

## Projet Deep-Learning

Intervenante : Mme GHARBI

Membre du groupe :

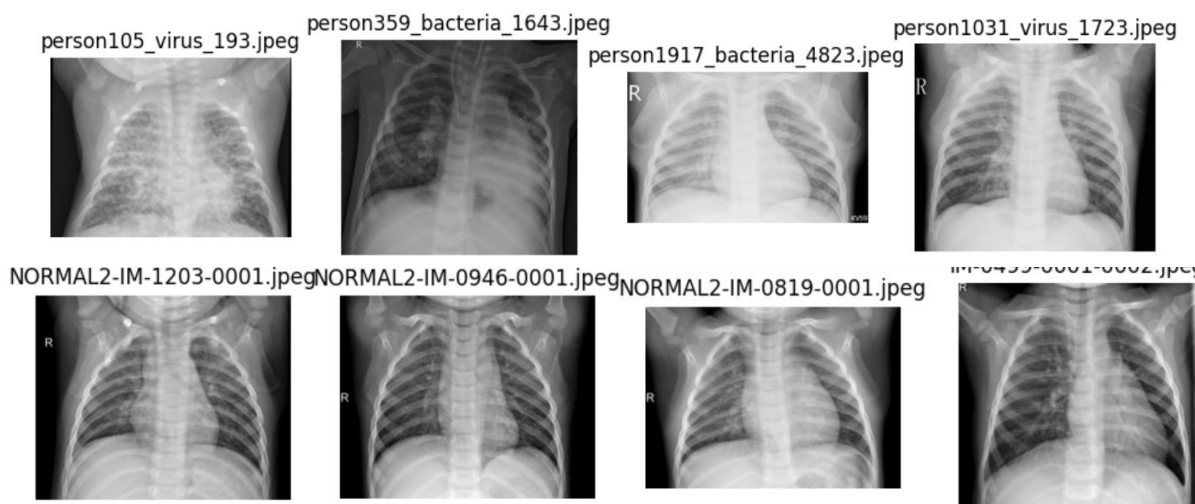
Yoni MICHARD- Semy BOUACID

Axel TOUSSENEL– Mohamed DIOUF

### **Rapport sur la Construction et l'Évaluation d'un Modèle d'Apprentissage en Profondeur pour la Classification d'Images Médicales**

#### **Introduction**

La pneumonie reste l'une des principales causes de décès chez les enfants et les adultes dans le monde, en particulier dans les pays en développement et chez les populations vulnérables. Selon l'Organisation mondiale de la santé (OMS), la pneumonie tue des centaines de milliers d'enfants de moins de cinq ans chaque année, principalement dans les pays à faible et moyen revenu. En 2020, la pandémie de COVID-19 a également mis en lumière l'importance critique de la pneumonie en tant que cause de morbidité et de mortalité dans le monde entier, en particulier chez les personnes âgées et les personnes atteintes de conditions médicales sous-jacentes. Selon l'OMS, la pneumonie est responsable d'environ 2 millions de décès par an dans le monde, dont une grande partie pourrait être évitée par des mesures préventives telles que la vaccination, l'accès aux soins de santé de base et la détection de la maladie à temps. L'objectif de ce projet est de construire un modèle d'apprentissage en profondeur capable de classer des images médicales en deux catégories : "Pneumonia" (pneumonie) et "Normal". Pour ce faire, nous avons utilisé un ensemble de données contenant des radiographies pulmonaires provenant de patients atteints ou non de pneumonie.



Nous avons adopté une approche combinée utilisant à la fois des réseaux de neurones convolutionnels (CNN) et des réseaux de neurones récurrents (RNN) pour capturer à la fois les informations spatiales et séquentielles des images.

