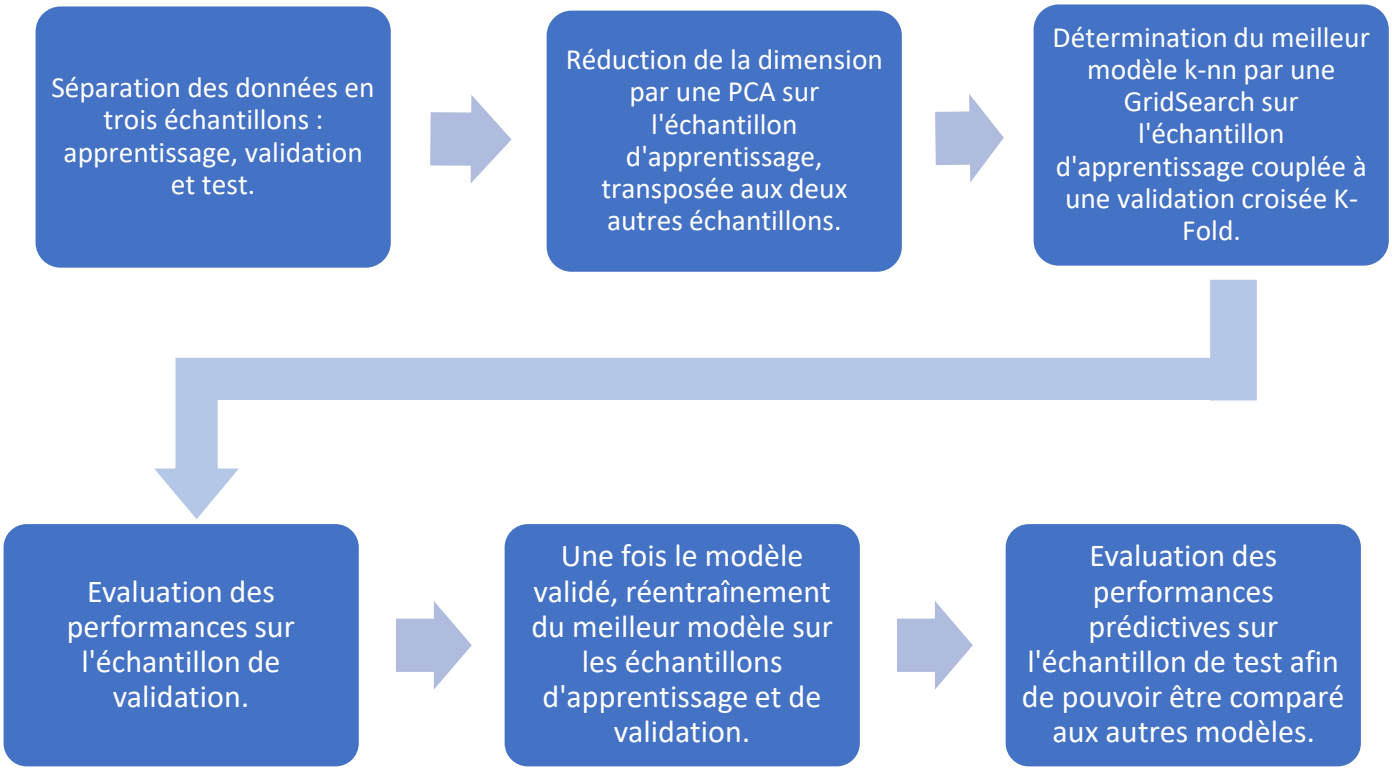


Figure 1 Processus de sélection et évaluation d'un modèle K-NN

Dans les deux cas, la procédure utilisée est la même, telle que décrite par la Figure 1. L'étape de Grid Search est ici importante, les données d'entrées étant les composants obtenus par une PCA, il est difficile de savoir a priori quelle métrique ou quel nombre de voisins sont les plus adaptés.

Les résultats obtenus sont synthétisés dans le tableau ci-dessous :

Type d'images	**Paramètres retenus**	**Performances sur l'échantillon de test**		
		Accuracy	**f1-score multiclasse**	**f1-score pour la classe Covid**
Sans masque (299 x 299)	Metric : minkowski (p=1) Nombre de voisins = 10	0.77	0.77	0.79
Avec Masque (299 x 299)	Metric : cosine Nombre de voisins = 10	0.66	0.65	0.62

Tableau 1 Performances du modèle k-NN

Les performances en prédiction du modèle se dégradent fortement lorsque l'on ajoute les masques aux images. Or les masques permettent de ne traiter que l'information pertinente au problème de classification. Il est donc possible que le modèle sans masque soit biaisé par un surajustement aux données. En effet, on avait par exemple noté durant la phase de visualisation que les niveaux de luminosité moyens dans les groupes Covid étaient inférieurs à ceux des autres classes. Les données n'ayant pas été augmentées à ce stade, cela peut influencer les résultats du modèle et expliquer le meilleur ajustement aux données.

Cette hypothèse est également appuyée par le fait que le f1-score sur la classe covid est le plus élevé des trois classes pour le modèle sans masque, alors que la version avec masque, de même que les autres modèles de Machine Learning entraînés dans les autres sections, mettent en évidence la difficulté relative à identifier cette classe par rapport aux deux autres.

L'algorithme K-NN sur les données masquées donne quant à lui des résultats très moyens, avec seulement une précision de 66% et un f1-score global de 65%, mais qui tombe à 62% pour le groupe Covid.

2.2 SVM

2.2.1 PCA(2) et SVM

Ce modèle de base est le plus simple parmi ceux présentés dans ce rapport. Chaque image est développée en une ligne, ensuite toutes les données sont traitées comme un tableau de données. Son fonctionnement est présenté dans la Figure 3.

Résultats :

Le score de ce modèle simple modèle est de 0.56. Les scores F1 diffèrent pour les 3 classes :

- 0.51 pour la classe 0 (sains) ;
- 0.59 pour la classe 1 (malades non-COVID) ;
- 0.57 pour la classe 2 (COVID).

Ce modèle présente une plus grande difficulté pour les personnes saines, que pour les deux classes de malades. Ceci correspond bien aux données sur lesquelles il a été entraîné (cf. Figure 3). En effet, les points des classes 1 et 2 (différents types de maladie) occupent davantage des régions précises alors que les points de la classe 0 (images saines) sont plus répartis.

2.2.2 PCA(90%) + SVM.

Ce modèle diffère du précédent par le fait de prendre toutes les premières composantes d'ACP pour garder 90% de variance des données, ce qui fait 47 composantes.

