

Detection Pneumonie Rapport

Made by: Akram OUNIS, Chawki Belhaddad

1. Introduction

L'avènement des techniques d'intelligence artificielle a ouvert de nouvelles perspectives dans le domaine de la santé, notamment pour l'analyse des images médicales. Ce projet présente une application visant à détecter des cas de pneumonie à partir d'images radiographiques du thorax.

2. Objectif du projet

L'objectif principal de ce projet est de développer un système capable de classifier des images radiologiques pulmonaires en deux catégories : images saines et images indiquant une pneumonie. Ce système aide à accélérer le diagnostic médical et à améliorer la précision de la détection. Pour atteindre cet objectif, trois méthodes sont utilisées : deux basées sur la régression logistique et une troisième utilisant un réseau de neurones convolutifs (CNN). La différence entre les deux approches avec la régression logistique réside dans le format des données d'entrée : la première méthode utilise un fichier CSV avec une seule colonne représentant une combinaison de caractéristiques (filtres Gabor, DCT, Fourier, PHOG), tandis que la deuxième méthode utilise quatre colonnes distinctes pour chaque caractéristique.

3. Contexte médical : La Pneumonie

La pneumonie est une infection respiratoire qui affecte les poumons, causée par des bactéries, des virus ou des champignons. Les radiographies thoraciques sont couramment utilisées pour détecter cette condition, mais l'interprétation des images peut être subjective et demande beaucoup de temps.

4. Méthodologie

4.1 Collecte de données

- **5200 images** radiographiques : 1324 "normales" et 3876 "pneumonie".
- Fournies par le professeur, couvrant divers cas de pneumonie.

4.2 Prétraitement des images

- **Égalisation de l'histogramme et réduction du bruit** (facultatif selon essais).
- **Normalisation** dans le CNN via `layers.Rescaling(1./255)` pour ramener les pixels à $[0,1]$.

4.3 Data Augmentation (pour le CNN)

- Utilisée afin de gérer le déséquilibre entre images "normales" et "pneumonie".
- Rotations, flips horizontaux/verticaux, etc.

4.4 Extraction de caractéristiques (pour la RL)

- **Gabor** : met en évidence textures/orientations.
- **Fourier** : représentation fréquentielle.
- **DCT** : compression et variations spatiales.
- **PHOG** : histogramme des gradients orientés (mise en évidence des contours).

Format des données RL

- **Single Feature** : toutes ces caractéristiques combinées en **une seule colonne**.
- **Multi Features** : chaque caractéristique dans **quatre colonnes distinctes** (Gabor, DCT, Fourier, PHOG).

4.5 Classification

Dans le cadre de la classification par régression logistique, les données extraites via différents filtres ont été normalisées. La normalisation a été effectuée en utilisant la moyenne des valeurs des vecteurs correspondant à chaque filtre. Cela permet de recentrer les données autour de zéro, réduisant ainsi l'effet des variations d'échelle entre les différents types de caractéristiques et améliorant la convergence du modèle d'apprentissage.

1. Régression Logistique (colonne unique)

- CSV = 1 colonne avec toutes les features combinées.

	Numéro d'image	Caractéristique	Étiquette
0	0	0.373273	0
1	1	0.246365	0
2	10	0.322182	0
3	100	0.452206	0
4	1000	0.269899	0
5	1001	0.554350	0
6	1002	0.326523	0
7	1003	0.458850	0
8	1004	0.372907	0
9	1005	0.290041	0

2. Régression Logistique (colonnes multiples)

- CSV = 4 colonnes (Gabor, DCT, Fourier, PHOG).

	Numéro d'image	Gabor	DCT	Fourier	PHOG	Étiquette
0	1565	0.136294	1.250935	0.020754	0.131356	1
1	2902	0.141953	1.820498	0.019932	0.132183	1
2	2013	0.129704	1.135493	0.026567	0.092114	1
3	1164	0.125387	0.914267	0.028794	0.090840	0
4	967	0.125992	1.182731	0.030288	0.100134	0
5	1061	0.102447	0.936791	0.026489	0.103865	0
6	1366	0.117745	1.003780	0.030066	0.156092	1
7	2728	0.148866	2.435213	0.022989	0.182133	1
8	4667	0.116666	1.430416	0.020619	0.079629	1
9	1710	0.110545	1.743097	0.022261	0.175128	1