## Prétraitement des Données

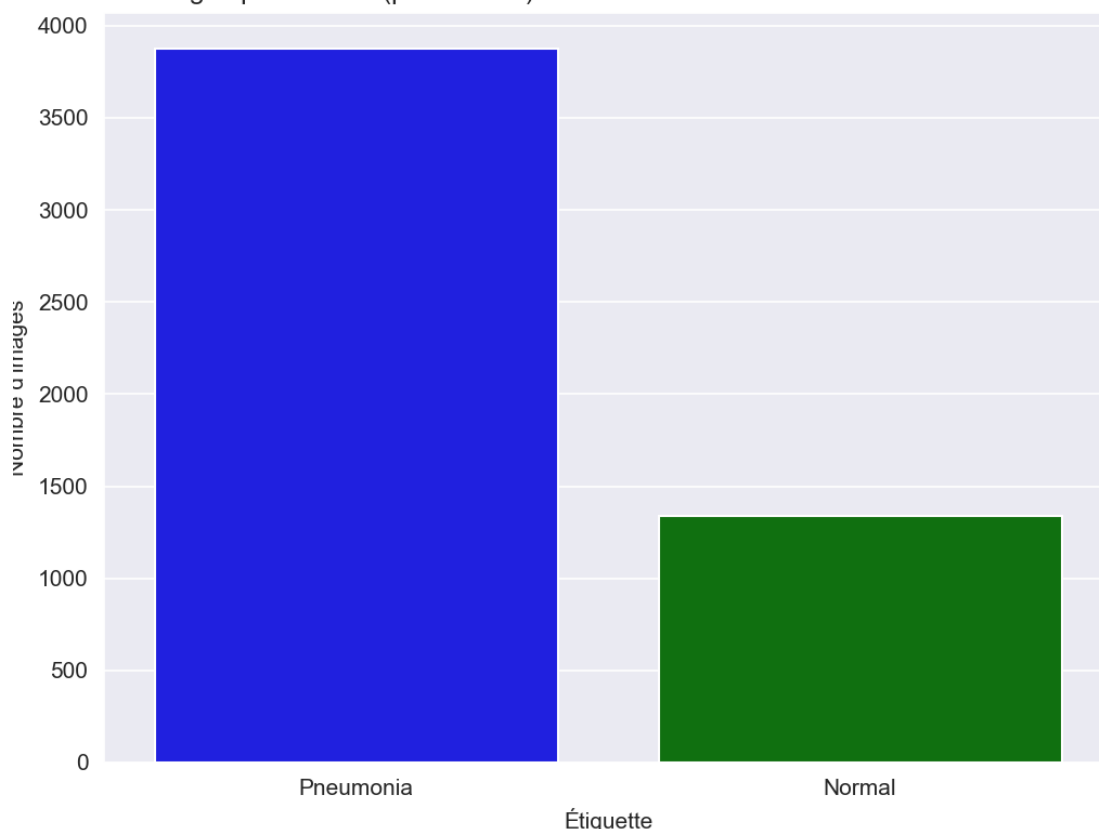
Avant de construire le modèle, nous avons effectué plusieurs étapes de prétraitement des données :

1. **Chargement des Données** : Les données ont été chargées à partir de trois répertoires différents : "train", "test" et "validation". Chaque répertoire contient des sous-répertoires pour chaque classe ("PNEUMONIA" et "NORMAL").

La fonction **get\_data** est responsable de la création d'un DataFrame contenant les données d'images pour l'entraînement, le test et la validation. En plus de cette fonctionnalité, elle fournit également des informations cruciales sur la répartition des données entre les classes "Normal" et "Pneumonia" dans chaque ensemble de données.

- Pour l'ensemble d'entraînement, nous avons un total de 1341 images normales et 3875 images de pneumonie.
- Pour l'ensemble de test, nous avons 234 images normales et 390 images de pneumonie.
- Pour l'ensemble de validation, nous avons 8 images normales et 8 images de pneumonie.

Nombre d'images pneumonia (pneumonie) et normal dans l'ensemble de données d'entraînement



Ces informations sont essentielles pour comprendre la distribution des classes dans nos ensembles de données, ce qui peut avoir un impact significatif sur l'entraînement et l'évaluation de notre modèle.

2. **Redimensionnement et Normalisation** : Les images ont été redimensionnées à une taille de 150x150 pixels et normalisées en divisant les valeurs des pixels par 255 pour les ramener dans la plage [0, 1].
3. **Augmentation des Données** : Nous avons utilisé l'augmentation de données pour augmenter la variabilité de notre ensemble de données. Cela inclut des opérations telles que la rotation, le zoom, le décalage horizontal et vertical, ainsi que le retournement horizontal des images.

### Construction du Modèle

Nous avons construit un modèle combiné CNN-RNN pour capturer à la fois les informations spatiales et séquentielles des images :

4. **Modèle CNN** : Nous avons construit un modèle CNN comprenant plusieurs couches convolutionnelles suivies de couches de pooling pour extraire les caractéristiques spatiales des images. Nous avons également ajouté un bloc résiduel pour aider le modèle à apprendre des représentations plus complexes.