Le score de ce modèle simple modèle est de 0.69. Les scores F1 diffèrent pour les 3 classes :

- 0.69 pour la classe 0 (sains) ;
- 0.72 pour la classe 1 (malades non-COVID) ;
- 0.65 pour la classe 2 (COVID).

Les scores F1 de ce modèle mènent à une interprétation assez différente : le modèle a plus de difficulté à distinguer les malades COVID des personnes saines, que pour les autres maladies des poumons.

Le rapport entre scores f1 des différentes classes change après ajout de composantes : le modèle a plus de difficultés à distinguer les malades COVID des personnes saines, que les autres maladies pulmonaires.

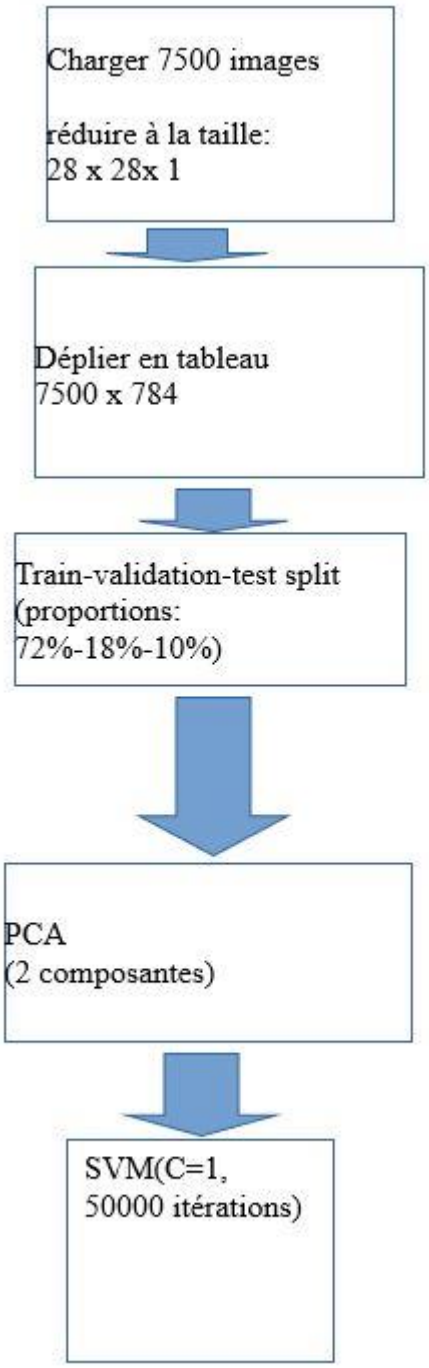


Figure 2 Pipeline modèle SVM sur les résultats d'une PCA à 2 composantes

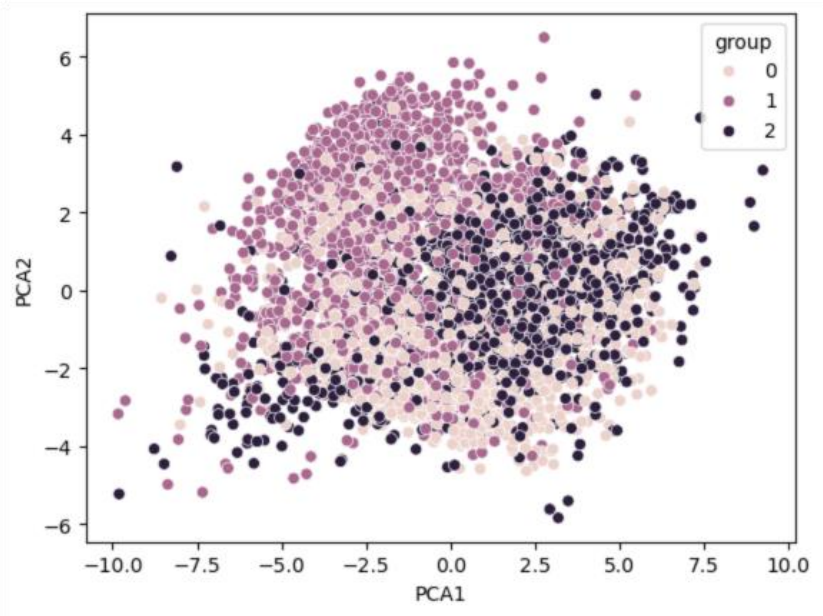


Figure 3 Projection des données sur les deux premières composantes de laPCA

2.3 Modèle Random Forest

2.3.1 Sur 4 classes

Parmi les modèles que nous avons retenu, il y a celui de Random Forest qui est plus facile et plus rapide à mettre en place pour une première approche. Sur ce modèle, l'ensemble des images a pu être traité grâce à une PCA mais nous avons choisi une résolution de 64*64 suite des contraintes liées à la RAM de nos ordinateurs étant assez restreintes. Tout d'abord via Visual studio code, nous avons utilisé le modèle sans masque qui nous a donné le rapport de classification suivant :

```
Accuracy : 0.7573824710607134
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.90      0.37      0.53       701
   Lung Opacity       0.75      0.70      0.72      1185
        Normal       0.73      0.93      0.82      2085
Viral Pneumonia       0.95      0.66      0.77       262

      accuracy                           0.76      4233
     macro avg       0.83      0.67      0.71      4233
  weighted avg       0.78      0.76      0.74      4233

AUC-ROC : 0.9404318489053559
```

Ce modèle a ensuite été optimisé grâce à l'utilisation de GridSearchCV sur les hyperparamètres :

```
Accuracy : 0.7961256791873376
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.80      0.58      0.67       701
   Lung Opacity       0.74      0.77      0.75      1185
        Normal       0.82      0.88      0.85      2085
Viral Pneumonia       0.87      0.85      0.86       262

      accuracy                           0.80      4233
     macro avg       0.81      0.77      0.78      4233
  weighted avg       0.80      0.80      0.79      4233

AUC-ROC : 0.9469869870993035
```

Un changement de résolution sur les images en 28*28 et une augmentation des données via ImageDataGenarator a permis d'avoir un f1-score plus élevé sur les classes COVID et Viral Pneumonia :

```
Accuracy : 0.7935270493739665
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.83      0.58      0.68       701
   Lung Opacity       0.73      0.74      0.74      1185
        Normal       0.81      0.89      0.85      2085
Viral Pneumonia       0.90      0.84      0.87       262

      accuracy                           0.79      4233
     macro avg       0.82      0.76      0.78      4233
  weighted avg       0.80      0.79      0.79      4233
```

Par souci de RAM, nous avons essayé de passer sur collab. Un autre modèle a été utilisé en remplaçant GridSearchCV par RandomSearchCV afin de gagner en temps de calcul mais les scores sont moins bons :

```
              precision    recall  f1-score   support

           0       0.74      0.21      0.32      4398
           1       0.69      0.61      0.65      7202
           2       0.67      0.91      0.77     12169
           3       0.79      0.57      0.66      1629

    accuracy                           0.68     25398
   macro avg       0.72      0.58      0.60     25398
weighted avg       0.69      0.68      0.65     25398
```

2.3.2 Sur 3 classes

Nous avons ensuite lancé le modèle avec masque en ayant un sous-ensemble de données équilibré en 3 classes (Covid, Normal, autre pathologie pulmonaire) de 7500 observations.