3. Réseau de Neurones Convolutifs (CNN)

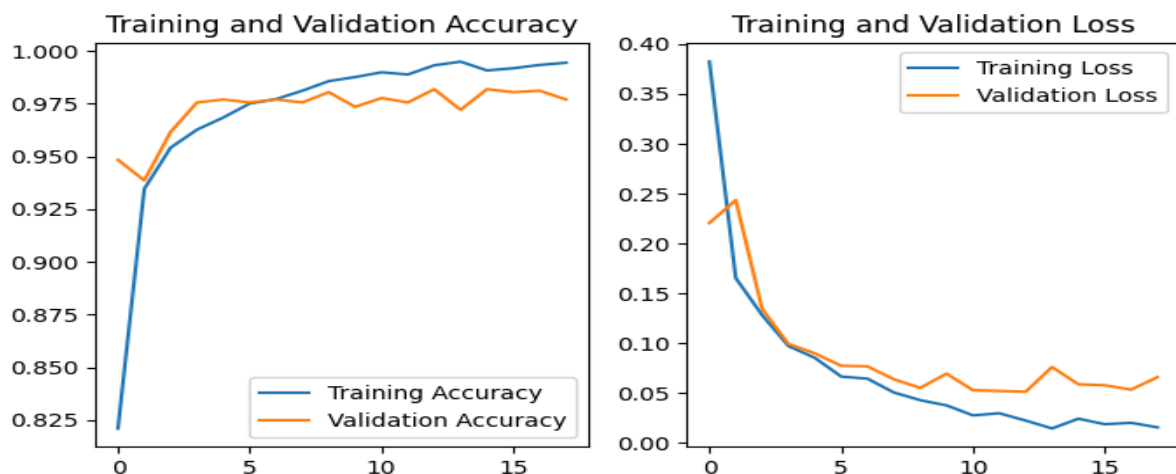
- Apprentissage **direct** sur les images (taille 128×128).
- Empilement de couches Convolution + Pooling + Dense (activation sigmoid).
- **Data augmentation** pour améliorer la généralisation.
- **Normalisation** des images (Rescaling 1/255).

5. Modèle de détection (CNN)

Le CNN est configuré avec :

- **Validation split** = 20 % (`validation_split=0.2`),
- **Batch size** = 32, **img_height** = 128, **img_width** = 128,
- **layers.Rescaling(1./255)** pour la normalisation.

Illustration de l'entraînement (loss et accuracy) :

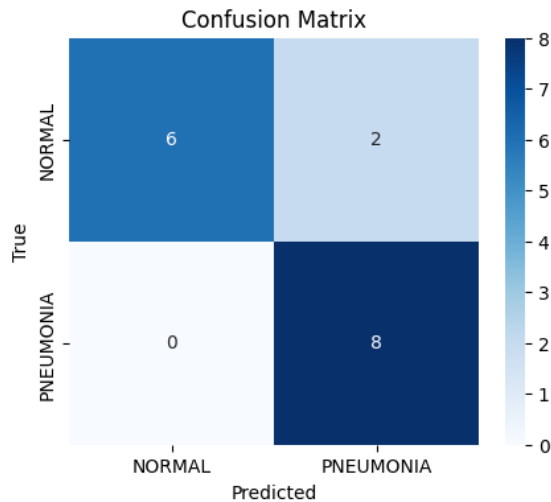


6. Évaluation des performances

Les trois modèles ont été évalués via la **précision (accuracy)**, parfois complétée par une **matrice de confusion** et un **rapport de classification** (précision, rappel, F1-score).