### Choix des Hyperparamètres pour le Modèle CNN :

- **Nombre de Couches Convolutionnelles** : Le choix du nombre de couches convolutionnelles peut affecter la capacité du modèle à extraire des caractéristiques pertinentes des images. Dans ce projet, nous avons utilisé cinq couches convolutionnelles pour capturer des informations à différentes échelles spatiales.
- **Taille des Filtres Convolutionnels** : La taille des filtres affecte la taille de la région spatiale sur laquelle les caractéristiques sont extraites. Des filtres plus grands peuvent capturer des caractéristiques plus globales, tandis que des filtres plus petits peuvent capturer des détails fins. Dans ce projet, nous avons utilisé des filtres de taille 3x3 pour chaque couche convolutionnelle.
- **Nombre de Filtres par Couche** : Le nombre de filtres par couche détermine le nombre de caractéristiques spatiales extraites à chaque étape du processus de convolution. Une augmentation du nombre de filtres peut permettre au modèle de capturer des informations plus complexes, mais cela augmente également la complexité du modèle et le temps d'entraînement. Dans ce projet, nous avons utilisé des nombres de filtres progressivement croissants (64, 128, 256, 512) pour les différentes couches convolutionnelles.
- **Taille des Couches de Pooling** : Les couches de pooling sont utilisées pour réduire la dimensionnalité de la représentation spatiale, tout en préservant les informations les plus importantes. La taille des couches de pooling affecte la quantité d'informations spatiales conservées après chaque opération de pooling.

Dans ce projet, nous avons utilisé des couches de pooling avec une taille de fenêtre de 2x2 et un pas de 2 pour réduire la taille spatiale des caractéristiques.

- **Taux de Dropout :** Le taux de dropout est une technique de régularisation utilisée pour réduire le surapprentissage en désactivant aléatoirement un pourcentage des neurones lors de l'entraînement. Un taux de dropout plus élevé peut aider à prévenir le surapprentissage, mais il peut également ralentir l'apprentissage du modèle. Dans ce projet, nous avons utilisé un taux de dropout de 0.3 après les couches de pooling pour la régularisation.

5. **Modèle RNN :** Pour capturer les informations séquentielles, nous avons utilisé un modèle RNN avec des couches bidirectionnelles de GRU, suivies de couches de pooling pour agréger les informations temporelles.

#### **Choix des Hyperparamètres pour le Modèle RNN :**

- **Nombre d'Unités GRU:** Le nombre d'unités GRU dans chaque couche détermine la capacité du modèle à capturer des dépendances temporelles dans les séquences d'entrée. Des nombres d'unités plus élevés peuvent permettre au modèle de capturer des informations plus complexes, mais cela augmente également la complexité du modèle. Dans ce projet, nous avons utilisé des unités GRU avec des tailles de 256, 128 et 64 dans les différentes couches GRU.
- **Dropout Spatial :** Le dropout spatial est une forme de régularisation spécifiquement conçue pour les réseaux de neurones récurrents. Il consiste à désactiver aléatoirement des entiers aléatoires dans la séquence d'entrée à chaque mise à jour de poids. Cela aide à prévenir le surapprentissage en introduisant du bruit dans les données d'entrée. Dans ce projet, nous avons utilisé un dropout spatial de 0.3 après chaque couche GRU pour la régularisation.
- **Type de Pooling :** Le choix entre le pooling global maximum et le pooling global moyen affecte la manière dont les informations temporelles sont agrégées dans les séquences GRU. Le pooling global maximum extrait les caractéristiques les plus importantes de la séquence, tandis que le pooling global moyen calcule la moyenne des caractéristiques sur toute la séquence. Dans ce projet, nous avons utilisé les deux types de pooling pour capturer différentes perspectives des données séquentielles.

6. **Modèle Combiné :**

- **Extraction des Caractéristiques Spatiales avec CNN :**
  - Le modèle CNN est utilisé pour extraire les caractéristiques spatiales des images médicales. Les couches convolutionnelles du CNN analysent les motifs

visuels et les structures importantes dans les images, capturant ainsi des informations telles que les contours, les textures et les motifs distinctifs.

- Après avoir traversé les couches convolutionnelles et de pooling, le CNN génère un vecteur de caractéristiques spatiales pour chaque image, qui représente les informations spatiales importantes extraites de l'image.