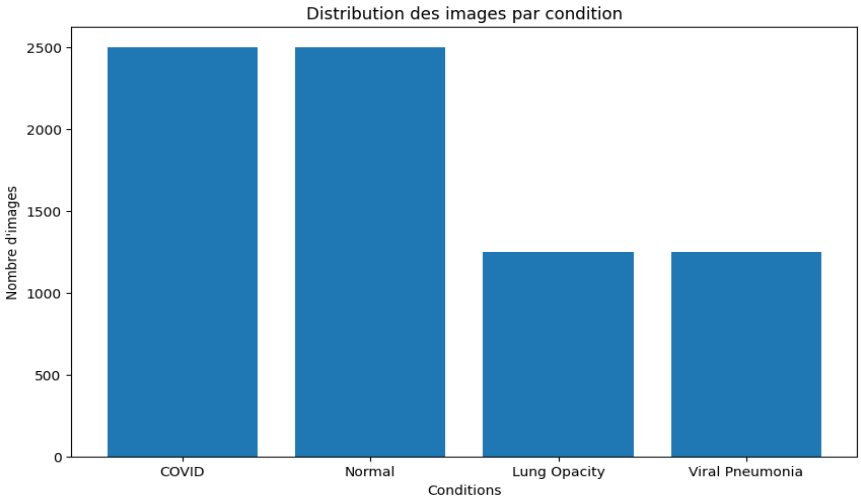


Figure 4 Nombre d'images de chaque condition échantillonnées pour former 3 groupes équilibrés

Les résultats suivants ont été obtenus sur des images 64*64 :

```
Accuracy: 0.718
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.62      0.64       527
           1       0.65      0.79      0.72       492
           2       0.88      0.75      0.81       481

    accuracy                           0.72      1500
   macro avg       0.73      0.72      0.72      1500
weighted avg       0.73      0.72      0.72      1500
```

On remarque une amélioration des performances en augmentant les données disponibles pour l'entraînement et une optimisation avec Grid Search qui a amélioré les performances du modèle en trouvant les paramètres optimaux.

2.4 XGBoost

Le modèle XGBoost a été étudié sur le problème à 4 classes avec équilibrage des classes. Les résultats sont présentés dans le rapport de classification ci-dessus, qui inclut des mesures de précision, rappel, F1-score, support, ainsi que l'accuracy globale et l'AUC-ROC. Voici une analyse détaillée de ces résultats.

Précision, Rappel et F1-score

1. **COVID :**
   - **Précision** : 0.80
   - **Rappel** : 0.78

- **F1-score** : 0.79
- **Support** : 433

Pour la classe COVID, la précision de 0.80 indique que 80% des prédictions étiquetées comme COVID sont correctes. Le rappel de 0.78 signifie que 78% des cas réels de COVID ont été correctement identifiés par le modèle. Le F1-score de 0.79, qui est la moyenne harmonique de la précision et du rappel, montre un bon équilibre entre ces deux métriques.

2. **Lung Opacity :**
   - **Précision** : 0.73
   - **Rappel** : 0.77
   - **F1-score** : 0.75
   - **Support** : 378

   Pour la classe Lung Opacity, la précision est de 0.73, ce qui indique que 73% des prédictions pour cette classe sont correctes. Le rappel de 0.77 montre que 77% des cas réels ont été détectés. Le F1-score est de 0.75, suggérant un bon équilibre entre la précision et le rappel, bien que légèrement inférieur par rapport à la classe COVID.

3. **Normal :**
   - **Précision** : 0.78
   - **Rappel** : 0.75
   - **F1-score** : 0.76
   - **Support** : 386

   Pour la classe Normal, la précision est de 0.78 et le rappel de 0.75, avec un F1-score de 0.76. Ces résultats montrent une performance légèrement inférieure comparée à la classe COVID mais similaire à Lung Opacity.

4. **Viral Pneumonia :**
   - **Précision** : 0.92
   - **Rappel** : 0.95
   - **F1-score** : 0.93
   - **Support** : 272

   Pour la classe Viral Pneumonia, la précision est exceptionnellement élevée à 0.92, avec un rappel de 0.95 et un F1-score de 0.93, ce qui montre une très bonne performance du modèle pour cette classe.

Métriques Globales

- **Accuracy** : 0.80
- **Macro avg** : 0.81 pour la précision, le rappel et le F1-score
- **Weighted avg** : 0.80 pour la précision, le rappel et le F1-score
- **AUC-ROC** : 0.9507428142905285

L'accuracy globale du modèle est de 0.80, ce qui signifie que 80% des prédictions étaient correctes. La moyenne macro et la moyenne pondérée montrent des valeurs de 0.81 et 0.80 respectivement, indiquant un bon équilibre global dans les performances du modèle à travers toutes les classes. L'AUC-ROC de 0.95 suggère une excellente capacité du modèle à distinguer entre les différentes classes.

Le modèle XGBoost montre des performances solides par rapport aux autres modèles de Machine Learning avec une accuracy globale de 0.80 et un AUC-ROC de 0.95, suggérant une bonne discrimination entre les classes. Les résultats pour les classes COVID et Viral Pneumonia sont particulièrement forts, tandis que les performances pour Lung Opacity et Normal peuvent bénéficier d'améliorations supplémentaires

3. Modèles de Deep Learning

A l'exception de XGBoost, les modèles de machine learning testés fournissent des performances assez limitées, même en optimisant leurs paramètres et augmentant la résolution des images. C'est pourquoi on s'intéresse maintenant à des modèles plus complexes de Deep Learnig.

Plusieurs types de modèles ont été étudiés : des CNN, des modèles de transfer learning, et des modèles d'extraction de features couplés à des méthodes de machine learning.

Dans tous les modèles présentés dans cette section, les images ont été augmentées par un générateur d'images de manière à améliorer leur robustesse.

