



Projet Deep Learning

Classification d'images médicales avec un CNN-RNN hybride

Élaboré par :

HAMZA REGUIG Hichem Abdennacer
MAZROUK Karim
ESSAMADI Mohamed Yassine
HATRI Nabil
KHALFOUN Rayan

Encadré par :

GHARBI Ines

Année Universitaire : 2023/2024

Table Des Matières :

1	<u>INTRODUCTION :</u>	4
2	<u>DATA :</u>	5
2.1	COLLECTE DES DONNEES :	5
2.2	PRETRAITEMENT DES DONNEES :	5
2.2.1	REDIMENSIONNEMENT :	5
2.2.2	NORMALISATION :	5
2.2.3	AUGMENTATION DES DONNEES :	6
2.3	ORGANISATION DES DONNEES :	6
2.3.1	REPertoire D'ENTRAINEMENT (TRAIN) :	6
2.3.2	REPertoire DE VALIDATION (VAL) :	6
2.3.3	REPertoire DE TEST (TEST) :	6
2.4	GESTION DES DONNEES :	6
2.5	IMPORTANCE DU PRETRAITEMENT ET DE L'ORGANISATION DES DONNEES :	7
3	<u>TECHNOLOGIES UTILISEES :</u>	8
3.1	COMPOSITION DU MODELE CNN-RNN :	8
3.1.1	COUCHE CNN :	8
3.1.2	COUCHE RNN (LSTM) :	8
3.1.3	COUCHES FINALES :	8
3.2	ENVIRONNEMENT DE DEVELOPPEMENT ET BIBLIOTHEQUES :	9
3.2.1	TENSORFLOW ET KERAS :	9
3.2.2	VISUALISATION AVEC MATPLOTLIB ET SEABORN :	9
3.2.3	NUMPY :	9
3.2.4	JUPYTER NOTEBOOK :	9
3.2.5	GIT POUR LA COLLABORATION :	9
3.2.6	STREAMLIT :	9
4	<u>EXPLICATION DU CODE :</u>	10
4.1	ENVIRONNEMENT DE DEVELOPPEMENT :	10
4.2	PREPARATION ET PRETRAITEMENT DES DONNEES :	10
4.3	CONSTRUCTION DU MODELE :	10
4.4	ENTRAINEMENT ET VALIDATION DU MODELE :	11
4.5	ÉVALUATION ET INTERPRETATION :	11
5	<u>ANALYSE DES METRIQUES DE PERFORMANCE :</u>	12
5.1	DETAIL DES RESULTATS OBTENUS :	12
5.1.1	PRECISION ET RAPPEL :	12
5.1.2	SCORE F1 :	12
5.1.3	EXACTITUDE :	13
5.1.4	MATRICE DE CONFUSION :	13
5.2	INTERPRETATION DES RESULTATS :	13

6	CONCLUSION :	14
----------	---------------------------	-----------

1 Introduction :

L'avancement rapide dans le domaine de l'intelligence artificielle a significativement impacté le secteur médical, en particulier dans le domaine de l'imagerie médicale. Les technologies de Deep Learning, notamment les réseaux de neurones convolutifs (CNN) et récurrents (RNN), ont prouvé leur efficacité dans l'analyse et la classification des images médicales, offrant ainsi des outils précieux pour l'assistance au diagnostic et l'amélioration des soins aux patients. Ces technologies permettent non seulement de réduire le temps de diagnostic mais aussi d'améliorer la précision des interprétations, ce qui est crucial dans les situations où chaque minute compte, comme lors de la détection rapide de maladies potentiellement mortelles telles que la pneumonie.

Le projet, qui va dans ce sens, vise à concevoir et développer un modèle de Deep Learning hybride qui intègre les capacités de reconnaissance de motifs des CNN avec la puissance de modélisation séquentielle des RNN.

CNN :

RNN :

L'objectif spécifique est de classifier des images de radiographies thoraciques en deux catégories principales : "Normal" et "Pneumonia". Ce modèle vise à démontrer non seulement une précision élevée mais aussi une capacité à généraliser à partir de données limitées.

Au cours de ce rapport, nous allons détailler l'ensemble du processus de développement du modèle, de l'analyse des données à l'évaluation de ses performances, en passant par le prétraitement des données, la conception du modèle, son entraînement et enfin son analyse. Chaque section apporte un éclairage sur les choix techniques et méthodologiques adoptés, reflétant les

défis et les solutions innovantes mises en œuvre tout au long du projet.