- **Capture des Informations Séquentielles avec RNN :**

- Le modèle RNN est utilisé pour capturer les informations séquentielles des caractéristiques spatiales générées par le CNN. Les réseaux de neurones récurrents sont bien adaptés à la modélisation de séquences, car ils peuvent prendre en compte les dépendances temporelles entre les étapes successives d'une séquence.
- Dans ce projet, les caractéristiques spatiales extraites par le CNN sont traitées comme une séquence temporelle, où chaque vecteur de caractéristiques représente une étape dans la séquence. Le RNN analyse cette séquence de caractéristiques spatiales pour capturer les relations temporelles et séquentielles entre les différentes caractéristiques.

- **Combinaison des Sorties CNN et RNN :**

- Une fois que le CNN et le RNN ont chacun extrait les informations spatiales et séquentielles, respectivement, les sorties des deux modèles sont combinées pour former un vecteur de caractéristiques combinées.
- Ce vecteur de caractéristiques combinées capture à la fois les informations spatiales et séquentielles importantes des images médicales. Il est ensuite passé à travers des couches denses pour la classification finale en tant que "Pneumonia" ou "Normal".

- **Formation du Modèle Combiné :**

- Le modèle combiné, qui prend à la fois les données d'entrée de l'image brute et les caractéristiques séquentielles extraites par le CNN, est alors entraîné sur les données d'entraînement pour apprendre à classifier correctement les images.
- L'entraînement du modèle combiné implique l'ajustement simultané des poids des couches CNN et RNN ainsi que des couches denses finales, afin d'optimiser la performance globale du modèle sur la tâche de classification.

Model: "functional\_3"

Layer (type)	Output Shape	Param #	Connected to
input_layer_2 (InputLayer)	(None, None, 12800)	0	-
dense_1 (Dense)	(None, None, 32)	409,632	input_layer_2[0][0]
dense_2 (Dense)	(None, None, 32)	1,056	dense_1[0][0]
dense_3 (Dense)	(None, None, 128)	4,224	dense_2[0][0]
dropout_1 (Dropout)	(None, None, 128)	0	dense_3[0][0]
input_layer (InputLayer)	(None, 150, 150, 1)	0	-
dense_4 (Dense)	(None, None, 1)	129	dropout_1[0][0]

Total params: 415,041 (1.58 MB)  
Trainable params: 415,041 (1.58 MB)  
Non-trainable params: 0 (0.00 B)

## 7. Entraînement et Évaluation du Modèle

- **Entraînement du Modèle :**

- **Algorithme d'Optimisation :** Nous avons utilisé l'algorithme d'optimisation "rmsprop" (Root Mean Square Propagation) pour ajuster les poids du modèle pendant l'entraînement. RMSprop est un algorithme d'optimisation adaptatif qui ajuste le taux d'apprentissage pour chaque paramètre en fonction de la moyenne exponentielle des carrés des gradients précédents. Cela permet un ajustement plus précis des poids du modèle et une convergence plus rapide.
- **Fonction de Perte :** Nous avons utilisé la fonction de perte "binary\_crossentropy" pour calculer la perte pendant l'entraînement. Binary Crossentropy est une fonction de perte couramment utilisée pour les problèmes de classification binaire. Elle mesure la différence entre les valeurs prédites par le modèle et les valeurs réelles, en tenant compte de la distribution de probabilité des classes.
- **Ajustement du Taux d'Apprentissage :** Nous avons utilisé une technique d'ajustement du taux d'apprentissage basée sur la surveillance de la précision de validation. Cette technique ajuste dynamiquement le taux d'apprentissage pendant l'entraînement en réduisant progressivement le taux d'apprentissage si la précision de validation cesse de s'améliorer pendant un certain nombre d'epochs consécutifs. Cela aide à stabiliser l'entraînement et à éviter le surapprentissage.

- **Évaluation du Modèle :**

- **Calcul de la Perte et de l'Accuracy :** Une fois l'entraînement terminé, nous avons évalué les performances du modèle sur l'ensemble de test en calculant la perte et l'accuracy. La perte est une mesure de la différence entre les valeurs prédites par le modèle et les valeurs réelles dans l'ensemble de test, tandis que l'accuracy mesure le pourcentage d'images correctement classées par le modèle.