De plus, afin d'améliorer la convergence et d'accélérer l'entraînement, des callbacks ont été introduits dans les programmes. Nous avons notamment utilisé les classes *EarlyStopping* et *ReduceLROnPlateau.*

3.1 CNN

3.1.1 CNN à 1 couche de convolution sur des images 28x28.

Le premier modèle de Deep Learning testé est un modèle CNN Ce modèle garde les données sous forme d'un tenseur, et applique un CNN avec une couche de convolution intermédiaire. Il est construit selon la procédure décrite en Figure 5.

Ses performances sont évaluées en cours d'entraînement sur l'échantillon de validation et un callback d'arrêt en cas d'absence d'amélioration de la précision sur ce dernier est ajouté afin d'éviter le surapprentissage.

Résultats.

Le score du modèle se trouve près de 0.69. Les scores F1 diffèrent pour les 3 classes :

- 0.68 pour la classe 0 (sains) ;
- 0.76 pour la classe 1 (malades non-COVID) ;
- 0.62 pour la classe 2 (COVID).

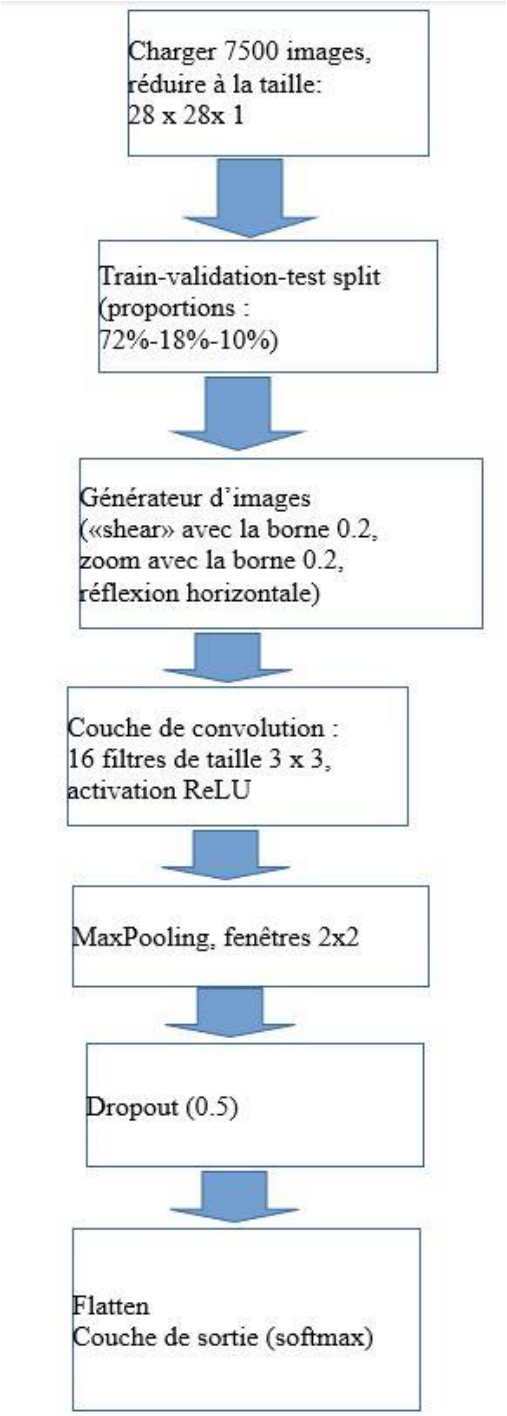


Figure 5 Procédure et architecture du CNN

Ce modèle présente un grand nombre de faux-négatifs (rappel pour la classe COVID : 56%). Voici quelques images du jeu de validation avec les classes vraies et prédites.

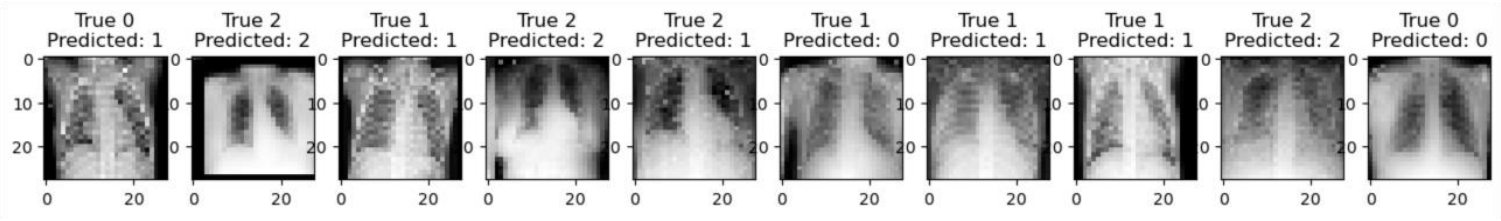


Figure 6 Prédictions du modèle CNN sur des images de test

3.1.2 CNN à trois couches sur des images 64x64 à 128x128

Le modèle est ensuite testé sur des images plus, de taille 64x64 ou 128x128. Quand les masques sont ajoutés, ils sont concaténés aux images suivant la 3^{e} dimension, ce qui donne des dimensions 28x28x2 ou 128x128x2. Des modèles avec 1, 2 ou 3 couches de convolution ont actuellement été testés. Celui à trois couches donne les meilleurs résultats et son organisation est illustrée dans le schéma suivant :