2 Data :

2.1 Collecte des Données :

Le projet utilise l'ensemble de données `chest_xray`, spécifiquement conçu pour la recherche en classification automatique des images de radiographie thoracique. Cet ensemble contient des images divisées en trois catégories principales : entraînement, validation, et test.

Chaque catégorie est ultérieurement classée en deux sous-catégories : 'Normal' et 'Pneumonia'. La répartition initiale inclut 5,232 images pour l'entraînement, 16 images pour la validation, et 624 images pour les tests. Les images de cette base de données proviennent de différentes sources publiques et ont été anonymisées pour respecter la confidentialité des patients.

2.2 Prétraitement des Données :

Le prétraitement des images est une étape cruciale pour assurer la performance du modèle de Deep Learning. Pour notre projet, ce processus inclut plusieurs étapes :

2.2.1 Redimensionnement :

Toutes les images sont redimensionnées à une résolution uniforme de 150x150 pixels pour garantir la cohérence des entrées du modèle.

2.2.2 Normalisation :

Les valeurs des pixels des images sont normalisées pour qu'elles se situent entre 0 et 1. Cette étape est cruciale pour accélérer la convergence du modèle durant l'entraînement en assurant que les gradients restent stables.

2.2.3 Augmentation des données :

Pour améliorer la généralisation du modèle et pour compenser le déséquilibre potentiel des classes dans l'ensemble de données, des techniques d'augmentation d'image comme la rotation, le zoom, et le décalage horizontal sont appliquées de manière aléatoire sur les images d'entraînement.

2.3 Organisation des Données :

Les données sont organisées en structures de répertoires qui facilitent leur accès durant les différentes phases de l'entraînement du modèle :

2.3.1 Répertoire d'entraînement (train) :

Contient la majorité des images et est utilisé pour ajuster les paramètres du modèle.

2.3.2 Répertoire de validation (val) :

Utilisé pour ajuster les hyperparamètres et pour évaluer la performance du modèle pendant l'entraînement afin d'éviter la suradaptation.

2.3.3 Répertoire de test (test) :

Sert à évaluer la performance finale du modèle, fournissant une estimation de son efficacité en pratique.

2.4 Gestion des Données :

Pour gérer efficacement les images et automatiser le processus de leur prétraitement, nous utilisons les fonctionnalités de « ImageDataGenerator » de Keras. Ce composant offre une interface pratique pour :

Appliquer le prétraitement : Comme la normalisation des images directement au moment du chargement, assurant que les images sont correctement formatées pour le modèle.

Augmenter les données : En intégrant des transformations aléatoires pour augmenter artificiellement la variété des images d'entraînement, ce qui aide à

éviter l'overfitting et améliore la robustesse du modèle.

Ces configurations assurent que les images sont non seulement traitées de manière cohérente à travers toutes les phases du projet, mais aussi qu'elles sont adaptées pour optimiser l'apprentissage du modèle en utilisant des techniques d'augmentation de données seulement pour l'ensemble d'entraînement.

2.5 Importance du Prétraitement et de l'Organisation des Données :

Le soin apporté à la préparation des données se reflète directement dans la performance du modèle de Deep Learning. Une bonne pratique en matière de prétraitement et de gestion des données permet non seulement d'améliorer la précision et la capacité de généralisation du modèle, mais également de réduire le risque de sur adaptation et d'accélérer le processus d'entraînement.

En suivant ces étapes rigoureuses, nous nous assurons que notre modèle hybride CNN-RNN est bien équipé pour apprendre des représentations efficaces et précises des images médicales, essentielles pour la classification fiable des pathologies thoraciques.

3 Technologies Utilisées :

Notre projet utilise une architecture combinant les technologies de réseaux de neurones convolutifs (CNN) pour analyser les images, et de réseaux de neurones récurrents (RNN) pour traiter les informations séquentielles. Cela permet au modèle de détecter les motifs visuels importants dans les radiographies tout en prenant en compte les séquences d'images ou d'événements, si nécessaire.

3.1 Composition du Modèle CNN-RNN :

3.1.1 Couche CNN :

- ☐ Les couches convolutives extraient les caractéristiques visuelles des images, comme les formes et les textures.
- ☐ Les couches de pooling réduisent la taille des données traitées, ce qui aide à rendre le modèle plus efficace et moins enclin à l'overfitting.

3.1.2 Couche RNN (LSTM) :

- ☐ Les informations provenant des couches CNN sont passées à une couche LSTM, qui est capable de gérer des données séquentielles et de reconnaître des motifs à travers le temps.