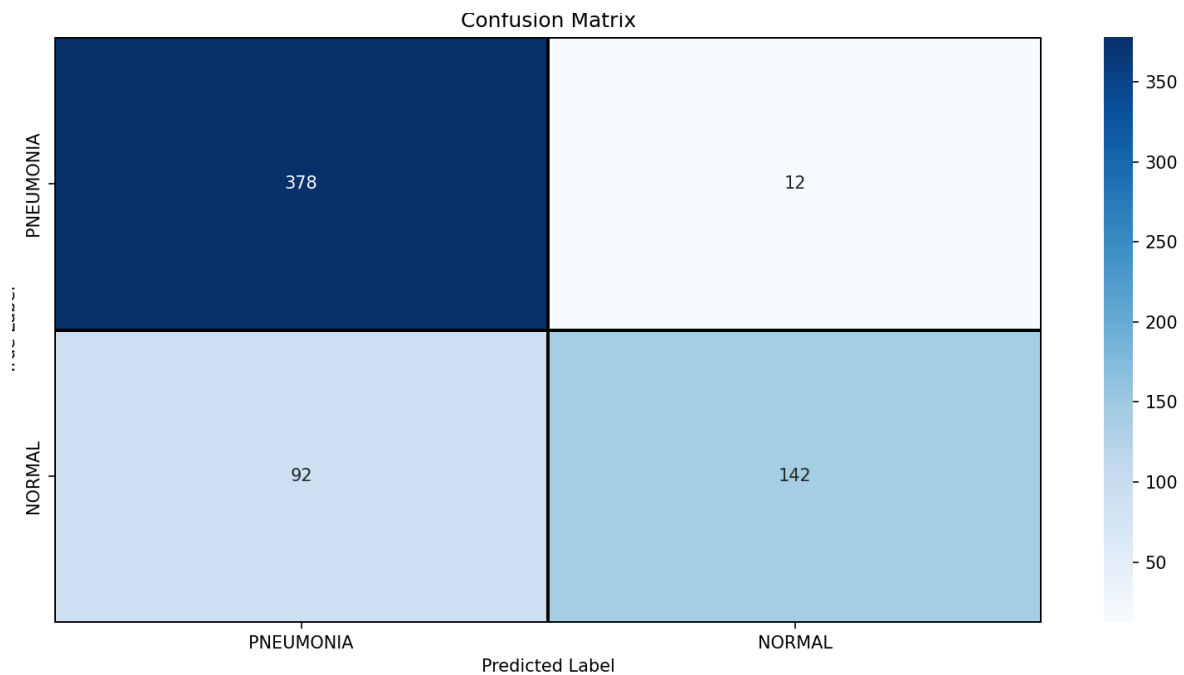
La perte du modèle est : 0.47833266854286194  
L'accuracy du modèle est : 83.81410241127014 %

Avec une perte de **0,47** et un accuracy de **84%**, notre modèle semble avoir une performance raisonnable sur des données de test mais reste tout de même largement améliorable.

- **Génération du Rapport de Classification** : Nous avons généré un rapport de classification pour évaluer les performances du modèle sur chaque classe. Le rapport de classification fournit des mesures de précision, de rappel, de F1-score et de support pour chaque classe, ce qui permet d'évaluer la capacité du modèle à classer correctement les images dans la classe 0 (Pneumonia) avec 88% F1-Score ; mais une précision beaucoup plus faible lorsqu'il s'agit de la classe 1 (normal).

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.81	0.97	0.88	390
Normal (Class 1)	0.92	0.62	0.74	234
accuracy			0.84	624
macro avg	0.86	0.80	0.81	624
weighted avg	0.85	0.84	0.83	624

- **Tracé de la Matrice de Confusion** : Nous avons tracé une matrice de confusion pour visualiser les performances du modèle en termes de classifications correctes et incorrectes pour chaque classe. Une matrice de confusion présente les prédictions du modèle par rapport aux vraies valeurs pour chaque classe, ce qui permet d'identifier les erreurs de classification et les tendances de performances du modèle.



Tout comme nous l'indiquait le rapport de classification, la matrice de confusion nous confirme que notre modèle a plus de mal pour prédire avec précision les imageries de personne normale.

## Conclusion

En conclusion, nous avons construit avec succès un modèle combiné CNN-RNN pour la classification d'images médicales. Ce modèle a démontré de bonnes performances en termes de précision et de capacité à distinguer les images de pneumonie des images normales. Cependant, il existe encore des possibilités d'amélioration, notamment en explorant d'autres architectures de modèles, en ajustant les hyperparamètres et en collectant des données supplémentaires pour améliorer la généralisation du modèle.