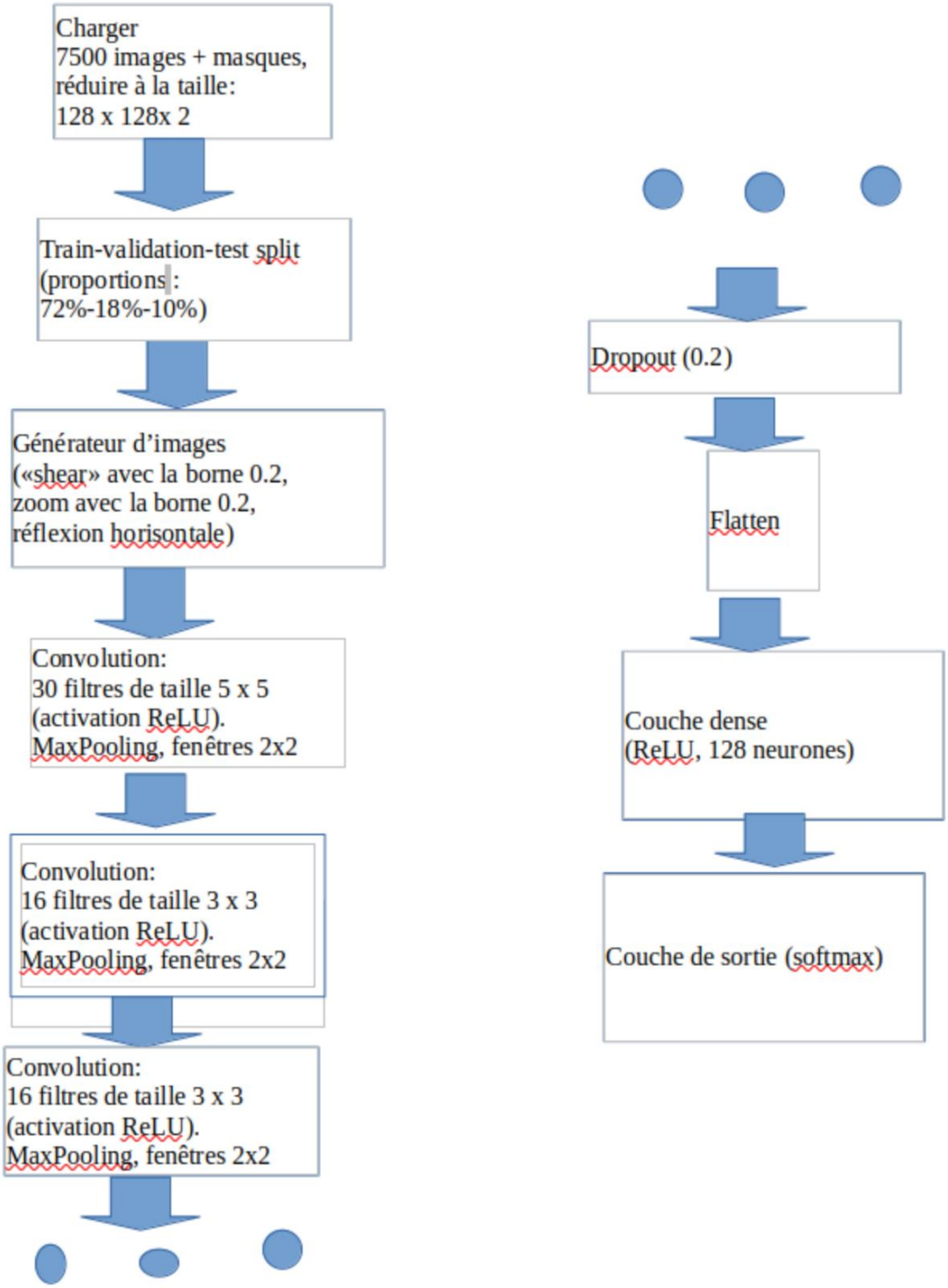


Figure 5 Workflow du meilleur modèle CNN (3 couches de convolution)

La figure 8 montre quelques images du jeu de validation avec leurs vrais labels ainsi que leurs labels prédits par le modèle.

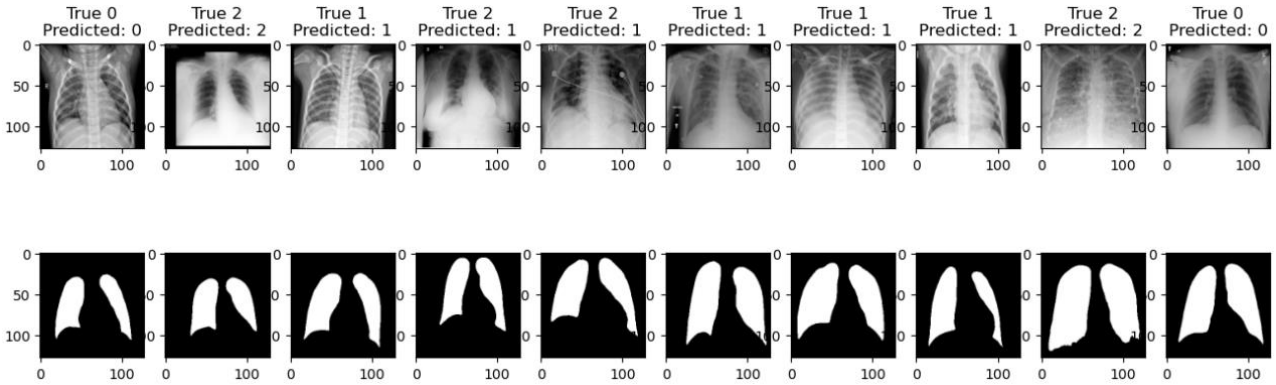


Figure 6 Prédictions du modèle CNN à 3 couches de convolution sur des images de test

Le tableau ci-dessous synthétise les scores de tous les modèles CNN testés :

Modèle / Résolution des images	CNN 1 couche	CNN 2 couches	CNN 3 couches
128x128x2		0.83	0.83
128x128		0.81	
64x64	0.71	0.77	
28x28x2	0.7		
28x28	0.69		

Le tableau ci-après donne les rappels pour la classe COVID :

Modèle / Résolution des images	CNN 1 couche	CNN 2 couches	CNN 3 couches
128x128x2		0.74	0.81
128x128		0.74	
64x64	0.55	0.69	
28x28x2	0.64		
28x28	0.57		

3.2 Extraction de features à partir de réseaux neuronaux

3.2.1 Avec Random Forest

Nous avons également testé l'utilisation de modèles neuronaux pour l'extraction de features tout en utilisant des forêts aléatoires pour la classification finale.

Après l'extraction des caractéristiques d'images pulmonaires à l'aide d'un ResNet50, un premier modèle combiné avec Random forest a été utilisé pour effectuer une classification finale. Les résultats obtenus sont issus du meilleur modèle sélectionné par un RandomizedSearchCV sur des images 64*64:

	precision	recall	f1-score	support
COVID	0.58	0.68	0.62	493
Normal	0.70	0.47	0.56	254
Lung Opacity	0.66	0.68	0.67	494
Viral Pneumonia	0.92	0.85	0.88	259
accuracy			0.67	1500
macro avg	0.72	0.67	0.69	1500
weighted avg	0.69	0.67	0.67	1500

Les paramètres peuvent être améliorés afin de gagner en performance. Un CNN a également été testé sur des images d'une dimension de 100*100 :

Classification Report:

	precision	recall	f1-score	support
0	0.56	0.87	0.68	750
1	0.86	0.48	0.61	750
2	0.89	0.81	0.85	750
accuracy			0.72	2250
macro avg	0.77	0.72	0.72	2250
weighted avg	0.77	0.72	0.72	2250

Le rapport de classification reste moyen mais encourageant si on rajoute des couches sur le CNN ou sur le ResNet50 avant de le combiner à un Random Forest.