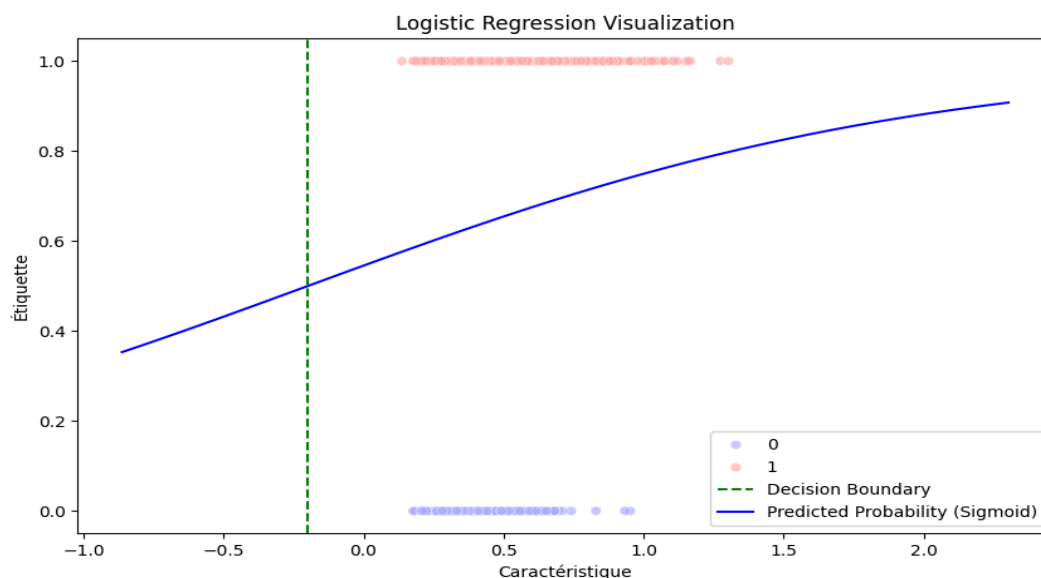
6.1 Résultats du CNN

- **Précision entraînement : 99.44%**
- **Précision validation : 97.70%**
- **Précision test : 87.5%**

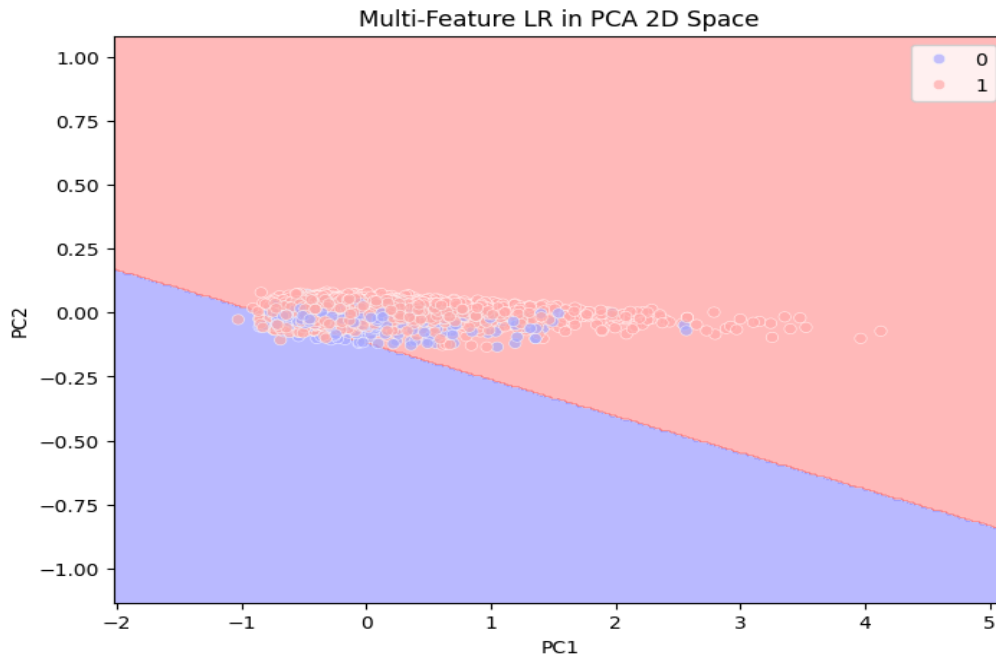


6.2 Résultats de la Régression Logistique

- **Single Feature** (une colonne combinant Gabor, DCT, Fourier, PHOG)
 - **Précision : 73%**



- **Multi Features** (4 colonnes : Gabor, DCT, Fourier, PHOG)
 - **Précision : 73%**



Les deux variantes de RL aboutissent à une précision similaire (0.73), significativement plus basse que celle obtenue par le CNN.

7. Conclusion et perspectives

Le projet montre clairement que :

- Le **CNN** dépasse la régression logistique (accuracy test = 0.875) grâce à la **data augmentation** et la **normalisation**.
- La **régression logistique** à une seule ou plusieurs colonnes de features plafonne à ~0.73, restant toutefois plus simple à mettre en place.
- Une limitation importante rencontrée lors de ce projet réside dans le **déséquilibre des données** entre les classes « normales » et « pneumonie ». La majorité des images du dataset représente des cas de pneumonie, tandis que les images normales sont beaucoup moins nombreuses. Ce déséquilibre peut biaiser les modèles de classification en les rendant plus enclins à prédire la classe majoritaire, ce qui explique en partie pourquoi la précision des modèles de régression logistique plafonne à ~73 %, quelle que soit la méthode d'extraction des caractéristiques utilisée (colonne unique ou colonnes multiples).
- Bien que la **régression logistique** reste une solution simple et rapide à mettre en place, elle est moins performante que les approches basées sur les **réseaux de neurones convolutifs (CNN)**, qui bénéficient de techniques telles que la **data augmentation** et la **normalisation des images** pour mieux gérer le déséquilibre des données.