3.1.3 Couches finales :

- ☐ Un mécanisme de dropout est utilisé pour éviter l'overfitting en "éteignant" aléatoirement certains neurones pendant l'entraînement.
- ☐ La couche de sortie utilise une fonction d'activation sigmoid

pour classer les images en catégories 'Normal' ou 'Pneumonia'.

3.2 Environnement de Développement et Bibliothèques :

3.2.1 TensorFlow et Keras :

- Nous utilisons TensorFlow pour gérer les calculs lourds liés au deep learning et Keras pour sa simplicité, permettant de construire et de tester des modèles rapidement.

3.2.2 Visualisation avec Matplotlib et Seaborn :

- Pour examiner les performances du modèle et présenter les résultats, nous employons Matplotlib pour les graphiques et Seaborn pour les cartes de chaleur (heatmap), notamment pour les matrices de confusion.

3.2.3 NumPy :

- Cette bibliothèque est essentielle pour la manipulation des données, nous permettant de transformer et d'ajuster les données d'images facilement.

3.2.4 Jupyter Notebook :

- Les Notebooks Jupyter sont utilisés pour leur capacité à combiner du code exécutable, des visualisations, et des explications textuelles dans un seul document, facilitant ainsi l'expérimentation et la documentation du processus de développement.

3.2.5 Git pour la collaboration :

- L'utilisation de Git nous a aidé à gérer les différentes versions du projet et à collaborer efficacement entre membres de l'équipe, en suivant les modifications et en intégrant les contributions de chacun.

3.2.6 Streamlit

- Nous avons utilisé Streamlit pour la visualisation des graphiques et nous avons mis en place une interface

permettant d'entraîner le modèle directement. De plus, nous avons offert la possibilité de modifier les métriques.

4 Explication du Code :

4.1 Environnement de Développement

Pour ce projet, nous avons utilisé Python en tant que langage de programmation principal, avec une combinaison de plusieurs bibliothèques puissantes pour le traitement des données et le développement de modèles de deep learning. TensorFlow, une bibliothèque de machine learning développée par Google, a été notre choix pour construire, entraîner et tester les modèles de deep learning en raison de sa flexibilité, de sa capacité à évoluer et de son intégration étroite avec Keras, une API de haut niveau pour les réseaux de neurones. Keras simplifie la création de modèles complexes et leur expérimentation, permettant ainsi une itération rapide et une implémentation intuitive des architectures CNN et RNN.

4.2 Préparation et Prétraitement des Données

La première étape du processus informatique a été la préparation et le prétraitement des données. Utilisant `ImageDataGenerator` de Keras, nous avons automatisé le redimensionnement des images, leur normalisation (conversion des valeurs de pixels pour qu'elles soient entre 0 et 1), et l'augmentation des données pour l'ensemble d'entraînement. L'augmentation des données est une technique qui aide à prévenir l'overfitting et à améliorer la généralisation du modèle en créant des versions modifiées des images d'entraînement (par exemple, retournées, décalées, zoomées).

4.3 Construction du Modèle

Le modèle a été construit en empilant des couches dans une architecture séquentielle utilisant l'API de Keras. Nous avons commencé

par des couches deconvolution (`Conv2D`), chacune suivie d'une couche de pooling (`MaxPooling2D`) pour réduire la dimensionnalité tout en capturant les caractéristiques essentielles. Après les couches convolutives, nous avons aplaté les données (`Flatten`) et les avons introduites dans une couche LSTM pour traiter les caractéristiques dans un contexte séquentiel. La dernière partie du réseau est composée d'une couche de dropout pour la régularisation, suivie d'une couche dense qui utilise la fonction d'activation sigmoid pour la classification binaire.

4.4 Entraînement et Validation du Modèle

L'entraînement du modèle a été réalisé en plusieurs phases. Chaque phase d'entraînement comprenait des pas d'entraînement et de validation où le modèle apprenait progressivement à classer les images tout en ajustant ses paramètres internes pour minimiser l'erreur de classification. Nous avons utilisé l'optimiseur Adam pour son efficacité dans la gestion des taux d'apprentissage et la fonction de perte d'entropie croisée binaire, qui est standard pour les tâches de classification binaire.

4.5 Évaluation et Interprétation

Après l'entraînement, le modèle a été évalué sur un ensemble de données de test pour vérifier sa capacité à généraliser à de nouvelles données. La performance a été mesurée en utilisant les métriques décrites précédemment, et les résultats ont été visualisés à l'aide de Matplotlib et Seaborn pour une meilleure interprétation. Des graphiques de la précision et de la perte à travers les époques ont été tracés pour visualiser le processus d'apprentissage et identifier tout signe de suradaptation ou sous-adaptation.