3.2.2 Avec XGBoost

Le modèle XGBoost ayant démontré de bons résultats en termes de performance (0.84 en accuracy et auc-roc 0.96), nous avons réalisé une série d'expériences en utilisant l'extraction de features par divers modèles, notamment EfficientNet, VGG, LeNet et UNet (cf. Annexe I) afin d'améliorer ses performances. Cependant le transfert n'a pas permis de les améliorer.

3.3Transfer Learning

Le Transfer Learning est une approche de Deep Learning dans laquelle les performances d'un modèle d'apprentissage sont améliorées en transférant de l'information depuis un domaine d'étude proche.

On utilise des modèles pré-entraînés sur de larges bases de données d'un domaine voisin de celui d'intérêt, ici la classification d'images, de manière à initialiser les poids de notre modèle.

Les modèles existants comportent une partie d'extraction de features composées d'un ensemble de convolutions, suivie d'un ensemble de couches denses pour la classification. Les modèles pré-entraînés fournissent les poids de la première partie, facilitant l'apprentissage du modèle. On peut également choisir de réentraîner une partie des couches de convolutions de l'extraction de features afin de s'adapter au mieux à notre problème cible.

On compare ici plusieurs modèles de transfert : VGG, EfficientNet et ResNet, tous des modèles entraînés sur la base *ImageNet* contenant plus de 14 millions d'images labellisées. Les trois sont appliqués sur nos données masquées selon la procédure ci-dessous :

Création des échantillons	• Sur un ensemble de 7500 images réparties en 3 groupes équilibrés, séparation des données en trois échantillons : apprentissage, validation et test. • Augmentation des images par un générateur intégrant des modifications de zoom, de distorsion et symétrie.
Entraînement et sélection du modèle	• Entraînement et ajustement du modèle sur l'échantillon d'apprentissage au regard de ses performances sur celui de validation. • Des tests successifs faisant varier les paramètres et la structure du modèle sont réalisés afin de sélectionner le plus performant, puis le modèle retenu est entraîné sur les images dans leur résolution originale et sauvegardé.
Evaluation des performances	• Evaluation des performances du modèle sur l'échantillon de test.

Les deux premiers modèles ont été implémentés sur le problème de classification à 3 classes et le dernier est présenté sur 4 classes.

3.3.1 VGG16

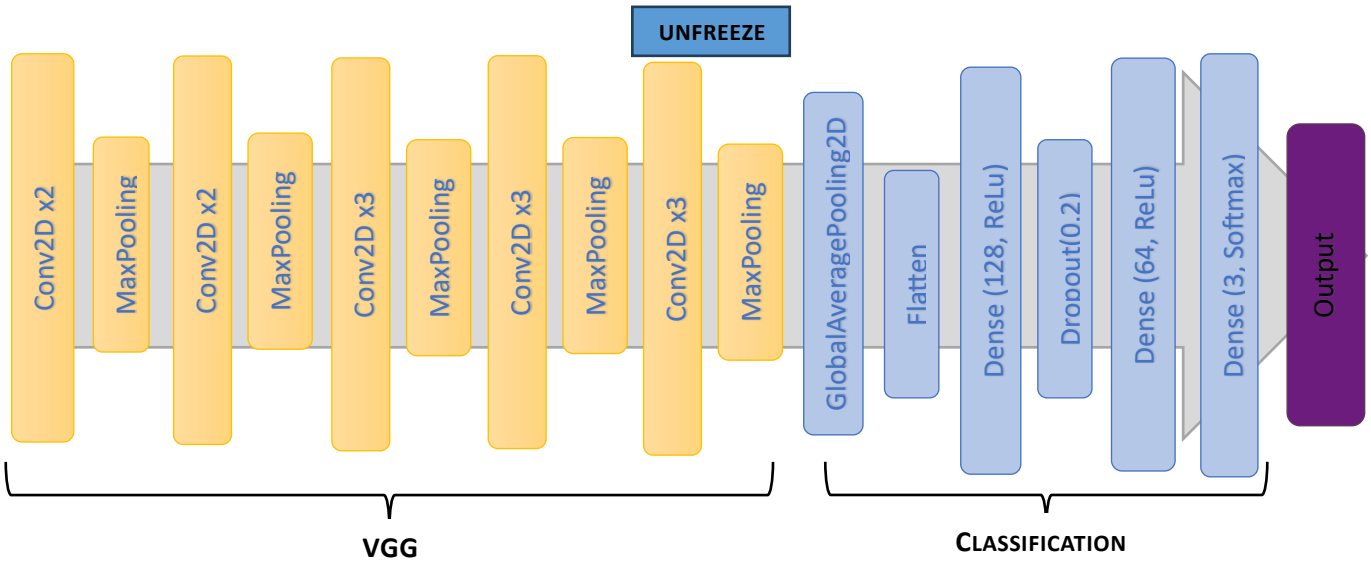
L'extraction de features fournie par VGG est constituée de 5 blocs convolutifs comprenant chacun entre 2 et 3 couches. On y ajoute des couches denses de classification.

Le modèle est ajusté sur les images de résolution réduite (64x64) afin de permettre une exécution suffisamment rapide pour tester plusieurs modélisations. Les principaux paramètres investigués sont les suivants :

- Le nombre de couches denses (de 1 à 5) et le nombre de neurones par couche ;
- Le nombre d'étapes de dropout et leur intensité entre les couches denses ;

- L'utilisation directe des poids pré-entraînés directement ou le réentraînement du dernier bloc de convolutions.

L'architecture du modèle retenu est la suivante :