Perspectives :

Pour améliorer les performances des modèles basés sur la régression logistique, des stratégies de gestion du déséquilibre peuvent être mises en œuvre :

1. **Sous-échantillonnage** : Réduire le nombre d'exemples de la classe majoritaire.
2. **Sur-échantillonnage** : Générer artificiellement plus d'exemples de la classe minoritaire à l'aide de techniques comme **SMOTE**(Synthetic Minority Over-sampling Technique).
3. **Ensembles de modèles** : Utiliser des modèles d'ensemble comme les **Random Forests** ou les **Gradient Boosted Trees**, qui sont plus robustes face aux datasets déséquilibrés.

4. Méthodologie

La méthodologie employée suit plusieurs étapes clés :

- **Collecte de données** : Utilisation des images radiographiques fournies par le professeur, comprenant un total de 5200 images, dont 1324 images normales et 3876 images présentant des cas de pneumonie.
- **Prétraitement des images** : Amélioration de la qualité visuelle des images pour optimiser l'analyse. **Normalisation** dans le CNN via `layers.Rescaling(1./255)` pour ramener les pixels à $[0,1]$.
- **Extraction de caractéristiques** : Pour les modèles basés sur la régression logistique, nous avons utilisé quatre types de filtres pour extraire des attributs pertinents des images :

- **Filtre Gabor** : Utilisé pour capturer les textures et les orientations dans une image, ce filtre est particulièrement efficace pour détecter les structures linéaires.
- **Transformée de Fourier (Fourier)** : Permet de représenter l'image dans le domaine des fréquences, utile pour détecter des motifs périodiques.
- **Transformée en Cosinus Discrète (DCT)** : Réduit les redondances spatiales, couramment utilisée pour la compression et l'analyse des variations.
- **Histogramme des Gradients Orientés (PHOG)** : Extrait les contours en comptant les gradients orientés, apportant des informations de forme et de structure. Ces caractéristiques sont combinées ou séparées selon le format des données d'entrée pour les modèles LR.
- **Classification** :
 - **Régression Logistique (RL) avec une seule colonne de caractéristiques** : Le modèle utilise un fichier CSV où les quatre types de caractéristiques extraites (Gabor, DCT, Fourier, PHOG) sont combinées dans une seule colonne. La régression logistique est utilisée pour modéliser la probabilité qu'une image appartienne à la classe « pneumonie » ou « normale ».
 - **Régression Logistique avec plusieurs colonnes de caractéristiques** : Ici, chaque type de caractéristique est placé dans une colonne distincte. Cette approche permet de mieux capturer les contributions individuelles de chaque caractéristique à la classification.
 - **Réseau de Neurones Convolutifs (CNN)** : Ce modèle d'apprentissage profond apprend directement des images brutes. Il passe par plusieurs couches convolutives pour extraire des représentations complexes avant de prédire la classe de chaque image. Ces trois modèles sont comparés en termes de précision et d'efficacité pour évaluer la méthode la plus adaptée à la tâche de détection de pneumonie.

5. Prétraitement et Extraction de Caractéristiques Le prétraitement comprend des opérations comme l'égalisation de l'histogramme, la réduction du bruit et la normalisation des dimensions des images. Les caractéristiques sont extraites à l'aide d'algorithmes qui analysent les motifs visuels.

6. Modèle de détection Un modèle de réseau de neurones convolutifs (CNN) est souvent utilisé pour les tâches de reconnaissance d'images. Ce modèle comporte plusieurs couches (convolutions, activations, et regroupements) pour apprendre des représentations complexes des images.

7. Évaluation des performances La performance du modèle est mesurée en termes de précision, rappel, et score F1. Une matrice de confusion est utilisée pour analyser les faux positifs et les faux négatifs.

8. Conclusion et perspectives Ce projet démontre l'efficacité des approches basées sur l'apprentissage profond pour le diagnostic médical. Les perspectives incluent l'amélioration du modèle avec des données plus diverses et l'intégration d'une interface utilisateur intuitive.