5 Analyse des Métriques de Performance :

Dans le cadre de notre projet de classification d'images médicales, nous avons mesuré la performance de notre modèle CNN-RNN en utilisant plusieurs métriques clés : la précision, le rappel, le score F1, l'exactitude, et la matrice de confusion. Ces indicateurs nous fournissent une vision complète des capacités et des points à améliorer pour notre modèle. En ce qui concerne la classification d'images médicales on obtient une précision globale de 84%, indiquant une bonne capacité à identifier correctement les cas de pneumonie, comparativement aux radiographies normales.

5.1 Détail des Résultats Obtenus :

5.1.1 Précision et Rappel :

Les résultats de notre modèle montrent une précision relativement élevée, ce qui indique que la majorité des prédictions de pneumonie par le modèle sont correctes. Cependant, le rappel, bien que satisfaisant, suggère que le modèle n'a pas détecté tous les cas de pneumonie, laissant passer quelques cas. Cela pourrait être critique dans un contexte médical où manquer un diagnostic de pneumonie pourrait avoir des conséquences graves pour le patient.

5.1.2 Score F1 :

Le score F1, qui équilibre la précision et le rappel, est également assez élevé, 83%. Cela signifie que le modèle est assez robuste en termes de performance générale sur les deux classes. Cependant, la quête d'un équilibre parfait entre précision et rappel reste un défi, surtout compte tenu de l'importance de ne pas

manquer de diagnostics positifs (faux négatifs).

5.1.3 Exactitude :

L'exactitude globale du modèle est de 81% ce qui signifie que, globalement, il fait plus de bonnes prédictions que de mauvaises. Toutefois, cette mesure peut parfois être trompeuse, surtout si les classes sont déséquilibrées. Il est donc important de la considérer en conjonction avec les autres métriques.

5.1.4 Matrice de Confusion :

La matrice de confusion révèle que le modèle a correctement identifié 388 cas de pneumonie sur 390 (vrais positifs) et 117 cas normaux sur 234 (vrais négatifs). Cependant, il a également classé incorrectement 117 cas normaux comme pneumonie (faux positifs) et a manqué 2 cas de pneumonie (faux négatifs). On observe donc qu'il y a beaucoup trop d'erreurs concernant les faux positifs, ce point est particulièrement préoccupant et mérite une attention supplémentaire lors des itérations futures du développement du modèle.

5.2 Interprétation des Résultats :

Ces résultats soulignent l'efficacité du modèle dans la classification des images de radiographie thoracique, tout en mettant en lumière des domaines spécifiques nécessitant des améliorations. Le faible taux de faux négatifs est rassurant, mais leur présence rappelle l'importance de continuer à affiner le modèle pour éviter de passer à côté de diagnostics cruciaux.

Pour les futures versions du modèle, il serait pertinent d'explorer des techniques avancées de traitement des données, d'augmentation des données, avec des données plus équilibrées entre les pneumonies et les

normaux.

6 Conclusion :

En conclusion, notre modèle CNN-RNN a démontré une capacité notable à identifier les cas de pneumonie dans les radiographies thoraciques, avec une forte sensibilité et une exactitude globale satisfaisante. Cependant, les défis demeurent, notamment en termes de précision et de réduction des faux positifs, qui sont essentiels pour améliorer la fiabilité et l'application pratique du modèle en milieu clinique.

Pour les prochaines étapes, il serait judicieux de se concentrer sur l'optimisation de la précision sans compromettre le rappel en explorant des approches telles que :

- **Équilibrage des classes :**

Techniques avancées pour gérer le déséquilibre des classes, telles que le suréchantillonnage des images normales ou le sous-échantillonnage des images de pneumonie.

- **Amélioration du prétraitement :**

Révision des méthodes de prétraitement des images pour mieux généraliser les caractéristiques à partir des données d'entraînement vers les données de test.

- **Ajustement des hyperparamètres :**

Expérimentation avec différents paramètres du modèle pour trouver un meilleur équilibre entre la sensibilité et la spécificité.

- **Incorporation de feedback clinique :**

Collaboration avec des professionnels médicaux pour intégrer leurs retours et ajuster les critères de classification en fonction des besoins cliniques réels.

Le développement de ce modèle représente une avancée significative dans le domaine de l'analyse automatisée d'images médicales et souligne l'importance de la collaboration continue entre les développeurs de technologies de l'IA et les professionnels de la santé pour optimiser l'impact clinique des outils de diagnostic assisté par ordinateur.