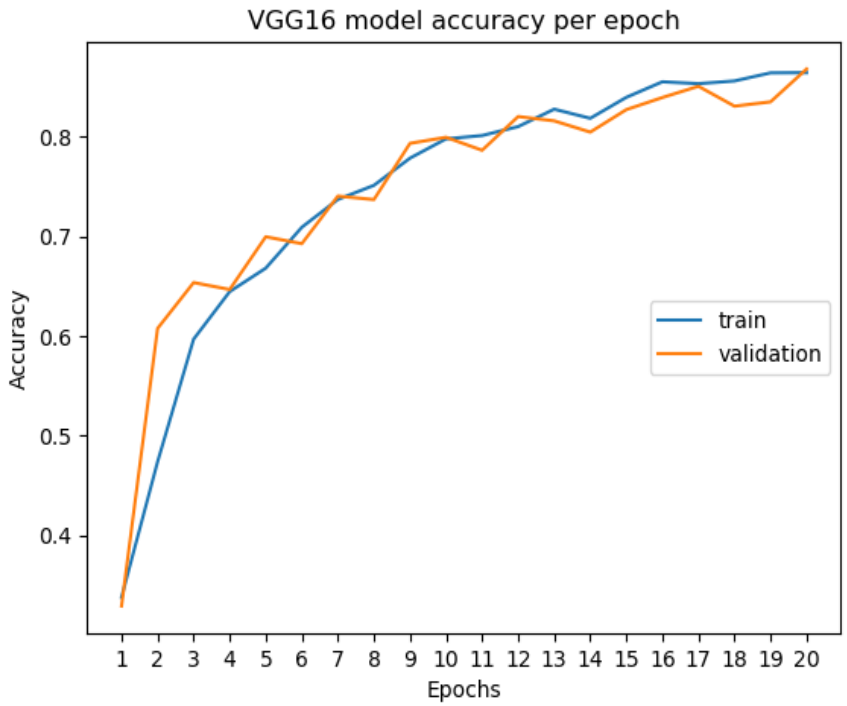


Figure 7 Précision du modèle en fonction du nombre d'epochs

Ce modèle a été implémenté sur le jeu de 7500 images en résolution 299x299 pour 20 epochs. L'évolution des performances du modèle sur les échantillons d'apprentissage et de validation sont données par la Figure 8. Le modèle atteint une précision de 87%, mais semble ne pas avoir convergé à l'issue des 20 epochs. Il faudrait donc le réentraîner sur un nombre plus important afin d'avoir une évaluation plus précise de ses performances réelles. La difficulté réside ici dans le coût computationnel de ce modèle : son temps d'entraînement sur les 20 epochs est de près de 6h.

Les performances en prédiction sont évaluées sur l'échantillon de test. On obtient une précision générale de 87% et un f1-score global de 85%. Le détail par classe est donné dans le tableau suivant :

	Precision	Recall	f1-score
COVID	0.85	0.86	0.85
Normal	0.85	0.91	0.88
Pulmonary Infection	0.91	0.85	0.88

Là encore, on note que la classe Covid est la plus difficile à identifier avec une précision de 85% et un recall de 86%.

3.3.2 EfficientNet

Le modèle de transfert suivant est EfficientNet, qui utilise une mise à l'échelle composée ajustant profondeur, largeur et résolution pour maximiser l'efficacité du modèle.

Le type de modèle EfficientNet à sélectionner étant dépendant de la dimension des images en entrée, les étapes exploratoires se sont faites, d'abord sur des images 64x64 en utilisant un modèle EfficientNetB3, puis sur les données de dimension 128x128 pixels avec un modèle B5.

Le passage à l'échelle des données n'a pas permis d'amélioration des performances du modèle et les investigations n'ont pas été plus poussée.

L'entraînement du modèle est bien moins long que pour VGG, cependant les performances sont plutôt décevantes : la précision globale du modèle obtenue est de 73% sur les images 128x128 et f1-score est de 73%.

	Precision	Recall	f1-score
COVID	0.69	0.71	0.70
Normal	0.67	0.79	0.73
Pulmonary Infection	0.86	0.71	0.78

3.3.3 ResNet

ResNet, ou Réseau Résiduel, est un modèle de transfert permettant l'entraînement de réseaux très profonds grâce aux connexions de saut qui facilitent la propagation du gradient, évitant la dégradation. Nous souhaitons utiliser une architecture ResNet50 pré-entraînée pour extraire des caractéristiques complexes des images. En effet le modèle ResNet50 est chargé avec des poids pré-entraînés sur *ImageNet*. Certaines couches supplémentaires ont été ajoutées pour que le modèle s'adapte spécifiquement à la réalisation de la classification des images de radiologie, incluant des couches de global average pooling, des couches dense et de dropout.

Résultats obtenus :

Le modèle est testé sur le problème de classification à 4 classes et les résultats obtenus sont les suivants :

```
F1-score: 0.8978579463711258
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.89      0.90      0.90       433
        Normal       0.88      0.89      0.89       386
   Lung Opacity       0.87      0.85      0.86       378
Viral Pneumonia       0.97      0.97      0.97       272

      accuracy                           0.90      1469
     macro avg       0.90      0.90      0.90      1469
  weighted avg       0.90      0.90      0.90      1469
```

Les résultats montrent que le modèle ResNet offre une performance robuste et équilibrée pour la classification des conditions pulmonaires, avec une précision et un rappel global de 90%. Les performances sont extrêmement bonnes sur la classe *Viral Pneumonia*, ce qui confirme les observations déjà réalisées sur les autres modèles de la relative facilité à distinguer ce groupe des autres. Il est également notable que les performances sur la classe Covid sont bien meilleures que pour les autres modèles avec une précision et un recall de respectivement 89 et 90%.

L'architecture de ResNet semble ainsi particulièrement adaptée au problème de classification d'images radio.

4. Synthèse des résultats obtenus

Le tableau ci-dessous synthétise les principaux résultats obtenus sur les images masquées. Seuls sont retenus les meilleurs modèles de chaque type. Notons que certains ont été implémentés sur le problème à 3 classes et d'autres à 4 classes, ce qui suppose de les comparer avec précautions. De plus les résolutions des images utilisées varient d'un modèle à l'autre en fonction d'une part, de si le modèle a été perçu comme prometteur, et donc étudié plus en profondeur, et d'autre part, de son coût computationnel.

Modèle	**Nombre de classes**	**Résolution des images**	**Scores globaux**		**Scores sur la classe Covid**	
			Précision	**f1-score**	**Précision**	**Rappel**
XGBoost	4	256 x 256	0.80	0.80	0.80	0.78
K-NN	3	299 x 299	0.66	0.65	0.58	0.60
Random Forest	4	64 x 64	0.73	0.73	0.66	0.64
CNN (3 couches de convolution)	3	128x128	0.83	0.81	0.82	0.81
Extraction de features + Random Forest	3	100x100	0.72	0.72	0.56	0.87
VGG16 (réentraînement de 4 couches)	3	299x299	0.85	0.85	0.85	0.86
EfficientNet	3	128x128	0.73	0.73	0.69	0.71
EfficientNet + VGG	4	256x256	0.71	0.71	0.69	0.60
ResNet + VGG	4	256x256	0.81	0.81	0.78	0.77
ResNet	4	256 x 256	0.90	0.90	0.89	0.90

Malgré les réserves énoncées ci-dessus, 4 modèles se démarquent :

- Le modèle XGBoost, dont les performances sont très honorables pour un modèle de Machine Learning ;
- Le CNN qui obtient de bons résultats, notamment sur la classe Covid et qui est un modèle de Deep Learning simple à implémenter ;
- VGG16 dont les résultats pourraient encore être améliorés en augmentant le nombre d'epochs, mais qui est très coûteux en temps de calcul
- Et ResNet qui obtient de loin les meilleurs résultats en classification.

Ces quatre modèles pourront dans la prochaine étape de cette étude être comparés du point de vue de leurs performances sur un même problème de classification ; de leur facilité d'implémentation et de calcul et de leur interprétabilité.

Pistes d'amélioration

Plusieurs stratégies peuvent être adoptées afin d'améliorer les performances des modèles retenus :

- Dans un premier temps, le passage à la résolution originale des masques (256x256) ou des images (299x299) peut permettre de gagner en précision. D'autre part, la phase exploratoire ayant montré que les performances sur les images avec masques sont meilleures et évitent mieux l'écueil du surapprentissage, on pourrait chercher à regénérer les masques par une méthode de segmentation (type UMAP).
- Un autre levier d'amélioration consisterait à optimiser plus finement et de manière automatisée les hyperparamètres des modèles. Par exemple, il est possible de jouer sur la complexité des modèles convolutifs en ajoutant une 4^{e} couche dans le but de mieux classifier les images difficiles. Plus généralement, les techniques de régularisation comme le dropout et la normalisation par lots permettent de stabiliser l'apprentissage (Srivastava et al., 2014; Ioffe & Szegedy, 2015).
- Un meilleur calibrage de la méthode d'augmentation des images pourrait également être étudié, du point de vue du nombre d'images générées et des paramètres des générateurs d'images afin d'améliorer la robustesse du modèle. Par exemple, les augmentations géométriques et photométriques introduisent une diversité spatiale et simulent des conditions d'éclairage variées (Perez & Wang, 2017 ; Howard, 2013).
- Enfin, des méthodes ensemblistes combinant différents types de modèles pourraient être investiguées (VotingClassifier ou Bagging) pour tenter de réduire le nombre de faux négatifs.

Les points cités ci-dessus doivent néanmoins faire l'objet d'un arbitrage gain/coût (en temps et en ressources de calcul) : la régularisation par exemple peut ralentir la convergence du modèle et la génération de données nécessite des ajustements précis pour éviter des augmentations non pertinentes. Les méthodes ensemblistes, quant à elles, augmentent significativement la complexité et le temps d'exécution.

Références

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157-166.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In International Workshop on Multiple Classifier Systems (pp. 1-15). Springer, Berlin, Heidelberg.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Thèse de doctorat.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (ICML).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 60.

Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197-227.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (ICML).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems (NeurIPS).

Annexes

Annexe 1 : Test d'extraction de features avec des modèles VGG, ResNet et LeNet combinés à XGBoost

Efficient Net VGG en transfert à XGBoost avec classe équilibrées

```
Précision : 0.7139874739039666
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.69      0.60      0.64       427
   Lung Opacity       0.67      0.72      0.69       366
        Normal       0.68      0.74      0.71       379
Viral Pneumonia       0.87      0.85      0.86       265

      accuracy                           0.71      1437
     macro avg       0.73      0.73      0.73      1437
  weighted avg       0.72      0.71      0.71      1437

AUC-ROC : 0.9176255101915743
Score F1 : 0.7132501793217237
```

Le rapport de classification présenté ci-dessus résume les performances du modèle XGBoost avec transfert de caractéristiques provenant de EfficientNet et VGG. Voici une analyse détaillée des différentes métriques :

Précision, Rappel et F1-score

1. **COVID :**
   - **Précision** : 0.69
   - **Rappel** : 0.60
   - **F1-score** : 0.64
   - **Support** : 427

   La précision de 0.69 pour la classe COVID indique que 69% des prédictions étiquetées comme COVID étaient correctes. Le rappel de 0.60 montre que 60% des cas réels de COVID ont été correctement identifiés. Le F1-score de 0.64 combine ces deux métriques, soulignant une performance modérée pour cette classe.

2. **Lung Opacity :**
   - **Précision** : 0.67
   - **Rappel** : 0.72
   - **F1-score** : 0.69
   - **Support** : 366

Pour la classe Lung Opacity, la précision est de 0.67 et le rappel de 0.72. Le F1-score de 0.69 montre un bon équilibre entre la précision et le rappel, bien que les performances pourraient être améliorées pour mieux détecter cette classe.

3. **Normal :**
    - **Précision** : 0.68
    - **Rappel** : 0.74
    - **F1-score** : 0.71
    - **Support** : 379

    Pour la classe Normal, la précision est de 0.68 et le rappel de 0.74. Le F1-score de 0.71 indique une performance relativement bonne mais améliorable, en particulier en termes de précision.

4. **Viral Pneumonia :**
    - **Précision** : 0.87
    - **Rappel** : 0.85
    - **F1-score** : 0.86
    - **Support** : 265

    La classe Viral Pneumonia montre les meilleures performances avec une précision de 0.87, un rappel de 0.85, et un F1-score de 0.86, indiquant une très bonne capacité du modèle à détecter cette classe.

Métriques Globales

- **Accuracy** : 0.71
- **Macro avg** : 0.73 pour la précision, le rappel et le F1-score
- **Weighted avg** : 0.72 pour la précision, 0.71 pour le rappel et le F1-score
- **AUC-ROC** : 0.9176255101915743
- **Score F1** : 0.7132501793217237

Les résultats montrent que le modèle XGBoost avec transfert de EfficientNet et VGG sont inférieurs à ceux obtenus avec le seul algorithme XGBoost. Pour améliorer ces résultats, des techniques d'optimisation des hyperparamètres comme la recherche en grille et la validation croisée pourraient être envisagées.

LeNetVGGXGBoost

```
Accuracy: 0.7617233991906689
Classification Report:
                precision    recall  f1-score   support

         COVID       0.79      0.30      0.43       697
   Lung Opacity       0.75      0.76      0.75      1177
        Normal       0.76      0.92      0.83      2065
Viral Pneumonia       0.81      0.79      0.80       262

      accuracy                           0.76      4201
     macro avg       0.78      0.69      0.70      4201
  weighted avg       0.76      0.76      0.74      4201

AUC-ROC: 0.9236279782746822
F1 Score: 0.7412737385173055
```

ResNet VGG

```
Rapport de classification :
                precision    recall  f1-score   support

         COVID       0.71      0.53      0.61       427
   Lung Opacity       0.67      0.71      0.69       366
        Normal       0.60      0.78      0.68       379
Viral Pneumonia       0.87      0.78      0.82       265

      accuracy                           0.69      1437
     macro avg       0.71      0.70      0.70      1437
  weighted avg       0.70      0.69      0.69      1437

AUC-ROC : 0.8966311302456889
Score F1 : 0.6867293459019678
```